Warmup Generations: A Task-Agnostic Approach for Guiding Sequence-to-Sequence Learning with Unsupervised Initial State Generation

Senyu Li[1,2] **Zipeng Sun**[1,2] **Jiayi Wang**[3]
Xue Liu[1,2] **Pontus Stenetorp**[3] **Siva Reddy**[1,2,4] **David Ifeoluwa Adelani**[1,2,4]
[1]Mila - Quebec AI Institute, [2]McGill University, [3]University College London,
[4]Canada CIFAR AI Chair
{senyu.li, siva.reddy, david.adelani}@mila.quebec
{zipeng.sun, xueliu}@mail.mcgill.ca
{jiaywang, p.stenetorp}@cs.ucl.ac.uk

Abstract

Traditional supervised fine-tuning (SFT) strategies for sequence-to-sequence tasks often train models to directly generate the target output. Recent work has shown that guiding models with intermediate steps—such as keywords, outlines, or reasoning chains—can significantly improve performance, coherence, and interpretability. However, these methods often depend on predefined intermediate formats and annotated data, limiting their scalability and generalizability. In this work, we introduce a task-agnostic framework that enables models to generate intermediate "warmup" sequences. These warmup sequences, serving as an initial state for subsequent generation, are optimized to enhance the probability of generating the target sequence without relying on external supervision or human-designed structures. Drawing inspiration from reinforcement learning principles, our method iteratively refines these intermediate steps to maximize their contribution to the final output, similar to reward-driven optimization in reinforcement learning with human feedback. Experimental results across tasks such as translation, summarization, and multi-choice question answering for logical reasoning show that our approach outperforms traditional SFT methods, and offers a scalable and flexible solution for sequence-to-sequence tasks[1].

1 Introduction

Recent advancements in large-scale pre-trained language models (LLMs), such as T5 (Raffel et al., 2019), GPT (Brown et al., 2020), and LLaMA (Touvron et al., 2023), have significantly improved performance on both predictive tasks (e.g., multi-choice question answering) and generative tasks (e.g., machine translation and summarization). These models have demonstrated exceptional capabilities in generating coherent and contextually relevant outputs by modelling dependencies across long sequences of data. Despite their successes, traditional supervised fine-tuning (SFT) methods for such models often focus on directly generating the target output without leveraging the benefits of intermediate steps or initial guidance (Sutskever et al., 2014).

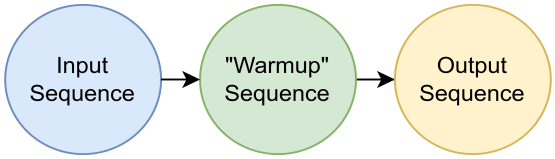

Figure 1: **High-level workflow of Warmup Generations.** The input is first used to generate an intermediate "warmup" sequence, which acts as a guiding context to improve the generation of the final target output for sequence-to-sequence tasks.

Research has shown that guiding models with intermediate steps, such as outlines, keywords, or reasoning chains, can significantly improve performance, coherence, and interpretability across tasks (Wang et al., 2022; Creswell and Shanahan, 2022). For instance, hierarchical frameworks for tasks such as story generation (Fan et al., 2019) and summarization (Amplayo et al., 2020) often first generate high-level structures, such as outlines or reasoning steps, before producing detailed outputs. These approaches highlight the utility of intermediate guidance in organizing complex tasks. Similarly, chain-of-thought (COT) (Wei et al., 2023) reasoning for predictive tasks extends this concept by demonstrating the value of logical steps by decomposing complex predictive tasks into explicit logical steps, demonstrating how structured reasoning between inputs and outputs can improve model performance. However, these approaches heavily rely on predefined intermediate formats and annotated data, which are costly to create due to human annotation efforts and often task-specific, thus limiting their scalability and adaptability to broader

[1]Our codes are available here.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8870–8880

applications.

In this work, we address these limitations by introducing a framework that enables models to generate an initial state, which we refer to as "warmup sequence." These warmup sequences act as preparatory steps, priming the model for the main generation task. Drawing inspiration from reinforcement learning (RL) principles, our method treats these steps as actions within a reward-driven framework, optimizing them to maximize their utility in improving the quality and coherence of the final target output. Importantly, this process operates without relying on predefined formats or external annotations, making it adaptable to a wide range of tasks and model architectures. This approach eliminates dependence on annotated data for intermediate steps, achieves generalization across tasks, and unifies the optimization of intermediate and final outputs, leading to improved final performance of the models.

