

Hansel: Output Length Controlling Framework for Large Language Models

Seoha Song*, Junhyun Lee*[†], Hyeonmok Ko

Samsung Research
{seoha.song, junhyun8.lee, felix.ko}@samsung.com

Abstract

Despite the great success of large language models (LLMs), efficiently controlling the length of the output sequence still remains a challenge. In this paper, we propose Hansel, an efficient framework for length control in LLMs without affecting its generation ability. Hansel utilizes periodically outputted hidden special tokens to keep track of the remaining target length of the output sequence. Together with techniques to avoid abrupt termination of the output, this seemingly simple method proved to be efficient and versatile, while not harming the coherency and fluency of the generated text. The framework can be applied to any pre-trained LLMs during the finetuning stage of the model, regardless of its original positional encoding method. We demonstrate this by finetuning four different LLMs with Hansel and show that the mean absolute error of the output sequence decreases significantly in every model and dataset compared to the prompt-based length control finetuning. Moreover, the framework showed a substantially improved ability to extrapolate to target lengths unseen during finetuning, such as long dialog responses or extremely short summaries. This indicates that the model learns the general means of length control, rather than learning to match output lengths to those seen during training.

Extended version — <https://arxiv.org/abs/2412.14033>

Introduction

Large language models (LLMs) have shown great promise in generating coherent and fluent text, making them useful for a variety of tasks (Arivazhagan et al. 2019; Karpukhin et al. 2020; Lewis et al. 2020; Zhang et al. 2020). However, a number of challenges still remain for LLMs, including the difficulty of efficiently controlling the length of the output sequence. Length control of the output sequence has practical importance in the real-world application of LLMs. Setting the length determines the level of details to include in the output, which is desired both in the aspect of content and system interface. While this task seems relatively mundane compared to the emergent abilities of LLMs, to the best of our knowledge length control has yet to be achieved even

*These authors contributed equally.

[†]Corresponding author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

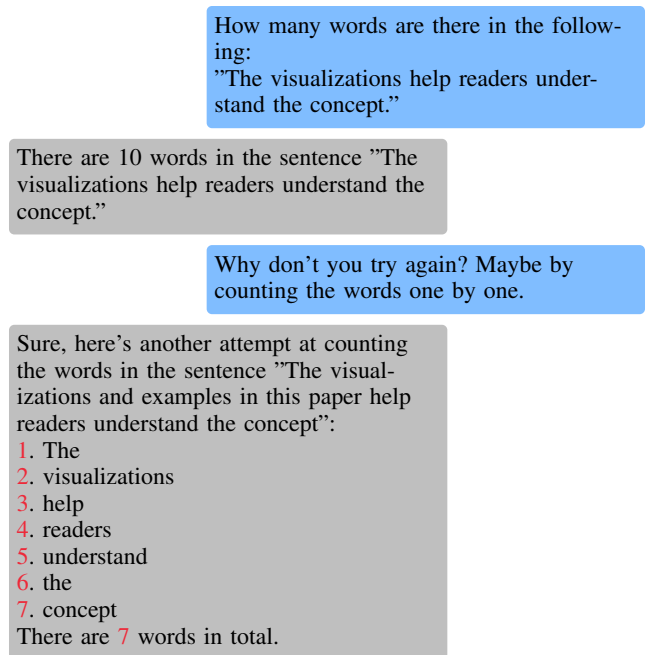


Figure 1: An example conversation with GPT. We have edited the example sentence for brevity.

for the larger LLMs (Achiam et al. 2023; Anil et al. 2023; Touvron et al. 2023) with hundreds of billions of parameters.

A couple of specific examples where length control is useful would be i) a news app where articles are accompanied by AI-generated summaries, or ii) a voice assistant that can tune the amount of information to be spoken. For the news app, the default length should be concise enough to fit the system interface, and the user may want to tune the summarization length to include or drop details. Even for simple voice assistant tasks such as telling the weather, “rain at noon” may be sufficient but sometimes precipitation or temperature might be needed. These examples show the benefits of various outputs from the identical input and controlling them is important and useful.

The study of length control dates back to the pre-transformer era. Kikuchi et al. (2016) first suggested feeding

the remaining length information to the LSTM as an embedding or the memory cell. After the advent of the transformer (Vaswani et al. 2017), there were attempts to include the length information in a modified form of positional encoding (Takase and Okazaki 2019). Other studies turn to the attention mechanism such as Length Attention (Yu et al. 2021) and Length-Aware Attention (Liu, Jia, and Zhu 2022).

Although the previous methods of length control were fairly successful, they mostly concentrate on the relatively small sequence-to-sequence models (Sutskever, Vinyals, and Le 2014) and not the nowadays prevalent decoder models. The main reason for this is two-fold. First, the aforementioned methods require application in the pretraining stage and show significant performance degradation when used only while finetuning. Large decoder models are mostly pre-trained once as a foundation model and finetuned for downstream tasks due to their large size, making the previous methods inapplicable. Second, with their better understanding of the context, the decoder models have some, although imperfect, length-controlling ability. The decoder LLMs would not output the exact required length, for example, 17 words, but it does understand commands such as shorter/longer and will tend to output longer texts when specified to output 25 words than when asked for 10 words.

