國立臺灣大學電機資訊學院暨中央研究院

資料科學學位學程

碩士論文

Data Science Degree Program

College of Electrical Engineering and Computer Science

National Taiwan University and Academia Sinica

Master Thesis

外掛式語言模型：利用一個簡單的迴歸模型控制文本生成

Plug-in Language Model: Controlling Text Generation with a Simple Regression Model

楊奈其

Nai-Chi Yang

指導教授: 馬偉雲 博士、鄭卜壬 博士

Advisor: Wei-Yun Ma, Ph.D., Pu-Jen Cheng, Ph.D.

中華民國 112 年 8 月

August, 2023

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

摘要

大型預訓練語言模型（LLMs）在海量數據的訓練中展示出無與倫比的能力，已經能夠生成與人類極為相似的文本。然而，在不進行微調或增加額外參數的條件下生成符合特定條件的文本，仍然是一個具有挑戰性的任務。

目前避免修改語言模型的策略，主要使用 prompts 或外加的分類器。這些分類器被開發用於決定或預測生成的 token 是否有助於達成所需目標。這些方法通過利用所需屬性的預測分數計算梯度，從而在推理階段改變下一個 token 的輸出分佈。然而，這些分類器模型通常需要使用語言模型的潛在狀態為輸入，這阻礙了使用許多現成的黑盒模型或工具。

為了克服這些限制，我們提出了外掛式語言模型（PiLM）作為解決方案。PiLM 利用強化學習直接使用黑盒工具協助調整潛在狀態來達成控制文本生成。同時我們訓練一個簡單的回歸模型取代反向傳播梯度這一緩慢的過程，使 PiLM 幾乎不會增加生成文本所需時間成本。通過在三種控制生成任務上的驗證，我們的方法展示出優於現有的基於梯度更新、加權解碼或使用 prompts 的方法的成果。

關鍵字：控制生成、自然語言生成、預訓練語言模型、強化學習、機器學習、深度學習

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

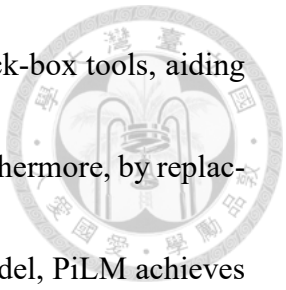
Abstract

Large-scale pre-trained language models (LLMs), trained on massive datasets, have displayed unrivaled capacity in generating text that closely resembles human-written text. Nevertheless, generating texts adhering to specific conditions without finetuning or the addition of new parameters proves to be a challenging task.

Current strategies, which avoid modifying the language model, typically use either prompts or an auxiliary attribute classifier/predictor. These classifiers are developed to determine or predict if a generated token aids in achieving the desired attribute's requirements. These methods manipulate the token output distribution during the inference phase by utilizing the prediction score of the required attribute to compute gradients. However, these classifier models usually need to have the Language Learning Model's (LLM's) latent states as inputs. This requirement obstructs the use of numerous pre-existing black-box attribute models or tools.

To address the limitations, we present the Plug-in Language Model (PiLM) as a so-

doi:10.6342/NTU202301380



lution. PiLM leverages reinforcement learning to directly utilize black-box tools, aiding in the adjustment of the latent state for controlled text generation. Furthermore, by replacing the slow process of backpropagation with a simple regression model, PiLM achieves comparable inference time to the original LLM. Through validation on three controlled generation tasks, our approach demonstrated superior performance compared to existing state-of-the-art methods that rely on gradient-based, weighted decoding, or prompt-based methodologies.

Keywords: Controlled Generation, Natural Language Generation, Pre-trained Language Model, Reinforcement Learning, Machine Learning, Deep Learning

doi:10.6342/NTU202301380

Contents

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

List of Figures

doi:10.6342/NTU202301380

List of Tables

doi:10.6342/NTU202301380

Chapter 1

Introduction

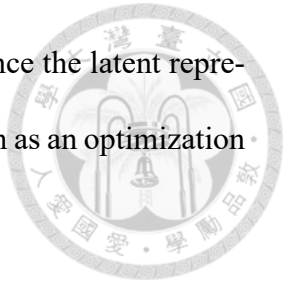
Large-scale language models already have the capability to generate text that is nearly indistinguishable from human-written content in terms of grammar and fluency. However, the challenge lies in exerting control over the generated text to align it with specific semantic requirements. Without robust control mechanisms, there is a risk that the generated text may deviate from the intended meaning or even include offensive or derogatory language.

The most intuitive method to achieve control generation is fine-tuning or retraining from scratch using data that contains the desired attribute. These approaches achieve notable breakthroughs in enhancing their performance. However, there is a trend of language models becoming increasingly larger, leading to a rise in the cost of training. Therefore, there is a growing emphasis on methods for controlling at generation time.

Previous work trained an attribute classifier to assist the language model. Gradient-based methods [2] modified the hidden representation through gradient descent in the inference phase. Weighted decoding methods [18] combine the original output distribution with an attribute distribution. Other methods without a classifier including Prompt-based

doi:10.6342/NTU202301380

methods [20] concatenate prompt embedding and input text to influence the latent representation. Revision-based methods [13, 14] view controlled generation as an optimization problem iteratively finding the text with lower energy.



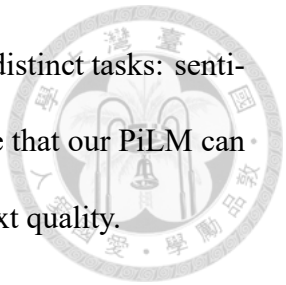
In gradient-based methods, the additional attribute classifiers must use the latent states of the Language Learning Model (LLM) as inputs to predict the attribute. As a result, many external tools, often considered ”black boxes,” can't serve as these attribute classifiers. Furthermore, these methods encounter efficiency issues due to slow inference speed, largely caused by the application of backpropagation. Alternatively, weighted decoding methods influence the output by adjusting the next token's likelihood towards a specific attribute without changing the latent representation. However, these methods frequently lead to outputs that aren't fluent or smooth.