Through experiments, we demonstrate that our method improves output quality across translation, summarization, and multi-choice question answering for logical reasoning, and is compatible with various model architectures, including encoder-decoder models like T5 (Raffel et al., 2020) and mT5 (Xue et al., 2021), as well as decoder-only models like Llama (Touvron et al., 2023). In addition, our method is simple to implement, requiring only about 10 additional lines of codes, without modifications to existing model architectures or reliance on task-specific annotations, and is grounded in a solid theoretical framework. These contributions establish an approach where models can autonomously discover and leverage a helpful initial state that increases the probability of the target sequence across diverse tasks to enhance the quality of the final generation.

2 Related work

Guiding generative models with intermediate steps has been widely explored to enhance coherence, interpretability, and task performance. Existing approaches can be categorized into explicit human-readable intermediate steps, and structured weakly supervised intermediate steps.

Human readable intermediate steps Plan-and-Write (Yao et al., 2019) introduces storyline-based planning, where a model first generates a structured sequence of key events before expanding them into a full story, improving coherence and creativity. Amplayo et al. (2020) employ content planning, explicitly modelling aspect and sentiment distributions to guide summary generation, thereby enhancing readability and informativeness.

Similarly, Wolfson et al. (2022) propose Question Decomposition Meaning Representations (QDMR), which break down complex questions into sequences of reasoning steps. These decompositions serve as an explicit intermediate representation, improving interpretability and guiding Text-to-SQL parsing by systematically mapping natural language queries to SQL. Baziotis et al. (2019) introduce a sequence-to-sequence-to-sequence model (SEQ³), where the intermediate step is a compressed version of the input sentence, explicitly represented in natural language.

Structured intermediate steps Some models introduce structured but weakly supervised intermediate steps, where the intermediate representations are partially interpretable but not explicitly labelled during training. Cheng et al. (2017) generate predicate-argument structures, which serve as an intermediate step in semantic parsing. Unlike explicit intermediate representations, these structures are learned through optimization-based search rather than direct supervision. Similarly, Jambor and Bahdanau (2022) propose Label Aligned Graphs (LAGr), where models predict node and edge labels to construct structured meaning representations aligned with input text, improving systematic generalization in semantic parsing. These representations enhance compositional generalization but still depend on predefined structural mappings. Herzig et al. (2021) introduce intermediate representations that transform meaning representations (e.g., SPARQL or SQL queries) into structured forms that improve compositional generalization while maintaining reversibility. While these methods balance interpretability and generalization, they still rely on task-specific constraints rather than fully flexible intermediate representations.

Reinforcement learning in NLP RL has also been applied in NLP to optimize model generation beyond traditional supervised learning for text summarization (Paulus et al., 2018), dialogue generation (Li et al., 2016) and machine translation (Wu et al., 2018). More recently, Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017) has been instrumental in aligning large-scale language models with human preferences, demonstrating the effectiveness of RL-based fine-

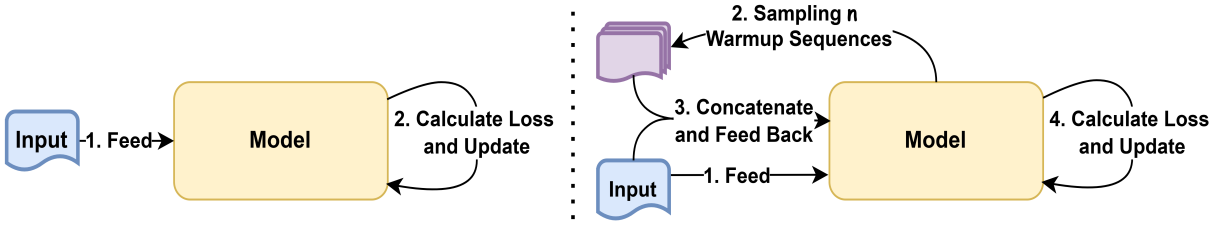

Figure 2: **Comparison of the traditional supervised fine-tuning methods (left) and our proposed method (right)**. The traditional method directly optimizes the mapping from input to output using annotated data, while our method dynamically generates and optimizes warmup sequences to guide the final output.

tuning. While RL provides a strong optimization framework, it does not inherently generate structured intermediate representations, instead refining model behaviour through reward-based learning.

In this paper, we learn intermediate steps freely, without predefined formats, constraints, search procedures, external supervision, annotated datasets or task-specific designs. Inspired by RL principles, our method integrates intermediate step/initial state generation and final output optimization into a unified framework. Our approach generalizes across tasks such as translation, summarization, and multi-choice question answering logical reasoning, and architectures, providing a flexible, scalable, and theoretically grounded solution for improving the quality of generation.