In this paper, we introduce a simple *finetuning* framework to improve the output length-controlling ability of LLMs. We first observe why generating outputs with a certain length is a difficult task for LLMs. LLMs are known to have limited mathematical ability even with the significant improvement from the zero-shot-CoT technique (Kojima et al. 2022). This not only applies to complex math problems but also to simple counting. Figure 1 shows a short example conversation with GPT¹, slightly edited for brevity. Because LLMs are not intrinsic counters, GPT sometimes outputs incorrect answers when asked to answer the number of words in a sentence. One way to correct this behavior is to prompt the model to explicitly count the words first and then output the answer as in Figure 1.

We build upon this observation and apply it to the task of output length control. With its lack of counting ability, the model would not be aware of how many words it has output and how many words are left at a certain point. To overcome this, we introduce the Hansel – **H**idden **A**rrangements (of special tokens) in **N**atural **S**equences for **E**xpected **L**ength – framework, which directly injects the information via the form of hidden special tokens. By augmenting a finetuning dataset with special tokens appearing regularly, the model can efficiently learn the relative position compared to the target length at a given time. This can be considered a simple hard-coded CoT method specifically for output length control. Our method performs well even when only used during finetuning without degradation in output quality. Phi2 (Li et al. 2023) based Hansel model showed great performance in mean absolute error from the reference length in four summarization and dialogue datasets (Table 1). Moreover,

¹We have used GPT-3.5 to generate the example. GPT-4 can generate similar conversations by explicitly instructing not to call Python scripts.

the error remains small for a wide range of arbitrary target lengths, where other methods diverge rapidly as the target length deviates from the dataset’s typical length (Figure 3).

Contribution The main contributions of this paper are as follows:

- We introduce the Hansel framework that learns the output length control during finetuning through repeated hidden tokens, which can be applied to decoder LLMs with large number of parameters.
- We empirically validate the effectiveness of Hansel by experimenting with it on four different datasets, various models, and several target lengths.
- We conduct a thorough comparison with the prompt-based length control method and show that Hansel outperforms the prompt-based method as it learns the general means of length control.

Related Work

Majority of previous studies have been done with sequence-to-sequence models. LenEmb and LenInit (Kikuchi et al. 2016) are learning-based methods where the length information is provided via an extra embedding and the LSTM memory cell, respectively. Takase and Okazaki (2019) modified the Transformer’s positional encoding to include the remaining length information. Fan, Grangier, and Auli (2018) train on examples with the length information (in ranges) included as a special marker and use the target length marker during inference. Other methods include utilizing the attention mechanism (Yu et al. 2021; Liu, Jia, and Zhu 2022) and prompts (Zhang et al. 2022b; He et al. 2022). While showing some promise in length-controlling ability, most methods previously mentioned require special pre-training and/or structural modification to the baseline model. Our method has some similarities with Fan, Grangier, and Auli (2018) in the sense that we use a special marker, and Zhang et al. (2022b) in that we utilize prompts. However, we use the special tokens periodically which is the crucial difference.

Decoder LLMs are now more common than encoder-decoder models, and they mostly come in more than billions of parameters. They tend to intrinsically have some degree of sense of the output length, and may properly reply to prompts such as “give me a shorter/longer answer.” The paradigm of length control in such models involves prompt learning (Raffel et al. 2020; Sanh et al. 2022; Liu et al. 2023a). Goyal, Li, and Durrett (2022) showed good results on prompt-based length control but only in the units of sentences. There is also work on prompt-based length control in the framework of reinforcement learning (Jie et al. 2023).

Recently, a new set of studies on length-control have emerged in the context of direct preference optimization (DPO) (Rafailov et al. 2024). DPO tends to generate longer responses (Singhal et al. 2024), which is in part because they are preferred over shorter ones while evaluation (Dubois et al. 2024). To prevent this over-verbosity, a number of papers have suggested methods to length-control, or length-desensitize during DPO. These include R-DPO (Park