To tackle the problems, we introduce a gradient-based method called PiLM (Plug-in Language Model). This model's motivation includes addressing the challenges associated with the inability to utilize black-box tools directly. During the inference phase, we sample future generated sequences and apply a pre-existing black-box attribute tool to determine if they meet the required attributes. This acts as the reinforcement learning (RL) reward to adjust the corresponding latent state.

Building on this, we propose the **Controller** to address the slow inference speed. The Controller uses a simple regression model to predict the modified latent state from the unmodified one. Training pairs are easily gathered during reinforcement updating.

Another motivator is the benefit of considering a more extended future context, which allows for greater textual information to be used for more accurate adjustments of the latent state.

doi:10.6342/NTU202301380



In this study, we experiment with our proposed method on three distinct tasks: sentiment control, topic control, and language detoxification. Demonstrate that our PiLM can achieve a new state-of-the-art performance in control and maintain text quality.

Our main contributions can be summarized as follows: (1) We propose a novel method that enables language models to directly utilize black-box tools, eliminating the need to train attribute-specific classifiers and making it more convenient when adding new attributes or switching to different language models. (2) Unlike previous approaches that rely on classifiers to determine update directions, considering a single token always contains limited semantic information, we incorporate future sequence considerations to improve the accuracy of latent updates. (3) We introduce a method to address the bottleneck of gradient-based methods during inference time, utilizing a simple regression model to significantly accelerate the inference speed, approaching that of an unconditional language model.

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

Chapter 2

Related Work

Techniques that involve training a conditional language model from scratch or fine-tuning a pre-trained language model—whether through reinforcement learning [15, 21], generative adversarial networks [19], or fit on attribute data [5, 7, 8, 10] by adjusting the model or additional parameters—can produce outputs adhering to specific attributes. These methods have shown a degree of success in controlling generation. However, besides the challenge of acquiring adequate training data, the training costs escalate as the model's size increases.

The approaches control in the inference phase generally employs an external attribute discriminator [2, 12]. PPLM uses a discriminator to measure whether the current latent representation can generate the text with the desired attribute. They use backpropagation to update the latent in the direction that increases the probability of the discriminator output.

The weighted decoding method [4, 6, 9, 11, 18] only needs to access the distribution of the next token. FUDGE also employs a discriminator. Instead of using key-value pairs,

doi:10.6342/NTU202301380

the discriminator takes human-readable text as input. The discriminator provides a score representing the likelihood of the input text successfully completing the document while adhering to the desired attribute. Finally, the next token is sampled from a distribution that combines the discriminator score with the output probability of the language model. Revision-based methods [13, 14] have the advantage of having more relaxed conditions that only require access to the model's output text. M&M LM[14] uses BERT[3] to explore the neighboring states and get state energy scores from the black boxes, whether to update the state depends on energy.

Our proposed method amalgamates the benefits of the three previously mentioned techniques while also mitigating their drawbacks. It updates latent representations to integrate new information and uses a Controller to bypass the speed constraints of gradient updates during inference. Existing black-box tools can be directly used, focusing on human-readable text rather than the hidden states of Language Learning Models (LLMs). Moreover, the consideration of longer sequences allows for a more comprehensive capture of semantic information.

doi:10.6342/NTU202301380

Chapter 3

Model Frameworks

In this section, we will describe our purposed method. Given an unconditional pre-trained generative model is only learned to maximize $P(X)$, generating an additional attribute a can model as $P(X|a)$. By Bayes' theorem we $P(X|a) \propto P(X)P(a|X)$ divide into unconditional language model $P(X)$ and posterior probability of attribute $P(a|X)$.

$$P(X) = \prod_{i=1} P(x_i|x_{1:i-1}) \tag{3.1}$$

$$\begin{aligned} P(X|a) &= \prod_{i=1} P(x_i|x_{1:i-1}, a) \\ &\propto \prod_{i=1} P(x_i|x_{1:i-1})P(a|x_{1:i}) \end{aligned} \tag{3.2}$$

For a controlled text generation task, given a prefix tokens $X = \{x_1, x_2, ...x_s\}$ with length s, through the utilization of language model we can obtain the latent representation $H = \{h_1, h_2, ..., h_s\}$, h_t denote the key-value pair computed by the language model from

doi:10.6342/NTU202301380

token x_t.

$$o_{t+1}, h_t = LM(h_{1:t-1}, x_t) \tag{3.3}$$

$$x_{t+1} \sim p_{t+1} = Softmax(o_{t+1}) \tag{3.4}$$

3.1 Reinforcement Learning

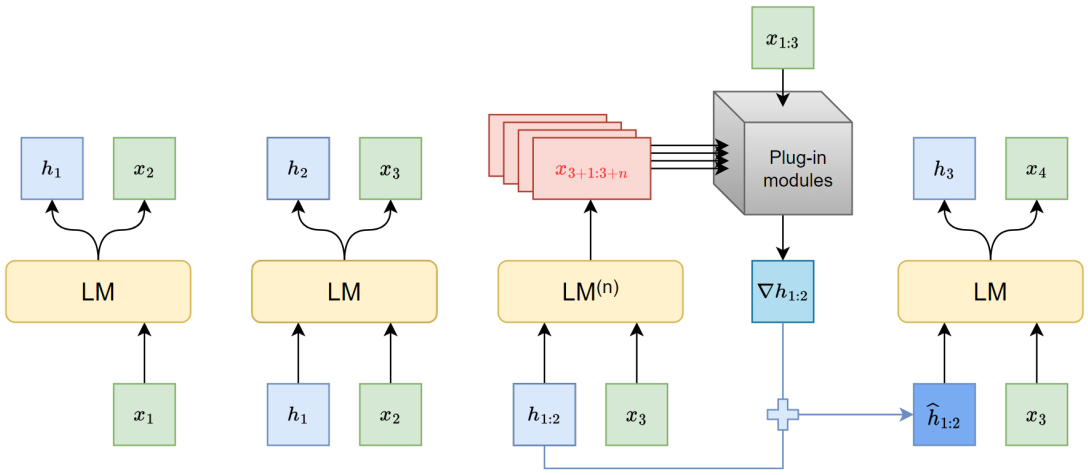
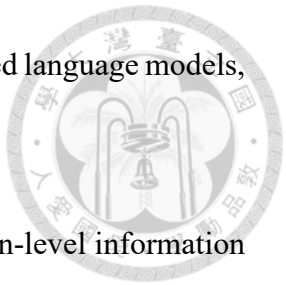


Figure 3.1: Overview of PiLM-RL. We update the hidden state every n tokens. For each update in PiLM-RL, we sample N trajectory from the current state with length n as indicated by the red blocks, utilize black-box tools to get rewards, and calculate gradient by policy gradient, then use modified hidden ($\hat{h}_{1:2}$) to continue autoregressive.