3 Formulation and Derivation

We reformulate the process of text generation by assuming that given a specific **input** x and the **target text** y_{target}, there exists an intermediate sequence, or the **initial state** c_{init} preceding y_{target}, where $\text{length}(c_{\text{init}}) \geq 0$. To be more specific, $c_{\text{init}} = \{c_1, c_2, \ldots, c_k\}$ is a sequence of tokens, where $k \geq 0$. The intermediate sequence c_{init} serves as a latent variable that conditions the generation of y_{target}. When $k = 0$, c_{init} is an empty sequence, reducing the framework to the traditional sequence-to-sequence paradigm:

$$P(y_{\text{target}}|x) = \sum_{c_{\text{init}}} P(c_{\text{init}}, y_{\text{target}}|x)$$

Using the chain rule, this can be decomposed as:

$$P(y_{\text{target}}|x) = \sum_{c_{\text{init}}} P(y_{\text{target}}|c_{\text{init}}, x)P(c_{\text{init}}|x)$$

Which can be rewritten in the form:

$$P(y_{\text{target}}|x) = \mathbb{E}_{c_{\text{init}} \sim P(c_{\text{init}}|x)}[P(y_{\text{target}}|c_{\text{init}}, x)]$$

Our objective is to maximize the probability of the target sequence y_{target} given the input x. Traditionally, the loss for maximizing $P(y_{\text{target}}|x)$ is:

$$L_{y_{\text{target}}} = -\log(P(y_{\text{target}}|x))$$

Which is equivalent to:

$$L_{y_{\text{target}}} = -\log(\mathbb{E}_{c_{\text{init}} \sim P(c_{\text{init}}|x)}[P(y_{\text{target}}|c_{\text{init}}, x)])$$

where c_{init} represents any possible initial state conditioning y_{target}. This expectation implies maximizing $P(y_{\text{target}})$ across all initial states, weighted by their probability $P(c_{\text{init}}|x)$.

Reward-Based Initial State Optimization

Since we lack labels for c_{init}, we train the model to generate c_{init} using reward-based optimization. A good initial state c_{init} increases the probability of y_{target}, while a poor c_{init} reduces it. The reward $R(c_{\text{init}})$ quantifies the quality of c_{init} in terms of its contribution to generating y_{target} given x:

$$R(c_{\text{init}}) = P(y_{\text{target}}|c_{\text{init}}, x)$$

Under a RL framework, we aim to maximize:

$$\mathbb{E}_{c_{\text{init}} \sim P(c_{\text{init}}|x)}[R(c_{\text{init}})]$$

which is equivalent to:

$$\mathbb{E}_{c_{\text{init}} \sim P(c_{\text{init}}|x)}[P(y_{\text{target}}|c_{\text{init}}, x)]$$

The loss for training c_{init} generation is defined as the negative log of the expected reward:

$$\begin{aligned} L_{c_{\text{init}}} &= -\log\left(\mathbb{E}_{c_{\text{init}} \sim P(c_{\text{init}}|x)}[R(c_{\text{init}})]\right) \\ &= -\log\left(\mathbb{E}_{c_{\text{init}} \sim P(c_{\text{init}}|x)}[P(y_{\text{target}}|c_{\text{init}}, x)]\right) \end{aligned}$$

As we can observe, this formulation aligns the optimization of y_{target} and c_{init} under the same loss function.

$$L_{c_{\text{init}}} = L_{y_{\text{target}}}$$

Directly minimizing $L_{c_{\text{init}}}$ or $L_{y_{\text{target}}}$ is computationally infeasible due to numerical underflow for long sequences. Instead, we minimize the expected cross-entropy loss based on Jensen's inequality:

Algorithm 1: Warmup Generations

Input: Training Data, Maximum Epochs E, Number of Samples n, Model Parameters θ

Output: Trained θ

Initialize:
  Model θ with pretrained weights;
for *epoch* $t = 1$ *to* E **do**
  for *each input* x *in Training Data* **do**
    Initialize total loss $\mathcal{L}_{\text{total}} \leftarrow 0$;
    for $i = 1$ *to* n **do**
      Sample $c_{\text{init}}^{(i)} \sim P(c_{\text{init}}|x;\theta)$;
      Compute loss $\mathcal{L}^{(i)}$ for $y_{\text{target}}^{(i)}$ conditioned on $c_{\text{init}}^{(i)}$ and x;
      $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + \mathcal{L}^{(i)}$;
    $\mathcal{L}_{\text{avg}} \leftarrow \mathcal{L}_{\text{total}}/n$;
    Update θ using gradient descent;
return *Trained* θ

$$
\begin{aligned}
&-\log\left(\mathbb{E}_{c_{\text{init}}\sim P(c_{\text{init}}|x)}[P(y_{\text{target}}|c_{\text{init}},x)]\right) \\
&\quad \leq \mathbb{E}_{c_{\text{init}}\sim P(c_{\text{init}}|x)}[-\log(P(y_{\text{target}}|c_{\text{init}},x))]
\end{aligned}
$$

Minimizing the expected cross-entropy loss indirectly minimizes an upper bound on $-\log(P(y_{\text{target}}|x))$, bringing us closer to maximizing $P(y_{\text{target}}|x)$.