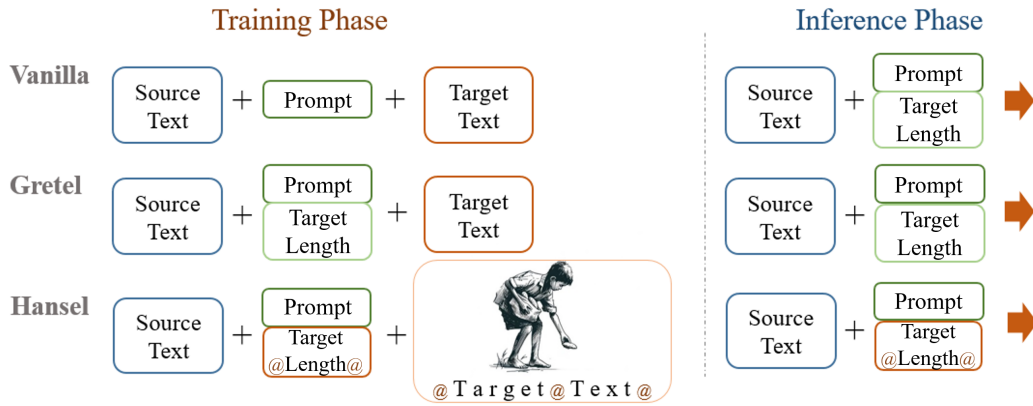


Figure 2: Schematic of the Hansel framework, compared with the vanilla and Gretel scheme. Vanilla is normal fine-tuning and Gretel is the prompt-based length-aware fine-tuning. Hansel receives the target length as a special token and regularly places additional special tokens (marked as @ in the figure) that inform the position while fine-tuning.

et al. 2024), SimPO (Meng, Xia, and Chen 2024), LIFT-DPO (Yuan et al. 2024), SamPO (Lu et al. 2024), and LD-DPO (Liu et al. 2024). However, with a different motivation, these works do not require outputs of specific lengths.

Method

As we have observed in Figure 1, LLMs are intrinsically not well suited for counting. The key idea of Hansel is to overcome such lack of counting ability by explicit counting and use it for counting the output words. In Hansel, we augment the fine-tuning training set by placing special tokens that indicate the remaining number of words. When fine-tuned with such dataset, the special tokens will also appear while inference and guide the model to output the desired length.

The Hansel Dataset

The Hansel framework utilize the ‘‘Hansel dataset,’’ which is the original dataset augmented with special tokens that indicate the remaining length of the output. (Thus, giving the name Hansel from the German fairy tale ‘‘Hansel and Gretel.’’ Hansel leaves a trail of pebbles in the woods to find the way back home.) From now on, we count the units of length as *words* unless specified. The method is trivial to apply in different units, such as characters, tokens, and sentences. Empirically, the difficulty of length control increases from sentence, token, word, and character.

Below we take an example from CNN/DM.

[Original] :

*Famous American foods created across United States.
Connecticut diner claims creation of the hamburger.
Onion rings were courtesy of cook at Pig Stand in Texas.*

[Hansel] :

$|2\rangle\langle 5|$ *Famous American foods created across* $|2\rangle$ *United States. Connecticut diner claims creation of the hamburger. Onion* $|1\rangle$ *rings were courtesy of cook at Pig Stand in Texas.* $|0\rangle$

The augmented special token $|x\rangle\langle y|$ means that there are $\Delta x + y$ words left until the target length and $|x\rangle \equiv |x\rangle\langle 0|$. Hyperparameter Δ is the stride between special tokens, excluding the very first special token.

For each example, we count the number of words in the reference output (l) and include $| \lfloor l/\Delta \rfloor \rangle \langle l\% \Delta |$ at the beginning. After $l\% \Delta$ words we insert $| \lfloor l/\Delta \rfloor \rangle$. The special tokens $| \lfloor l/\Delta \rfloor - 1 \rangle, | \lfloor l/\Delta \rfloor - 2 \rangle, \dots, | 0 \rangle$ are inserted in Δ words interval until the end of the reference. In the above CNN/DM example, $l = 25$ and $\Delta = 10$. Since $\lfloor l/\Delta \rfloor = 2$ and $l\% \Delta = 5$, the first special token becomes $|2\rangle\langle 5|$ and $|2\rangle$ follows after 5 words.

When finetuning with the above dataset, there is a risk that the output will be abruptly terminated after the $|0\rangle$ token without actually finishing the sentence. To avoid this, we introduce one more hyperparameter δ , which indicates the maximum residual words after the $|0\rangle$ token. We set aside a portion of examples and consider the total length as $l - 1, l - 2, \dots, l - \delta$ instead of l . The purpose of a nonzero δ is to train with examples where the response does not terminate with $|0\rangle$, and thus educate the model to finish the sentence naturally even when an unfinished response encounters $|0\rangle$. The previous example for $\delta = 2$ would be as follows,

[Hansel ($\delta = 2$)] :

$|2\rangle\langle 3|$ *Famous American foods* $|2\rangle$ *created across United States. Connecticut diner claims creation of the* $|1\rangle$ *hamburger. Onion rings were courtesy of cook at Pig Stand* $|0\rangle$ *in Texas.*

We randomly choose 20% of the samples and assign $l - 1, l - 2, \dots, l - \delta$ uniformly. We also label mask $N = 10$ tokens preceding $|0\rangle$ while training. This is to ensure that the model does not explicitly learn incomplete sentences preceding $|0\rangle$. A nonzero δ plays a crucial role in the Hansel framework preventing abrupt termination and its treatment while training will be detailed in the following section.