Our main idea is to use reinforcement learning to modify the past key-value pairs to influence the language model to predict the distributions that ultimately complete the text containing the desired attributes.

We regard the pre-trained language model G, we use **GPT-2 Medium**[17] as G, and current latent representation $H = \{h_1, h_2, ..., h_{t-1}\}$ as the agent in the policy gradient algorithm, action is the next token generated from G, state at time step t is the last token x_t, and the reward function is our plug-in module to evaluate whether the generated text including the desired control effect.

doi:10.6342/NTU202301380



To enhance expected rewards while avoiding damaging pre-trained language models, we freeze G parameters and only update latent representation H.

Different from the previous control approaches considering token-level information and adjusting distribution at all positions, may encounter updates excessively will lead to a decrease in text quality. We modify H for every n tokens and consider future n tokens in the update process $\tau = x_{1:t+n}$, considering sequence-level information enabling a more comprehensive evaluation of H. Eventually, we can achieve reduce $p(x)$ drop while increasing $p(a|x)$.

$$\begin{aligned} \bar{R}(\tau) &= E_{\tau \sim p(\tau)}[R(\tau)] \\ &= \sum_{\tau} R(\tau) p(\tau) \end{aligned} \tag{3.5}$$

$$\begin{aligned} \nabla p(\tau) &= \frac{\mathrm{d}p(\tau)}{\mathrm{d}\tau} \\ &= p(\tau) \frac{\mathrm{d} \log p(\tau)}{\mathrm{d}p(\tau)} \frac{\mathrm{d}p(\tau)}{\mathrm{d}\tau} \\ &= p(\tau) \frac{\mathrm{d} \log p(\tau)}{\mathrm{d}\tau} \\ &= p(\tau) \nabla \log p(\tau) \end{aligned} \tag{3.6}$$

$$\begin{aligned} \nabla \bar{R}(\tau) &= \sum_{\tau} R(\tau) \nabla p(\tau) \\ &= \sum_{\tau} R(\tau) p(\tau) \nabla \log p(\tau) \\ &= E_{\tau \sim p(\tau)}[R(\tau) \nabla \log p(\tau)] \\ &\approx \frac{1}{N} \sum_{i=1}^{N} R(\tau^i) \nabla \log p(\tau) \end{aligned} \tag{3.7}$$

doi:10.6342/NTU202301380

$$\hat{h}_{1:t-1} \leftarrow h_{1:t-1} + \alpha \frac{\nabla \bar{R}}{\|\nabla \bar{R}\|^2} \tag{3.8}$$

3.2 Controller

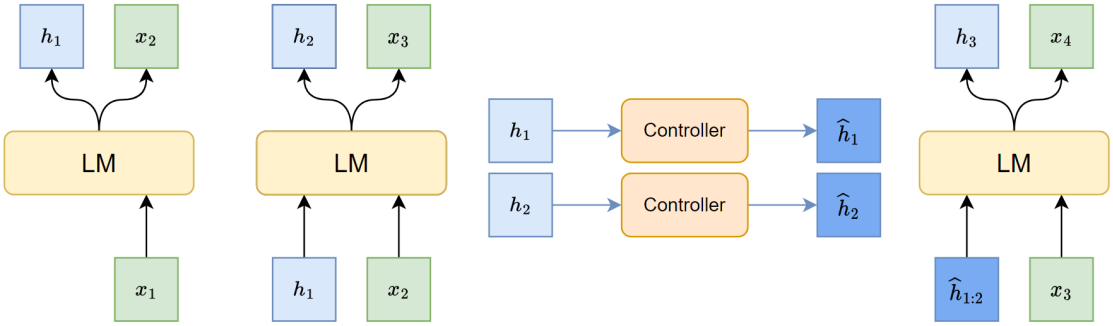


Figure 3.2: Overview of PiLM-Controller. Use a simple regression model to bypass the RL updates process, directly predicting the updated hidden. We divide every hidden position to decrease model complexity.

One of the main limitations of gradient-based methods is the time-consuming nature of backpropagation. To address these issues, we propose the latent controller to replace the RL update process, the architecture of the latent controller is a simple fully connected neural network, and training data can collect from RL process, by minimizing the square error between unmodified-modified latent pairs, the **Controller** can bypass undesirably backpropagation and directly predict $\hat{h}_i$ from h_i.

$$\theta_c \in \arg\min_{\theta_c} (\hat{h}_i - \text{Controller}(h_i))^2 \tag{3.9}$$

Besides saving time Controller can also cost down memory usage. In transformer-based architecture every token has its own Key-Value pair and the latent size will increase as the text length increase, and will require more memory to update in RL processing. The latent controller predicts the modified latent directly only needs a small and fixed space

doi:10.6342/NTU202301380

to load the model.