To approximate the expected value, we use Monte Carlo sampling with n samples of c_{init}:

$$
\begin{aligned}
L_{\text{final}} &= \mathbb{E}_{c_{\text{init}}\sim P(c_{\text{init}}|x)}[-\log(P(y_{\text{target}}|c_{\text{init}},x))] \\
&\approx \frac{1}{n}\sum_{i=1}^{n} -\log(P(y_{\text{target}}|c_{\text{init},i},x))
\end{aligned}
$$

Thus, minimizing the expected cross-entropy loss over sampled contexts is an effective approach to optimize text generation tasks.

4 Warmup Generations Approach

A general overview of our method is illustrated in Figure 2. Unlike traditional SFT methods, where the loss is computed solely based on the input, our approach introduces an intermediate generation step. After receiving the input, the model first generates n warmup sequences. The final loss is then computed as the average of n individual losses, each conditioned on both the input and one of the generated warmup sequences. The pseudo-code outlining the implementation of our method is provided in algorithm 1.

4.1 Implementation for Encoder-Decoder Models

Encoder-decoder structured models have a clear separation between input and output. For this type of model, the input is processed through the encoder, and then n c_{init} are generated using beam search with sampling. Each generated c_{init} is followed by a separator and fed into the decoder. Subsequently, the cross-entropy loss of y_{target} is calculated n times, conditioned on the input and each of the n generated c_{init}. Finally, the average of these n losses is taken as the final loss.

The inference process follows the same logic shown in Figure Figure 2, given an input sequence, the model first generates a warmup sequence. This sequence is then concatenated with a separator and fed back into the beginning of the decoder. The model then generates the target sequence conditioning on both the original input and the generated warmup sequence. The final output consists of the tokens generated after the concatenated separator.

4.2 Implementation for Decoder-Only Models

Similar to encoder-decoder models, the input is first fed into the model, and n c_{init} are sampled using beam search. The input is then concatenated with n c_{init} sequences, followed by a separator, and fed back into the model. The final loss is the average cross-entropy loss of y_{target} conditioned on the n c_{init} sequences and the input.

During inference, similar to encoder-decoder models, the warmup sequence is first generated and then appended to the input sequence, followed by a separator. The combined sequence is then fed back into the model to generate the target sequence. The final output consists of the tokens produced after the concatenated separator.

4.3 Rationale Behind Using a Separator Between c_{init} and y_{target}

The inclusion of separators helps the model to distinguish the boundary between c_{init} and y_{target}, preventing y_{target} from being treated as a continuation of c_{init}. This enhances both the stability and efficiency of training. Since the separators are deterministically appended to the end of each c_{init}, the probability distributions of c_{init} and c_{init}

Model	Warmup	Macro F1	Accuracy
T5-base	✓	**50.00**	**50.06**
	×	49.16	49.19
T5-large	✓	**55.50**	**55.62**
	×	54.46	54.54
Llama-3.2-1B	✓	**32.63**	**33.55**
	×	30.12	31.70

Table 1: Performance of each model on LogiQA2 for Macro F1 and Accuracy.

followed by the separator remain the same. Additionally, since the model is rewarded based on the generation of c_{init}, the inclusion of separators does not disrupt this training process.

5 Experiments

In this section, we present the tasks and corresponding datasets used, the models selected for the experiments and the results obtained.

5.1 Tasks and Datasets

We evaluated our approach on three datasets spanning three tasks: FLORES (Team et al., 2022) for testing, WMT for training for the translation task; LogiQA2 (Liu et al., 2023) for logical reasoning multi-choice QA; and XSum (Narayan et al., 2018) for summarization; Specifically, we used WMT19 datasets (Barrault et al., 2019) for the fine-tuning of de-en (en-de), ru-en (en-ru), and zh-en (en-zh) and the fr-en (en-fr) data from the WMT14 dataset (Bojar et al., 2014).