The Hansel Framework

The Hansel framework essentially finetunes the pre-trained model with the Hansel dataset. One full example consists of a source, a prompt including the target length of the output, and the Hansel augmented output. The target length information is provided as the first special token.

We compare the Hansel framework with two baseline methods – prompt finetuning and the finetuning without length information. We dub the two as the Gretel and vanilla framework, respectively. Comparing the three frameworks’ training example, Hansel has [source]-[prompt with target length (special token)]-[Hansel output]; Gretel has [source]-[prompt with target length]-[original output]; and vanilla has [source]-[prompt without target length]-[original output]. For inference Gretel have [source]-[prompt with target length] as the context, while Hansel has [source]-[prompt with target length (special token)]. Vanilla has two versions, [source]-[prompt with/without target length]. We specifically dub the former (with target length) version as vanilla*.

An analogy with the three methods can be made for a person giving a talk. The vanilla model would correspond to giving a talk for a specific duration without any time-keeping practice. Gretel is trained by knowing how long each practice talk was, and required to give one for a certain time. Hansel’s practice talk was done with the knowledge of the total time, and a time-keeper who reminds the speaker every 5 minutes – and the same time-keeper attends the actual talk as well. The schematic of the three frameworks is depicted in Figure 2. We also present training and inference examples for each in the Appendix².

Experimental Setup

We used batch size 512 and AdamW optimizer (Loshchilov and Hutter 2018) with 5×10^{-5} learning-rate and parameters $\beta_1 = 0.9$, $\beta_2 = 0.95$. We finetune the pretrained LLMs for 2 epochs. See the Appendix for how the number of epochs affects the performance. Finetuning and inference are conducted using 8 Nvidia Tesla V100 GPUs (40GB). We set the maximum token number to 1722 and truncated longer examples.

Dataset

We consider two tasks in this work, text summarization and dialogue response. Experiments are done with two datasets for each task.

CNN/DM (Hermann et al. 2015) is a large-scale news dataset from the Cable News Network (CNN) and the Daily Mail (DM). We follow (Nallapati et al. 2016), and treat the news articles as the source documents and the corresponding highlights (usually in bullet points) as the summaries.

XSum (Narayan, Cohen, and Lapata 2018) is a highly abstractive summarization dataset of news articles from the British Broadcasting Corporation (BBC). BBC has one-line headlines at the beginning of each article, which are regarded as summaries.

²Please refer to the extended version (arxiv:2412.14033) for the Appendix.

DailyDialog (Li et al. 2017) is a multi-turn dialogue dataset on various daily conversation topics, including relationships, ordinary life, and work. The raw data are crawled from English-learning websites and thus are human-written.

MultiWOZ (Zang et al. 2020) is a task-oriented dialogue dataset spanning 8 domains – restaurant, hotel, attraction, taxi, train, hospital, bus, and police. MultiWOZ also consists of human-human written conversation.

Metric

For evaluation, we use both metrics that measure the performance of the length control and the quality of the output.

Mean Absolute Error To evaluate the length control ability of the model, we simply use the mean absolute error (MAE) of the output length from the target length. The MAE is measured as follows:

$$\text{MAE} = \frac{1}{N} \sum^N |l(\text{output}) - l(\text{target})|. \quad (1)$$

$l(x)$ is the length of x and N is the number of samples.

ROUGE Recall-Oriented Understudy for Gisting Evaluation (Lin 2004) measures the lexical similarity between two texts by comparing the overlap of n -grams. Variants of ROUGE include the n -gram matching ROUGE- n (R- n) and the longest common subsequence matching ROUGE-L (R-L). Among these, we show R-L in the main text and include results for R-1, R-2 in the Appendix. We use the Google Research implementation of ROUGE (<https://github.com/google-research/google-research/tree/master/rouge>) with the default settings.

G-Eval To overcome the lexical bias from ROUGE, we also evaluate G-Eval (Liu et al. 2023b), which is a framework for using LLMs with chain-of-thoughts (CoT) and a form-filling paradigm to assess the quality of natural language generation outputs. For the summarization task, the evaluation is in coherence (on a scale of 1-5), consistency (1-5), fluency (1-3), relevance (1-5); for the dialogue task, naturalness (1-5), coherence (1-5), engagingness (1-3), and groundedness (1-5). The score scale and the used prompts are from the original paper (Liu et al. 2023b). We use GPT-4 (Achiam et al. 2023) as the evaluator.

Baseline Model

To demonstrate that Hansel is applicable regardless of the base model’s positional representation method, we perform experiments on four different types of position methods – Rotary positional embeddings, ALiBi, learned positional embeddings, and T5 bias. For consistency, we used models with parameter sizes within the range of 2.7B to 3.0B for all methods.

Rotary Rotary positional embedding (Su et al. 2024) encodes the position with rotation matrices, only retaining information on purely relative position. For the model using rotary positional embedding, we use Microsoft’s Phi-2 with 2.7B parameters (Li et al. 2023).