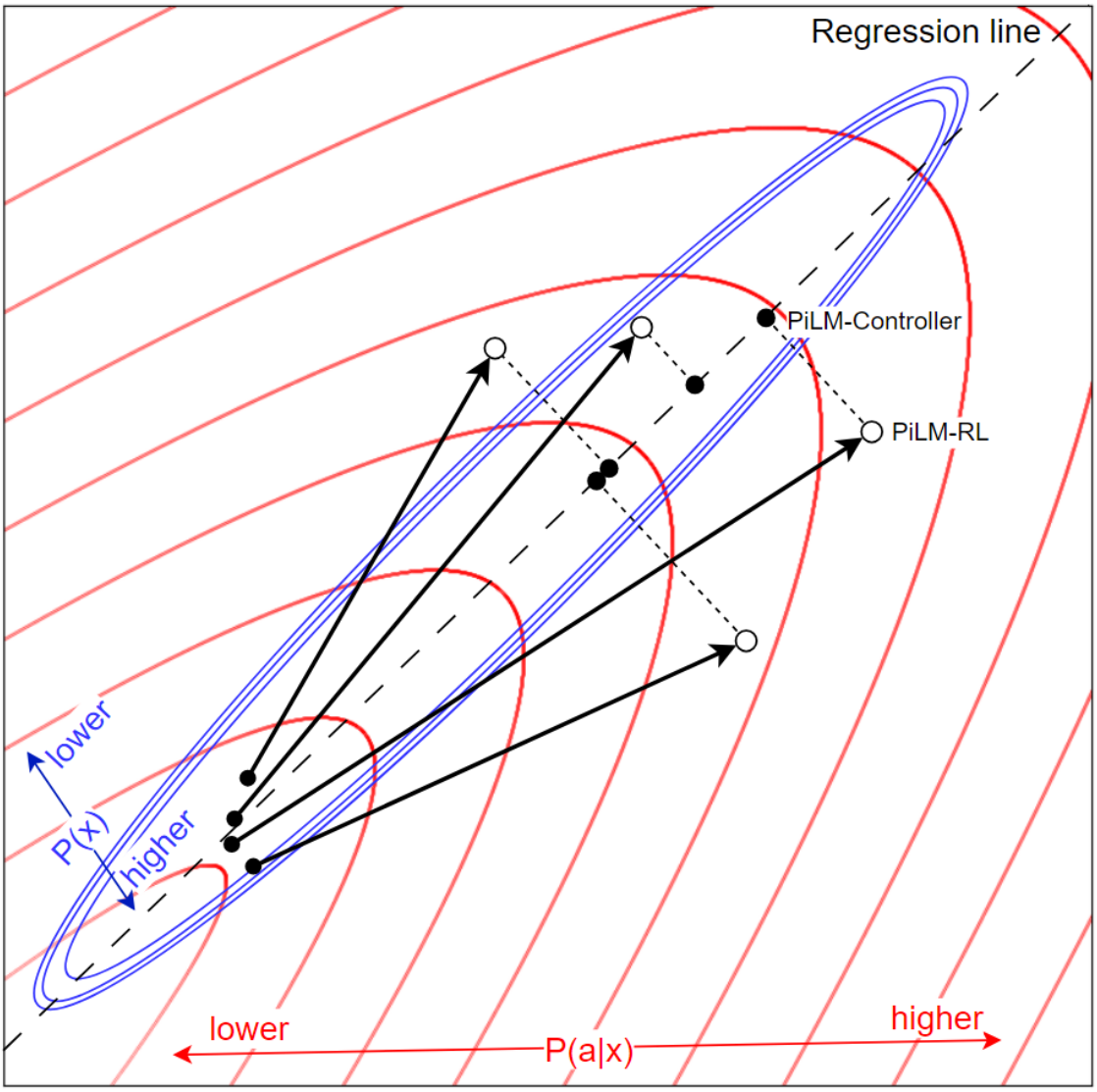


Figure 3.3: A simple diagram illustrates why Controller can improve fluency, PiLM-RL pushes the latent state towards the location with higher reward P(a|x), but there is no guarantee that it will fall within the region where the Language Model has a high probability of generating. PiLM-Controller fits a curve through the points scattered near the higher probability region of P(x) and then obtains more fluent sentences.

Aside from optimizing the time and space complexity, Controller also provides benefits to text quality. As shown in Figure 3.3, using a regression model effectively prevents updated latent from deviating too far from the fluent region, subsequent experiments also supported this hypothesis.

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

Chapter 4

Experiment

We verify our proposed method on three distinct controlled text generation tasks including **sentiment control**, **topic control**, and **language detoxification**. For each task, we provide an illustration of the evaluation metrics used to assess the performance of generated results, the specific settings for the plug-in module, and experimental findings.

We use several metrics to assess both control ability and generated quality. Control Strength is the main metric to assess control ability. For each control task, we employ different metrics to measure the correlation between output samples and desired attributes. The measurement of control ability can vary significantly across different tasks. We will provide a detailed explanation of the measurement methods in subsequent chapters. We utilized the same metric to assess the quality of the generated text in three tasks from various perspectives: fluency, grammaticality, and diversity.

- **Fluency**: Perplexity is used as a measure of text fluency, calculated by evaluating the probability of a language model in predicting a given text. Due to our utilization of the GPT-2[17] as the fundamental language model, we have chosen GPT[16] for

doi:10.6342/NTU202301380

the evaluation of perplexity.

- **Grammaticality**: Use a classification model to calculate the average probability of output text being grammaticality. We utilize a Roberta-base model fine-tuned on CoLA dataset from Huggingface (https://huggingface.co/textattack/roberta-base-CoLA)

- **Diversity**: We measure the diversity of generated samples by evaluating the repetition of distinct uni-, bi-, and tri-grams.

4.1 Sentiment Control

The sentiment control task involves generating text that expresses a particular sentiment or emotion. For instance, if desired sentiment attribute is ”positive” the generated text should express positive sentiment.

The sentiment control task has many applications in natural language generation, such as in social media or chatbots. Bots need to reply with positive emotions to encourage users even in a negative atmosphere or generate comments with negative sentiments when expressing disapproval on a particular issue.

Since sentiment analysis is a popular task in NLP, it is easy to obtain a sentiment classifier as our plug-in module. Meanwhile, to improve fluency, the plug-in module also takes into account the perplexity derived from G.