5.2 Models

We used T5-base (223M) and T5-large (738M) for summarization, T5-base, T5-large, and Llama-3.2-1B (1.24B) for multiple-choice logical reasoning, and mT5-base (582M) and mT5-large (1.23B) for translation. These models, covering both encoder-decoder and decoder-only architectures, serve as well-established benchmarks in their respective categories and are widely recognized for their effectiveness.

5.3 Metrics

We used the BLEU score (Papineni et al., 2002), COMET score [2] (Rei et al., 2020), and ChrF++ score (Popović, 2015) for translation. For logical reasoning multiple-choice, we used macro F1 and accuracy, and for summarization, we employed ROUGE score (1, 2, L) (Lin, 2004) and BERTScore [3] (Zhang* et al., 2020).

5.4 Experimental Settings

For fine-tuning all tasks, we used a learning rate of 2e-5, with the warmup sequence's maximum sampled length capped at 8 tokens. Models were trained for 10 epochs. Due to computational constraints, we randomly selected 50,000 samples from the training set for fine-tuning in translation and summarization tasks. During fine-tuning, warmup sequences were generated using a beam size of 4, with 4 warmup sequences sampled per training sample for each loss calculation.

For each task, we selected the checkpoint that achieved the highest metric score on the validation set and reported its performance on the test set. Specifically, for translation, checkpoints were selected based on the COMET score; for summarization, based on BERTScore; and for logical reasoning multiple-choice, based on the macro F1 score. For LogiQA2, each model was fine-tuned 3 times with different random seeds, and we report the average performance of the selected checkpoints.

5.5 Results and Discussions

We put our experiment results in Table 1, 2, and 3.

Warmup generations consistently enhance performance across tasks Across all three tasks, models utilizing warmup generations outperform those employing traditional SFT methods. The most significant gains are observed in translation tasks, where mT5-base achieves an average improvement of 1.57 BLEU, 1.32 COMET, and 1.60 ChrF++ scores across 8 language pairs. For multiple-choice logical reasoning, the T5-base model trained with warmup generations achieves 0.84 higher macro F1 and 0.87 higher accuracy compared to models using traditional SFT. A similar trend is observed in summarization, where the T5-base model with warmup generations yields gains of 0.45 ROUGE-1, 0.32 R2, 0.36 RL, and 0.1 BERTScore.

Performance gains are robust to increases in model size When scaling the models from base to large, we observe similar or even greater performance gains. In translation, mT5-large exhibits a greater average improvement than mT5-base, with a higher BLEU gain of 1.69, a COMET increase

[2]The implementation of COMET was from huggingface.

[3]The implementation of BERTScore was from huggingface, with the model type set to "roberta-large".

Model	Warmup	Metric	de-en	ru-en	zh-en	fr-en	en-de	en-ru	en-zh	en-fr	Avg
mt5-base	✓	BLEU	**29.64**	**20.78**	**13.31**	**30.64**	**20.43**	**11.72**	**22.64**	**27.95**	**22.14**
	×		27.89	19.41	12.66	28.73	18.10	10.64	20.95	26.14	20.57
	✓	COMET	**83.12**	**78.19**	**77.05**	**82.99**	**75.11**	**69.64**	**75.89**	**76.89**	**77.36**
	×		82.21	77.60	75.50	81.95	72.88	68.13	74.45	75.60	76.04
	✓	ChrF++	**55.15**	**46.48**	**39.48**	**55.64**	**47.78**	**32.09**	**25.05**	**52.66**	**44.29**
	×		53.63	45.57	38.01	53.84	45.43	30.71	23.44	50.87	42.69
mt5-large	✓	BLEU	**34.71**	**25.76**	**18.66**	**35.63**	**26.84**	**16.30**	**27.89**	**36.29**	**27.76**
	×		34.12	24.93	17.76	34.84	24.91	14.29	27.29	30.39	26.07
	✓	COMET	**86.74**	**82.51**	**82.54**	**86.37**	**82.67**	**77.43**	**82.19**	**83.50**	**82.99**
	×		86.28	82.30	81.83	86.12	81.47	74.43	82.18	80.91	81.94
	✓	ChrF++	**59.65**	**51.15**	**45.45**	**59.39**	**53.19**	**38.34**	**29.72**	**59.28**	**49.52**
	×		58.75	50.80	44.21	58.87	51.44	34.47	29.17	54.33	47.75

Table 2: Performance on translation tasks with the comparison of the BLEU score, COMET score and ChrF++ score across different language pairs. For example, "de-en" denotes the source language to be German, and the target language to be English.