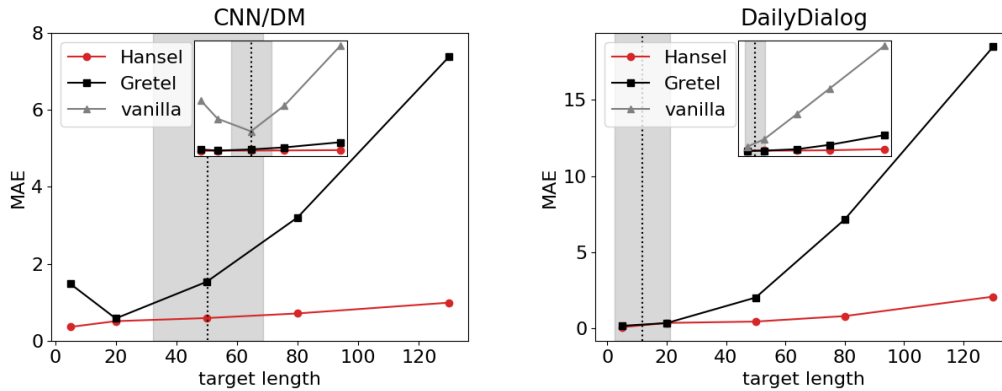


Figure 3: The extrapolation of the length control methods with different target lengths. The dashed line (shaded region) indicates the mean length (\pm standard deviation) of the dataset. While the MAE of other methods increases drastically when the target length is different from that of the dataset, our method (Hansel) shows robust performance.

Dataset	Model	MAE	R-L	G-Eval
DailyDialog	vanilla*	7.67	0.11	2.54
	vanilla	7.01	0.25	3.51
	Gretel	0.24	0.27	3.41
	Hansel	0.09	0.33	3.55
MultiWOZ	vanilla*	7.24	0.13	1.86
	vanilla	5.50	0.30	3.59
	Gretel	0.11	0.34	3.55
	Hansel	0.05	0.34	3.54
XSum	vanilla*	5.35	0.33	3.18
	vanilla	5.35	0.33	3.20
	Gretel	0.47	0.33	3.22
	Hansel	0.26	0.34	3.26
CNN/DM	vanilla*	15.23	0.30	3.84
	vanilla	14.82	0.30	3.86
	Gretel	1.56	0.30	3.82
	Hansel	0.39	0.31	3.86

Table 1: The length control experiment using Phi-2 as the pre-trained LLM. The target length is set to the reference’s actual length. Hansel and Gretel both showed good results without performance degradation.

ALiBi ALiBi (Press, Smith, and Lewis 2022) is another relative positional method that is known to extrapolate well to longer sequences while inferring. This adds a bias linear to the relative distance between the tokens. We experiment on BLOOM-3B (Workshop et al. 2022) for ALiBi.

Learned positional embedding Positional embedding uses absolute embeddings that are added to token embeddings. Unlike the other absolute method of positional encoding (Vaswani et al. 2017), the positional embeddings are learned throughout training. We use OPT-2.7B (Zhang et al. 2022a) for this method.

T5 Bias The T5 bias (Raffel et al. 2020) is a relative positional embedding method (Shaw, Uszkoreit, and Vaswani

Dataset	Model	$l(\text{target})$				
		5	20	50	80	130
DailyDialog	vanilla	5.17	13.62	43.11	73.06	122.99
	Gretel	0.15	0.33	2.01	7.15	18.51
	Hansel	0.06	0.34	0.43	0.79	2.07
MultiWOZ	vanilla	5.76	9.73	39.39	69.39	119.39
	Gretel	0.12	0.17	2.89	9.96	23.85
	Hansel	0.03	0.12	0.32	0.78	7.47
XSum	vanilla	13.71	4.30	32.34	62.29	112.23
	Gretel	0.96	0.50	1.80	3.42	10.37
	Hansel	0.49	0.31	0.45	0.64	1.02
CNN/DM	vanilla	41.39	26.55	16.28	37.21	85.66
	Gretel	1.49	0.59	1.54	3.21	7.38
	Hansel	0.37	0.52	0.60	0.72	1.00

Table 2: The length extrapolation results (MAE) of Phi-2. This is also plotted in Figure 3.

2018) similar to rotary and ALiBi but with a learned bias for each distance. We use the natural choice of T5-3B representing this method.

Results

We first use Phi-2, which adapts rotary positional embedding, as the baseline model and finetune it with the four datasets in the three frameworks – vanilla, Gretel, and Hansel. We then measure their performance using the test sets, asking the model to perform the task in a length-controlled manner. Here the target length was that of the reference. We set $\Delta = 20$ and $\delta = 1$ for this experiment. Note that when the model infinitely generates repeated output, we exclude them from the statistics such as MAE, ROUGE-L, and G-Eval. Instead we count the infinite generates samples separately and present them in the next section.