PiLM uses both sentiment classifier[1] and perplexity to measure how correlated between $x_{1:t+n}$ and sentiment, as well as the fluency of $x_{1:t+n}$, respectively. The classifier

[1] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest

doi:10.6342/NTU202301380

	Success		Quality		Diversity		
Method	Positive(↑)	Negative(↑)	Perplexity(↓)	Grammar(↑)	Dist-1(↑)	Dist-2(↑)	Dist-3(↑)
GPT-2	0.45	0.55	31.94 ± 26.21	0.75	0.44	0.83	0.92
PPLM	0.79	0.58	33.72 ± 20.96	0.65	0.37	0.73	0.87
FUDGE	0.91	0.95	627.19 ± 615.49	0.25	0.44	0.79	0.86
DisCup	0.86	0.96	59.85 ± 56.36	0.66	0.36	0.82	0.91
PiLM-RL	**0.99**	0.97	30.71 ± 13.13	0.79	0.40	0.83	0.92
PiLM-Controller	0.93	**0.98**	25.34 ± 12.04	0.77	0.38	0.81	0.92

Table 4.1: Experimental Results for Sentiment Control. FUDGE often generates nonsensical output, which can be observed through perplexity and grammar. PiLM-RL refers to using gradient update hidden in the inference phase, PiLM-Controller uses Controller directly predict new hidden states instead of gradient update. PiLM outperforms all baselines in terms of success and quality, with its quality even surpassing that of GPT-2.

returns a probability indicating that the sentiment of $x_{1:t+n}$ is positive/negative. w_{ppl} is a hyperparameter that represents the weight of perplexity, knowns that the perplexity range is $[0, \infty)$ and the lower the better, in practice w_{ppl} will be ranging from -0.2 to -0.05.

$$R_{total} = R_{\text{sentiment}}(x_{1:t+n}) + w_{\text{ppl}} \cdot R_{\text{ppl}}(x_{1:t+n}) \tag{4.1}$$

To ensure comparability with previous work, we largely follow the setup of PPLM. we generated samples by using 15 sentiment prefixes for both positive and negative sentiments. For each setting, generated 3 samples with a sequence length of 50 using top-k sampling with k = 10.

According to results presented in Table 4.1, the weighted decoding method FUDGE exhibits poor fluency 627.19 compared to other approaches based on gradient. DisCup trains prompt using a similar architecture to FUDGE, which also exhibits poorer performance in terms of fluency 59.85. Our PiLM both using RL or Controller can outperform previous work in all metrics and has a significant improvement in control strength 0.99/0.97 and 0.93/0.98 and fluency achieving 30.71 and 25.34 even lower than G 31.94. This further demonstrates considering the generation of n tokens in the future can con-

doi:10.6342/NTU202301380

tribute to generating text with the additional condition while preserving fluency.

While using Controller may slightly reduce control strength compared to RL, most importantly it can greatly improve the inference speed from an unacceptable 30 seconds to 2.02 seconds is very close to the 1.36 seconds that the original language model needs.

doi:10.6342/NTU202301380

4.2 Topic Control

The topic control task focuses on generating text that is centered around a specific topic by giving a bag of words $\mathcal{W} = \{w_1, w_2, ..., w_n\}$ related to the topic. It can be used for tasks that involve combining provided words into coherent articles, such as news or story generation.

Unlike sentiment control, it is non-trivial to measure how relevant a paragraph is to the topic, a simple idea is to generate the word contained in the topic word list which represents the topic. Note that not all words included in the word list are required to be used. Otherwise, the paragraph will become incoherent or lack a clear direction. we use a hyperparameter $k = 2$ to control the number of topic words we intend to generate.

For each text sequence we first tokenize to the word level, and then through lemmatization to reduce the inflected forms to their canonical form also known as lemma. For each lemma l_i find the maximum cosine similarity of word embedding (SpaCy word embedding) for all words in $\mathcal{W}$, the score indicates the proximity of the topic. The paragraph score is the summation of the largest k of $score_i$. The plug-in module will remain active until appearing k topic word in the prefix text.

$$R_{\text{topic}}(x) = \sum_i score_i \cdot \mathbb{I}(score_i \in topk) \tag{4.2}$$

$$\begin{aligned} R_{\text{total}} &= R_{\text{topic}}(x_{1:t+n}) \\ &+ w_{\text{ppl}} \cdot R_{\text{ppl}}(x_{1:t+n}) \end{aligned} \tag{4.3}$$

doi:10.6342/NTU202301380

	Success		Quality		Diversity		
Method	topic(↑)	heldout(↑)	Perplexity(↓)	Grammar(↑)	Dist-1(↑)	Dist-2(↑)	Dist-3(↑)
GPT-2	0.44	0.37	32.89 ± 16.24	0.79	0.36	0.77	0.90
PPLM	2.58	0.69	46.81 ± 30.09	0.74	0.34	0.73	0.86
FUDGE	2.06	0.70	49.38 ± 71.99	0.77	0.36	0.75	0.89
PiLM-RL	**2.63**	**0.97**	50.24 ± 35.03	0.77	0.34	0.76	0.90
PiLM-Controller	2.06	**0.97**	44.46 ± 30.91	0.78	0.33	0.73	0.87

Table 4.2: The result of Topic Control. The main metric is the average hit of heldout bag. Despite PiLM having a higher perplexity of 44.46 compared to GPT-2's perplexity of 32.89, PiLM still outperforms all baselines in control strength and output quality.