Model	Warmup	Rouge R1	Rouge R2	Rouge RL	BERT Score
T5-base	✓	**37.63**	15.51	30.32	**90.50**
	×	37.18	15.19	29.96	90.40
T5-large	✓	**40.67**	18.23	33.21	**91.11**
	×	40.21	17.84	32.75	91.05

Table 3: Performance on summarization tasks for Rouge (R1, R2 and RL) and BERTScore.

of 1.05 (slightly lower but still comparable), and a greater ChrF++ gain of 1.77. For logical reasoning, T5-large benefits more from warmup generations than T5-base, achieving performance gains of 1.04 in Macro F1 and 1.08 in Accuracy. Similarly, in summarization, T5-large outperforms its counterpart without warmup generations, with improvements of 0.46, 0.39, and 0.46 in ROUGE-1, ROUGE-2, and ROUGE-L scores, respectively.

Performance gain extends to decoder-only architecture The decoder-only model, Llama-3.2-1B, gains from warmup generations in multiple-choice logical reasoning, achieving performance improvements of 2.51 in Macro F1 and 1.85 in Accuracy.

Warmup generations improve lexical alignment more than semantic richness in summarization For summarization, using warmup generations achieves BERTScore increases by 0.06–0.1 points on a scale of 100, indicating that while warmup sequences enhance word selection and fluency, they do not significantly impact semantic richness. This suggests that, for summarization, warmup sequences help the model better mimic the word choices made in the reference summaries, leading to higher word-level alignment (ROUGE scores). However, they do not push the model to generate additional semantic content beyond what it would traditionally learn to extract through standard training, which explains the smaller improvement in BERTScore.

Overall, warmup sequences consistently improve performance across tasks, models, and architectures, but their effectiveness is task-dependent. For encoder-decoder models, logical reasoning benefits more in larger models, while translation shows variable gains depending on the language pair. Summarization, in contrast, benefits uniformly across scales. While the decoder-only model, LLaMA-3.2-1B can leverage warmup sequences effectively, its relative performance remains lower than encoder-decoder models like T5.

5.6 Ablation Studies

5.6.1 Number of Samples

To assess the impact of the number of sampled warmup sequences during training, we analyze both training loss trends and test set performance across different sample numbers.

As shown in Figure 3, increasing the number of sampled warmup sequences generally accelerates convergence and reduces final training loss. However, the differences between 4 and 6 sequences for LogiQA2, and 4, 6, and 8 sequences for de-en translation are relatively small, suggesting that beyond a certain threshold, additional samples do not significantly reduce training loss further.

The results for LogiQA2 in Table 6 indicate that increasing the number of samples does not lead to strictly monotonic improvements. The default

Lang Pair	Warmup Sequence	Content
		Direct Core Phrases
de-en	"pyramid is the only one of"	T: The Cheops pyramid is the only one of the seven world wonders ... G: The Great Pyramid at Giza is the only one of the seven wonders ...
fr-en	"a British traveller in"	T: Similarly, a British traveller in Spain could confuse ... G: Similarly, a British traveller in Spain may mistake ...
zh-en	"a search on the Internet for"	T: Search on the Internet for a response to hostile environment courses ... G: A search of the Internet for 'Hostile environment course' ...
ru-en	"because there was no national"	T: ... and because there was no national executive or judicial power, ... G: ... and, because there was no national executive or judiciary, ...
		Similar Phrases
de-en	"female travelers are recommended"	T: Women: It is recommended that all women travelling claim to be married, ... G: Women: It is recommended that any women travellers say that they are ...
fr-en	"Please contact us directly"	T: In all cases, you must reserve by telephone directly from the aircompany. G: In all cases, you must book by phone directly with the airline.
zh-en	"shows a changing temperature"	T: The ultraviolet image shows that the changes in the night temperature ... G: Infrared images show that the temperature variations from night and day ...
ru-en	"According to the Japanese nuclear"	T: According to the nuclear authority of Japan, radioactive cezai and ... G: According to Japan's nuclear agency, radioactive caesium and iodine ...

Table 4: Examples of Direct Core Phrases and Similar Phrases for different language pairs. "T" indicates translations that are model-generate, and "G" indicates the golden label.