The result is shown in Table 1. First, we comment on the vanilla*. Vanilla* uses the prompt with target length while trained without such target. (recall that vanilla is the conven-

Task	Context
Summarization	Article: A group of scientists have invented a gadget which turns pee into electricity. They hope it could be used in some poorer countries, as a way to create clean and renewable energy. The system was developed by scientists at the Bristol Bioenergy Centre (BBiC) in the Bristol Robotics Laboratory at the University of the West of England (UWE Bristol). The invention will be tried out this week on festival-goers at the Glastonbury music festival in Somerset. Urine, or wee is fed into a special machine called a microbial fuel cell, which contains a type of bacteria. These tiny bacteria break down the wee and create electrons. These electrons are then forced through an electronic circuit, creating a flow of electricity. This electricity can then be used to power all sorts of things, including your phone! [Summarize in # words.]
5 words	Scientists have invented a generator.
20 words	Scientists have come up with a new way to make electricity – by using wee, or urine, as a power source. Scientists have invented a machine which turns wee into electricity. It works by using bacteria to break down the wee. This creates electrons which are forced through an electronic circuit. This electricity can then be used to power all sorts of things, including your phone! Scientists hope it could be used in some poorer countries, as a way to create clean and renewable energy. It will be tried out this week on festival-goers at the Glastonbury music festival in Somerset.
80 words	Scientists have come up with a new way to make electricity – by using wee, or urine, as a power source. Scientists have invented a machine which turns wee into electricity. It works by using bacteria to break down the wee. This creates electrons which are forced through an electronic circuit. This electricity can then be used to power all sorts of things, including your phone! Scientists hope it could be used in some poorer countries, as a way to create clean and renewable energy. It will be tried out this week on festival-goers at the Glastonbury music festival in Somerset.
Abrupt termination	A: I am looking for a hotel. B: There are a total of 9 hotels. Huntingdon Marriotte Hotel is a good choice with a 4 star rating. Would you like there phone number? A: I actually need a place that has free wifi and is in the south part of town. And it needs to be a hotel, not a guesthouse. [Reply in 18 words.]
$\delta = 0$	There are two hotels that meet your needs. The Aylesbray Lodge Guest House and The Lensfield Hotel. Any?
$\delta = 1$	The Lensfield Hotel is a hotel in the south with free wifi. Would you like to book it?
Multiple units	A: Have you any round-neck sweater? [Reply in # sentences and # words.]
4 s 20 w	We have. Take a look at this one. It’s made of cotton. It’s very breathable and it’s not expensive either.
3 s 30 w	We have some nice round-neck sweaters. They are the latest fashion and are very popular with young people. Why don’t you try one on and see how you like it?
2 s 40 w	We have some nice round-neck sweaters in different styles and colors. If you like this one, you can try it on in the fitting room over there, and if you like it, you can buy it at the cashier’s desk.

Table 3: Example context and outputs of the Phi-2 Hansel model. $\Delta = 20$ and $\delta = 1$ unless otherwise stated.

tional fine-tuned model) In dialogue datasets, this results in a model breakdown and the output quality greatly degrades, which can be checked in the ROUGE-L and G-Eval scores. Length control is meaningless if the model loses its generation ability, and therefore we do not consider the vanilla* model from now on. The vanilla model does not attempt the length control as the prompt does not include the target length, but its results are included in Table 1 to demonstrate the other models (Hansel and Gretel) does not have quality degradation (ROUGE-L and G-Eval). We include an expanded version of Table 1 in the Appendix which includes R-1, R-2, and the four different measures in G-Eval.

The Gretel model shows significant improvement over the vanilla on length control with a much smaller MAE. The Hansel model also shows good length control performance, better than that of Gretel. Hansel’s MAE decrease over that of Gretel ranged from 45% to 75% among datasets. One salient feature is that Hansel’s length control ability did not degrade as much for long-target datasets, while the MAE tends to be larger with a longer target length. Due to its long average target length, CNN/DM has the largest MAE for all three frameworks, however Hansel improved the MAE by a significant 75% over Gretel.

Target length extrapolation While the result in Table 1 showed a better length control ability of Hansel, especially for long targets, one might consider Gretel model already

has fairly small MAE and would be enough. To demonstrate the true strength of Hansel, we now turn to a different and perhaps a more important setting of length control with arbitrary target length.

There is a distinction between length control *over a dataset* and the model having a *general* length control capability regardless of the trained dataset. With the target length set as the reference length, Table 1 demonstrated the former. For the model to claim general length control ability, it should be able to target arbitrary lengths, different from the reference length seen during training. We target 5, 20, 50, 80, and 130 words and measure the MAE for each case. The results are shown in Table 2 and is plotted in Figure 3. (the other two plots are presented in the Appendix) We also show an example of the Hansel model outputs for a summarization task in Table 3, requested to generate outputs in various lengths.