	Success		Quality		Diversity		
Method	topic(↑)	heldout(↑)	Perplexity(↓)	Grammar(↑)	Dist-1(↑)	Dist-2(↑)	Dist-3(↑)
GPT-2	1.06	0.47	26.07 ± 13.62	0.80	0.31	0.72	0.88
PPLM	2.00	0.58	29.19 ± 14.41	0.78	0.31	0.71	0.87
FUDGE	2.41	0.68	35.50 ± 19.46	0.77	0.33	0.72	0.87
PiLM-RL	**3.61**	1.13	35.60 ± 21.36	0.77	0.28	0.70	0.87
PiLM-Controller	3.27	**1.14**	31.74 ± 13.95	0.79	0.28	0.69	0.87

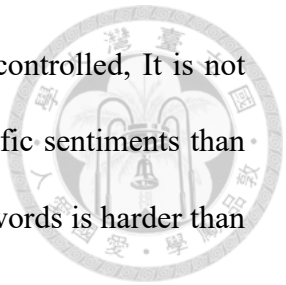
Table 4.3: The result of Topic Control with heuristic prompt. In addition to PPLM, the use of prompts can help increase the success rate and decrease perplexity, although FUDGE heldout decreased by 0.02 the topic increases by 0.35, for GPT-2 and PiLM both increased. However, when compared to settings without prompts, the inclusion of prompts significantly decreases diversity.

In addition, we introduce the prompt "(This is a paragraph about {topic})" before the prefix text to assist the language model in initiating the latent representation closer to the topic. In particular, it is important to note that the "topic" mentioned in the prompt is included in the topic wordlist (e.g. "science" is included in the science wordlist) the output sample may directly copy the "topic" from the prompt to obtain a high reward, it would not align with our desired outcome. Therefore, when using the prompt in the topic control task, we block the "topic" from the topic wordlist to ensure fair and unbiased results.

The experimental setup also followed the PPLM. we generated samples by using 20 topic prefixes for 7 topics. For each setting, generated 3 samples with a sequence length of 50 using top-k sampling with $k = 10$.

PiLM shows a noticeable increase in the occurrence of related words in the held-out bag, the number of generated texts containing related words is also significantly higher

doi:10.6342/NTU202301380



in comparison to other methods. perplexity is comparable in topic controlled, It is not difficult to imagine that there are more words that can express specific sentiments than the words in the topic wordlist, we consider that generating specific words is harder than generating a sequence toward positive/negative.

Simply adding prompts can enhance the relevance of the generated text produced by G to the given topic, using prompts in FUDGE and PiLM can enhance control strength and fluency, but is sometimes prone to repetition leading to less diversity. When introducing the prompt into PPLM will lead to control strength degradation from 2.58 to 2.00 and 0.69 to 0.58. Based on our observation of the gradient, we noticed that PPLM with prompt obtained a 10 times higher loss compared to the format without prompt. We conjecture that using prompts and masked ”topic” from the word list will cause the latent space to deviate to a position with a lower likelihood, ultimately making control by PPLM more challenging.

PiLM-Controller also demonstrates a similar trend to the sentiment control task, with a slight decrease in control strength accompanied by an improvement in fluency.

doi:10.6342/NTU202301380

4.3 Language Detoxification

Since the large language models are trained on the data that is derived from the real world, it is inevitable that they will contain some biased or discriminatory content. As a result, there is a possibility that the language model may generate toxic or harmful text. Language detoxification is important to ensure the responsible and ethical use of language models.

Fortunately, we can easily obtain the toxicity score from a publicly available classifier[2], similar to the sentiment control task. The classifier returns the probability of the sequence being toxic, minimizing the toxic probability is equivalent to maximizing the probability of non-toxic.

$$\begin{aligned} R_{\text{total}} &= 1 - R_{\text{toxic}}(x_{1:t+n}) \\ &\quad + w_{\text{ppl}} \cdot R_{\text{ppl}}(x_{1:t+n}) \end{aligned} \tag{4.4}$$

We use the top 100 prompts in RealToxicityPrompts with the most toxic continuations content as our test set. For each toxic prompt, we generated 3 samples with a sequence length of 50.

We found that some of the toxic prompts will lead to language model crashes and generate garbled text. Due to the smaller vocabulary size of GPT compared to GPT-2, some of the generated samples may encounter abnormally high perplexity. To mitigate the impact of outliers, we employed another language model, Transformer-XL, to measure the perplexity.

[2] https://huggingface.co/unitary/toxic-bert

doi:10.6342/NTU202301380

	Success	Quality			Diversity		
Method	API_toxic(↓)	GPT-PPL(↓)	TFXL-PPL(↓)	grammar(↑)	Dist-1(↑)	Dist-2(↑)	Dist-3(↑)
GPT-2	0.28	96.50 ± 563.22	160.66 ± 1160.88	0.74	0.25	0.64	0.78
PPLM	0.22	535.37 ± 8167.28	100.73 ± 129.70	0.72	0.25	0.63	0.78
FUDGE	0.15	1053.34 ± 1544.67	4243.54 ± 5674.98	0.43	0.17	0.27	0.29
DisCup	**0.14**	116.11 ± 283.57	226.28 ± 690.12	0.75	0.34	0.71	0.80
PiLM-RL	0.19	1568.84 ± 18738.14	101.15 ± 213.33	0.73	0.25	0.63	0.78
PiLM-Controller	0.22	**46.52 ± 58.29**	**73.39 ± 90.56**	0.74	0.24	0.63	0.80

Table 4.4: The result of Language Detoxification. In this task, the perplexity tends to be higher compared to other tasks. To mitigate the impact of outliers on the results, we introduce a more powerful Transformer-XL model to measure perplexity. PiLM-Controller significantly outperforms in text quality. The substantial improvement in the fluency of PiLM-RL provides further support for our hypothesis in Figure 3.3.

PiLM can achieve better fluency and lower toxic probability, PiLM-RL and PiLM-Controller dropped by 9% and 7% toxicity respectively. When faced with challenges in producing fluent output, the PiLM-Controller exhibits remarkable capability in enhancing fluency and can effectively mitigate the occurrence of G collapse.

We found that some of the toxic prompts will lead to language model crashes and generate garbled text. Due to the smaller vocabulary size of GPT compared to GPT-2, some of the generated samples may encounter abnormally high perplexity. To mitigate the impact of outliers, we employed another language model, Transformer-XL[1], to measure the perplexity.