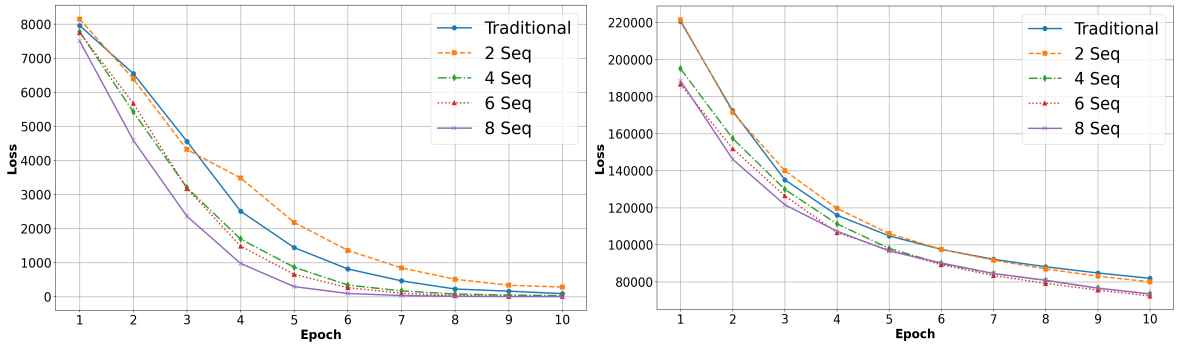

Figure 3: **Models' training loss after each epoch of fine-tuning**. The left figure represents T5-base on the LogiQA2 dataset, while the right graph represents mT5-base on the WMT19 dataset for the de-en language pair.

setting of 4 samples achieves 50.00 Macro F1 and 50.06 Accuracy while increasing to 6 samples provides only a slight improvement (50.19 Macro F1 and 50.19 Accuracy). However, moving from 6 to 8 samples results in the largest jump, with Macro F1 increasing to 50.99 and Accuracy to 50.95. Notably, reducing the number of samples to 2 still achieves 50.09 Macro F1 and 50.17 accuracy, this suggests that even a small number of warmup samples can provide meaningful improvements.

The results for de-en translation exhibit a similar trend. While increasing the number of samples improves translation quality up to 6 sequences, beyond this point, the gains become marginal or even slightly decrease. Specifically, BLEU improves from 28.51 (2 samples) to 29.70 (6 samples) but slightly drops to 29.62 with 8 samples. COMET score follows a similar pattern, increasing from 82.45 to 83.24 before slightly decreasing to 83.06, while ChrF++ continues improving slightly, though the changes are minimal beyond 6 samples.

These findings suggest that while increasing the number of warmup sequences reduces training loss and improves test-time performance, there exists an optimal range—such as 6 to 8 samples for de-en translation—beyond which the benefits diminish. The initial improvements stem from better approximations of the probability distribution of warmup

Lang Pair	Devtest (%)	Dev (%)
de-en	55.05	56.57
en-de	32.71	33.41
fr-en	40.94	42.75
en-fr	34.42	32.97
zh-en	43.91	42.90
en-zh	63.03	63.54
ru-en	46.13	48.80
en-ru	33.51	33.48

Table 5: Overlap rates of the warmup sequence for different language pairs on Flores200 devtest and dev datasets, generated by the mT5-base model.

	de-en translation			LogiQA2	
Sample	BLEU	COMET	ChrF++	Macro F1	Accuracy
4	29.64	83.12	55.15	50.00	50.06
2	28.51	82.45	54.07	50.09	50.17
6	**29.70**	**83.24**	55.26	50.19	50.19
8	29.62	83.06	**55.38**	**50.99**	**50.95**

Table 6: Performance of mT5-base (de-en translation) and T5-base (LogiQA2) models trained with varying numbers of warmup sequences sampled during training.

sequences, leading to more effective learning. However, adding too many samples can introduce redundancy or increased variance, limiting further performance gains.

5.7 Qualitative Analysis

We performed a qualitative analysis of the translation task to investigate the role of warmup sequences. As the results shown in Table 4, we found that these warmup sequences can be primarily categorized into two types:

- Direct Core Phrases: These warmup sequences can be directly identified in both the labels and the generated outputs.
- Similar Phrases: Expressions that are semantically similar to important components in the labels and outputs.

For example, "a British traveller in" and "a series of events that" can be directly found in both the output and ground-truth labels. This indicates that for certain scenarios, initial states function as core-information extractors, guiding the model to generate outputs focusing on these core concepts.

Meanwhile, in other cases, the warmup sequences are more semantically related rather than exact phrase matches. For instance, "when they are in danger" is semantically related to "they perceive a threat" in the label, and "Please contact us directly" aligns with "book by phone directly with the airline." In such scenarios, the initial states serve as semantic guides, enabling the model to generate outputs that capture the intended meaning without relying on exact phrase matching.

To further evaluate this dual role, we calculated the overlapping rate of the words in the initial states that also appear in the labels in Table 5. As we can see, "X-eng" all achieve a overlapping rate over 40%, with "de-en" reaching 56.57% at the most. On "Eng-X", the overlapping rate also reach at least 32.71% on "en-de", and get to the highest on "en-zh" with the rate of 63.53%. This means that generally, over 40% of the words in the warmup sequence could be found in the ground-truth, indicating a strong alignment between the initial states and the ground-truth data. By acting as both direct extractors and semantic interpreters, the initial states ensure the generated outputs remain closely aligned with the intended semantics and structure of the target language.