For all datasets and models, the MAE increases as the target length deviates from \pm (standard deviation) of the dataset (which is depicted as shaded regions in the figure). However, Hansel shows a robustly small (order of magnitude smaller than Gretel for longest targets) MAE throughout the test range. This indicates that the Hansel model has learned how to control the output length in general, while Gretel’s length control ability is more specific to the trained dataset. This ability of Hansel is beneficial for both training efficiency and

Dataset	Δ	δ			
		0	1	3	5
DailyDialog	10	0.05	0.08	0.12	0.16
	20	0.08	0.09	0.12	0.16
	40	0.16	0.13	0.17	0.17
MultiWOZ	10	0.03	0.04	0.05	0.07
	20	0.05	0.05	0.07	0.09
	40	0.09	0.08	0.08	0.09
XSum	10	0.10	0.20	0.26	0.37
	20	0.19	0.26	0.32	0.38
	40	0.27	0.25	0.45	0.44
CNN/DM	10	0.09	0.40	0.60	0.83
	20	0.17	0.39	0.60	0.76
	40	0.53	0.49	0.81	0.88

Table 4: The effect of hyperparameters Δ and δ on the length control ability. All entries are MAE compared with the reference length. Minimum values excluding $\delta = 0$ in bold.

Dataset	Model	ALiBi	Learned	T5 Bias
DailyDialog	vanilla	6.98	6.00	8.69
	Gretel	0.24	0.60	0.33
	Hansel	0.15	0.21	0.18
MultiWOZ	vanilla	5.67	5.68	5.63
	Gretel	0.14	0.21	0.25
	Hansel	0.12	0.11	0.22
XSum	vanilla	5.25	5.03	4.73
	Gretel	0.47	0.69	0.92
	Hansel	0.39	0.37	0.47
CNN/DM	vanilla	14.46	14.70	13.80
	Gretel	1.78	2.80	3.62
	Hansel	0.98	0.69	1.09

Table 5: The length control experiments showing MAE with baseline models using different positional embeddings. The target length is set to the reference’s actual length.

robustness in actual applications of length control.

The effect of hyperparameters We investigate the effect of the hyperparameters Δ and δ . We vary Δ from $\{10, 20, 40\}$ and δ from $\{0, 1, 3, 5\}$ and measure the MAE and show the results in Table 4.

The MAE increases with δ as we have trained the model with examples with larger residuals. Although $\delta = 0$ results were best in terms of MAE, we avoid selecting those from the quality perspective. Manually going through $\delta = 0$ outputs we identified instances where the output was abruptly terminated near the target length. We show an example of this in Table 3. The nonzero δ was introduced for precisely this reason and the result suggests $\delta = 1$ is sufficient for that purpose. Small Δ is also unnecessary (as it generates excessive special tokens) and we conclude that $\Delta = 20$ and $\delta = 1$ is the best combination of hyperparameters.

Various positional embeddings As the Hansel framework only requires the Hansel dataset and does not alter the model architecture or positional embeddings, it can be applied to any baseline models. To demonstrate this, we experiment on three more models, BLOOM, OPT, and T5. These represent models with ALiBi, learned, and T5 positional embeddings, respectively. Moreover, the T5 is an encoder-decoder model while the others are decoder models. The results are presented in Table 5.

While each model has a different base performance, the tendency is largely similar to Table 1 – Hansel controls length the best without significant loss of quality. This demonstrates the adaptability of the Hansel framework.

Infinite generation Additionally, there is an interesting byproduct of this length control. Infinite generation is a quite common problem in LLMs, which gets more prominent for small on-device models. There are reports of this issue even in very recent models such as Llama 3 (Dubey et al. 2024). If the model has the length controlling ability, this automatically solves the infinite generation issue. For the length extrapolation results in Table 2, vanilla and Gretel experienced infinite generation 8.4 and 15.3 times per 10,000 generations, respectively, while there was no such incident for Hansel. Detailed data of this experiment is shown in the Appendix.

Versatility and generalizability The framework is also versatile enough to control multiple units. For example, it is possible to train a model that can control both the number of sentences and words. One just needs to prepare the Hansel dataset which consists of two different families of special tokens, each for a sentence and word. An identical algorithm can be applied for augmentation. Examples of sentence and word number control are shown in Table 3. Moreover, Hansel’s length control can also work as on the fly – one just need to specify the special token to turn on the length control ability.

We have also performed ablation studies to ensure Hansel does not affect the output distribution and is generalizable to other domains and tasks. The model’s output on a separate task was not affected by the special tokens, and found that the generalization was possible with only 50 additional instruction samples. Please refer to the Appendix for details of the ablation studies.

Conclusion

We have introduced the Hansel framework that efficiently enables LLMs to control the length of their output. The Hansel framework only utilizes the Hansel dataset which is an augmentation from the original dataset, and thus can be applied to models regardless of architecture and positional encoding. Moreover, it is large pre-trained model friendly as it works through finetuning, and retains its length control ability even when the target length is out of the typical range of the finetuned dataset. The method outperforms the traditional prompt-based finetuning in the MAE from the target length. It especially showed significant improvement in extrapolated target length, demonstrating that the model learns the general means of length control.