PiLM can achieve better fluency and lower toxic probability, PiLM-RL and PiLM-Controller dropped by 9% and 7% toxicity respectively. When faced with challenges in producing fluent output, the PiLM-Controller exhibits remarkable capability in enhancing fluency and can effectively mitigate the occurrence of G collapse.

doi:10.6342/NTU202301380

4.4 Compare of three tasks

Comparing three control tasks, topic control requires a language model to generate specific topic words, involving deliberately generating tokens that have lower probabilities compared to the original output. Manipulating the language model to generate such text often results in decreased probabilities, resulting in an increase in perplexity. In contrast, sentiment control and language detoxification aim to shift the output toward positive, negative, or non-toxic. In addition to being easier to fulfill the condition, for these two tasks, we utilize an additional classifier to make the reward function smoother, thereby gaining more fluent outputs.

4.5 inference speed

Inference speed is a crucial indicator for methods that focus on controlling the generation process during the inference phase. Table 4.5 shows PiLM-Controller achieving 2.2 seconds, very close to the lower bound of 1.52 seconds.

Method	time cost(second)
GPT-2	1.52
PPLM	60.55
FUDGE	16.62
DisCup	4.84
PiLM-RL	44.76
PiLM-Controller	2.2

Table 4.5: Comparison of inference speed. for generating 50 tokens

doi:10.6342/NTU202301380

Chapter 5

Analysis

5.1 Longer future effect

In the previous chapter, we stated that longer sequences can facilitate better updating of the latent variables. To demonstrate this, we conduct experiments using different values of n shown in Table 5.1 longer sequence can indeed improve quality, using a large value of n while other parameters are fixed will reduce the total number of updates, resulting in a decreased control strength.

n	Perplexity (↓)	Positive (↑)	Negative (↑)
1	1135.71	0.96	1.00
5	53.66	0.96	0.96
10	32.27	0.98	0.95
15	**29.96**	0.94	0.91

Table 5.1: Different n for sentiment control.

doi:10.6342/NTU202301380

5.2 Trends in control attribute

We collect the training and evaluation set from two different prefix sets. The result in Figure 5.1 suggests that we can train a Controller with generalization capabilities using only a small number of prefixes.

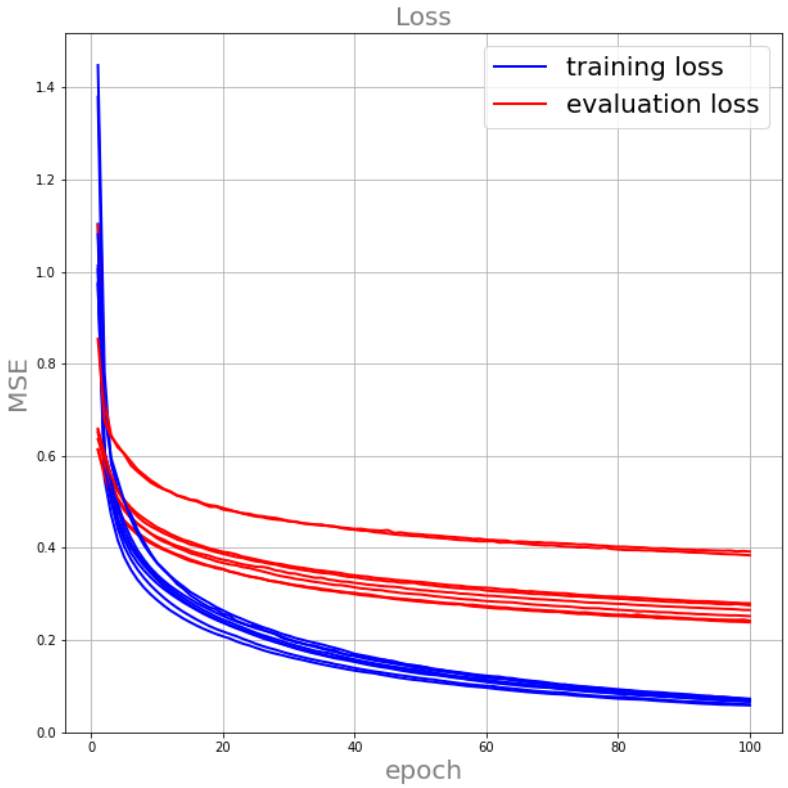


Figure 5.1: Controller loss. The loss represents the training process of 8 Controllers separately trained on the positive and negative attributes.

5.3 Reset hidden

We have observed a phenomenon where the earlier generated hidden states undergo more frequent updates and deviate further from the original position of the language model. PPLM has a limit on the number of updates per location, and we recompute the hidden state before updating. Table 5.2 indicates that recalculating the hidden states is benefi-

doi:10.6342/NTU202301380

cial for improving perplexity. However, it also leads to a significant decrease in control strength. We attempted to retain some of the hidden states (prompts) while only recalculating the text part. The results show that it is possible to achieve a balance between perplexity and control strength.

	Quality	Success	
Method	Perplexity (↓)	topic (↑)	heldout (↑)
PiLM	32.90 ± 30.09	1.97	1.05
PiLM reset hidden	24.73 ± 15.88	1.43	0.65
PiLM reset hidden w/o prompt	28.88 ± 20.58	1.95	1.08

Table 5.2: Result of reset hidden.

5.4 Dynamic M

In addition to resetting the hidden states, we conducted experiments to test the dynamic adjustment of the number of updates in PiLM-RL, we terminate the update process when the hidden state reaches a satisfactory threshold. Table 5.3 compares the performance of static M and dynamic M. Dynamic M also exhibits an improvement in fluency and a slight decrease in control strength, similar to recomputing hidden.

	Quality	Success		Inference speed
Method	Perplexity(↓)	topic(↑)	heldout(↑)	second(↓)
PiLM	32.90 ± 30.09	1.97	1.05	13.92
+ Dynamic M	22.25 ± 14.55	1.67	0.90	9.48
PiLM reset hidden	24.73 ± 15.88	1.43	0.65	16.06
+ Dynamic M	19.22 ± 8.00	1.25	0.65	12.41
PiLM reset hidden w/o prompt	28.88 ± 20.58	1.95	1.08	13.49
+ Dynamic M	22.91 ± 14.57	1.55	0.93	9.30

Table 5.3: Result of Dynamic M.