6 Conclusions

In this work, we introduced a task-agnostic framework with theoretical proof and derivation, for improving sequence-to-sequence learning through warmup generations, where models learn to generate intermediate sequences to enhance final output quality. Unlike traditional approaches, our method learns intermediate steps in an unsupervised manner, improving performance across diverse tasks without requiring task-specific annotations. Experiments demonstrate that warmup sequences consistently benefit both encoder-decoder and decoder-only models across different sizes. Analysis reveals that warmup sequences aid generation by extracting key phrases and providing semantically related guidance, resulting in more fluent and contextually accurate outputs. Additionally, increasing the number of sampled warmup sequences accelerates convergence and enhances test-time performance, though gains diminish beyond a certain threshold. However, the performance gains vary across tasks and architectures, highlighting the need for further investigation into how different task types and model structures influence warmup effectiveness. Overall, by introducing and demonstrating the effectiveness of warmup sequences across multiple seq2seq tasks, this work lays the groundwork for further research into leveraging intermediate generations to enhance model training and generation.

Acknowledgment

This paper originated as a course project in a graduate-level class at McGill University and was supported by the IVADO R3AI Regroupement Grant. We would like to thank Arkil Patel for his valuable support and guidance as the course teaching assistant during the early stages of this work.

Limitations

While our proposed framework demonstrates improvements across various tasks, there are several limitations to address. The first is increased training time. The framework relies on sampling multiple initial states during training, introducing computational overhead compared to traditional supervised fine-tuning methods. This can make training more resource-intensive, particularly for large-scale datasets or deployment in constrained environments. Future work could explore more efficient sampling strategies or adaptive selection methods to mitigate this cost. Another limitation is that warmup sequences primarily enhance the model's lexical-level understanding rather than deeper reasoning or structural-level improvements. As shown in our experiments, warmup generation aids the model in selecting key phrases and improving word choice, but it does not explicitly introduce or infer new knowledge beyond what is present in the input. Future research could explore how warmup sequences might be adapted to facilitate higher-level abstraction or knowledge augmentation, potentially bridging gaps in implicit reasoning. Finally, our framework has not been tested on decoder-only models for generative tasks. While experiments on LogiQA2 demonstrate improvements for decoder-only architectures, the application of warmup sequences to open-ended text generation (e.g., summarization or translation) in decoder-only models remains unexplored. This poses potential challenges, as decoder-only models lack explicit input-output alignments found in sequence-to-sequence tasks, making it unclear whether warmup sequences would be equally effective. Investigating warmup generation within causal language models is an important direction for future work.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2020. Unsupervised opinion summarization with content planning. *Preprint*, arXiv:2012.07808.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQ^3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 673–681, Minneapolis, Minnesota. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2017. Learning structured natural language representations for semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44–55, Vancouver, Canada. Association for Computational Linguistics.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *Preprint*, arXiv:2208.14271.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.

Jonathan Herzig, Peter Shaw, Ming-Wei Chang, Kelvin Guu, Panupong Pasupat, and Yuan Zhang. 2021. Unlocking compositional generalization in pre-trained models using intermediate representations. *Preprint*, arXiv:2104.07478.

Dora Jambor and Dzmitry Bahdanau. 2022. LAGr: Label aligned graphs for better systematic generalization in semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3295–3308, Dublin, Ireland. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023. Logiqa 2.0 — an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1–16.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Preprint*, arXiv:1409.3215.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Rationale-augmented ensembles in language models. *Preprint*, arXiv:2207.00747.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Tomer Wolfson, Daniel Deutch, and Jonathan Berant. 2022. Weakly supervised text-to-SQL parsing through question decomposition. In *Findings of the*

Association for Computational Linguistics: NAACL 2022, pages 2528–2542, Seattle, United States. Association for Computational Linguistics.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. *Preprint*, arXiv:2010.11934.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7378–7385.

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

A Selection of Seperator

The separator token for T5 and mT5 was set to " || ", as this symbol is rarely used in natural text, making it an ideal choice for separating different parts of the sequence.

B Warmup Sequence for Summarization and Logical Reasoning

The warmup sequences for summarization follow a similar pattern to those in translation, predominantly consisting of either Direct Core Phrases or Similar Phrases. For logical reasoning, the warmup sequence is identical to the target sequence, representing only the final letter choice that indicates the predicted answer.