References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Anil, R.; Dai, A. M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Arivazhagan, N.; Bapna, A.; Firat, O.; Lepikhin, D.; Johnson, M.; Krikun, M.; Chen, M. X.; Cao, Y.; Foster, G.; Cherry, C.; et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.
- Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P. S.; and Hashimoto, T. B. 2024. AlpacaFarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.
- Fan, A.; Grangier, D.; and Auli, M. 2018. Controllable Abstractive Summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*.
- Goyal, T.; Li, J. J.; and Durrett, G. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.
- He, J.; Kryscinski, W.; McCann, B.; Rajani, N.; and Xiong, C. 2022. CTRLsum: Towards Generic Controllable Text Summarization. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.
- Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.
- Jie, R.; Meng, X.; Shang, L.; Jiang, X.; and Liu, Q. 2023. Prompt-Based Length Controlled Generation with Reinforcement Learning. *arXiv preprint arXiv:2308.12030*.
- Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Kikuchi, Y.; Neubig, G.; Sasano, R.; Takamura, H.; and Okumura, M. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.
- Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35.
- Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Li, Y.; Bubeck, S.; Eldan, R.; Del Giorno, A.; Gunasekar, S.; and Lee, Y. T. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.
- Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; and Niu, S. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.
- Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.
- Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023a. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.*, 55(9).
- Liu, W.; Bai, Y.; Han, C.; Weng, R.; Xu, J.; Cao, X.; Wang, J.; and Cai, X. 2024. Length Desensitization in Directed Preference Optimization. *arXiv preprint arXiv:2409.06411*.
- Liu, Y.; Iter, D.; Xu, Y.; Wang, S.; Xu, R.; and Zhu, C. 2023b. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Liu, Y.; Jia, Q.; and Zhu, K. 2022. Length control in abstractive summarization by pretraining information selection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
- Lu, J.; Li, J.; An, S.; Zhao, M.; He, Y.; Yin, D.; and Sun, X. 2024. Eliminating Biased Length Reliance of Direct Preference Optimization via Down-Sampled KL Divergence. *arXiv preprint arXiv:2406.10957*.
- Meng, Y.; Xia, M.; and Chen, D. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.
- Nallapati, R.; Zhou, B.; dos Santos, C.; Guulçehre, Ç.; and Xiang, B. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In Riezler, S.; and Goldberg, Y., eds., *Proceedings of the 20th SIGLL Conference on Computational Natural Language Learning*.
- Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Park, R.; Rafailov, R.; Ermon, S.; and Finn, C. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.
- Press, O.; Smith, N. A.; and Lewis, M. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *10th International Conference on Learning Representations (ICLR 2022)*.

- Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.
- Sanh, V.; Webson, A.; Raffel, C.; Bach, S.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Raja, A.; Dey, M.; Bari, M. S.; Xu, C.; Thakker, U.; Sharma, S. S.; Szczechla, E.; Kim, T.; Chhablani, G.; Nayak, N.; Datta, D.; Chang, J.; Jiang, M. T.-J.; Wang, H.; Manica, M.; Shen, S.; Yong, Z. X.; Pandey, H.; Bawden, R.; Wang, T.; Neeraj, T.; Rozen, J.; Sharma, A.; Santilli, A.; Fevry, T.; Fries, J. A.; Teehan, R.; Scao, T. L.; Biderman, S.; Gao, L.; Wolf, T.; and Rush, A. M. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations*.
- Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.
- Singhal, P.; Goyal, T.; Xu, J.; and Durrett, G. 2024. A Long Way to Go: Investigating Length Correlations in RLHF. In *First Conference on Language Modeling*.
- Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Takase, S.; and Okazaki, N. 2019. Positional Encoding to Control Output Sequence Length. In Burstein, J.; Doran, C.; and Solorio, T., eds., *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)*.
- Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Workshop, B.; Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- Yu, Z.; Wu, Z.; Zheng, H.; XuanYuan, Z.; Fong, J.; and Su, W. 2021. LenAtten: An Effective Length Controlling Unit For Text Summarization. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.
- Yuan, W.; Kulikov, I.; Yu, P.; Cho, K.; Sukhbaatar, S.; Weston, J.; and Xu, J. 2024. Following length constraints in instructions. *arXiv preprint arXiv:2406.17744*.
- Zang, X.; Rastogi, A.; Sunkara, S.; Gupta, R.; Zhang, J.; and Chen, J. 2020. MultiWOZ 2.2: A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*.
- Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*. PMLR.
- Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022a. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- Zhang, Y.; Zhang, X.; Wang, X.; Chen, S.-q.; and Wei, F. 2022b. Latent prompt tuning for text summarization. *arXiv preprint arXiv:2211.01837*.