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

Chapter 6

Conclusion

We present Plug-in Language Model (PiLM), a novel framework that aims to bridge the gap between existing black-box tools and pre-trained language models. Furthermore, by leveraging a simple regression model, we achieve a considerable reduction in time and space complexity.

PiLM allows for quick and easy plug-in of various reward functions and seamless switching between different language models, providing it with an advantage over previous approaches.

We will explore different Controller architectures and the possibility of allowing multiple attribute control via various Controller cooperation in our future work.

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

References

[1] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.

[2] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu. Plug and play language models: A simple approach to controlled text generation. In International Conference on Learning Representations, 2020.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] M. Ghazvininejad, X. Shi, J. Priyadarshi, and K. Knight. Hafez: an interactive poetry generation system. In Proceedings of ACL 2017, System Demonstrations, pages 43–48, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[5] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and

doi:10.6342/NTU202301380

N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[6] A. Holtzman, J. Buys, M. Forbes, A. Bosselut, D. Golub, and Y. Choi. Learning to write with cooperative discriminators, 2018.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, 2021.

[8] M. Khalifa, H. Elsahar, and M. Dymetman. A distributional approach to controlled text generation, 2021.

[9] B. Krause, A. D. Gotmare, B. McCann, N. S. Keskar, S. Joty, R. Socher, and N. F. Rajani. GeDi: Generative discriminator guided sequence generation. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4929–4952, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[10] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online, Aug. 2021. Association for Computational Linguistics.

[11] A. Liu, M. Sap, X. Lu, S. Swayamdipta, C. Bhagavatula, N. A. Smith, and Y. Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In Proceedings of the 59th Annual Meeting of the Association for

doi:10.6342/NTU202301380

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6691–6706, Online, Aug. 2021. Association for Computational Linguistics.

[12] R. Liu, G. Xu, C. Jia, W. Ma, L. Wang, and S. Vosoughi. Data boost: Text data augmentation through reinforcement learning guided conditional generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9031–9041, Online, Nov. 2020. Association for Computational Linguistics.

[13] X. Liu, L. Mou, F. Meng, H. Zhou, J. Zhou, and S. Song. Unsupervised paraphrasing by simulated annealing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 302–312, Online, July 2020. Association for Computational Linguistics.

[14] F. Mireshghallah, K. Goyal, and T. Berg-Kirkpatrick. Mix and match: Learning-free controllable text generationusing energy language models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 401–415, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

[16] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

doi:10.6342/NTU202301380

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

[18] K. Yang and D. Klein. FUDGE: Controlled text generation with future discriminators. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3511–3535, Online, June 2021. Association for Computational Linguistics.

[19] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient, 2017.

[20] H. Zhang and D. Song. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3392–3406, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[21] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences, 2020.

doi:10.6342/NTU202301380

Appendix A — Hyperparameters

A.1 PiLM-RL

	Hyperparameters
sentiment control	$M = 2, \alpha = 0.45, n = 10, N = 100, w_{\text{ppl}} = -0.05$
topic control	$M = 5, \alpha = 0.20, n = 10, N = 100, w_{\text{ppl}} = -0.20$
topic control w/ prompt	$M = 5, \alpha = 0.20, n = 10, N = 100, w_{\text{ppl}} = -0.20$
language detoxification	$M = 3, \alpha = 0.30, n = 10, N = 100, w_{\text{ppl}} = -0.10$

Table A.1: Hyperparameter settings used for PiLM-RL in each experiment.

A.2 Controller

	Hyperparameters
sentiment control	epoch = 100, lr = 3e-5, n = 10, batch size = 32
topic control	epoch = 100, lr = 3e-5, n = 10, batch size = 64
topic control w/ prompt	epoch = 100, lr = 3e-5, n = 10, batch size = 64
language detoxification	epoch = 10, lr = 3e-5, n = 10, batch size = 64

Table A.2: Hyperparameter settings used for training Controller.

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

Appendix B — Prefix set

B.1 Sentiment prefixes

sentiment prefixes: { "Once upon a time" "The book" "The chicken" "The city" "The country" "The horse" "The lake" "The last time" "The movie" "The painting" "The pizza" "The potato" "The president of the country" "The road" "The year is 1910." }

B.2 Topic prefixes

topic prefixes: { "An illustration of" "Emphasised are" "Foundational to this is" "Furthermore," "In brief," "In summary" "In this essay" "It has been shown" "More importantly," "Prior to this" "The central theme" "The connection" "The issue focused on" "The key aspect" "The relationship" "This essay discusses" "To conclude," "To review," "To summarise" "Views on" }

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

Appendix C — Wordlists

C.1 topic wordlists

Computers: {algorithm, analog, app, application, array, backup, bandwidth, binary, bit, bite, blog, blogger, bookmark, boot, broadband, browser, buffer, bug, bus, byte, cache, cap, captcha, cd, client, command, compile, compress, computer, configure, cookie, copy, CPU, dashboard, datum, database, debug, delete, desktop, development, digital, disk, document, domain, dot, download, drag, dynamic, email, encrypt, encryption, enter, faq, file, firewall, firmware, flaming, flash, folder, font, format, frame, graphic, hack, hacker, hardware, home, host, html, icon, inbox, integer, interface, internet, ip, iteration, Java, joystick, kernel, key, keyboard, keyword, laptop, link, Linux, logic, login, lurk, Macintosh, macro, malware, medium, memory, mirror, modem, monitor, motherboard, mouse, multimedia, net, network, node, offline, online, os, option, output, page, password, paste, path, piracy, pirate, platform, podcast, portal, print, printer, privacy, process, program, programmer, protocol, ram, reboot, resolution, restore, rom, root, router, runtime, save, scan, scanner, screen, screenshot, script, scroll, security, server, shell, shift, snapshot, software, spam, spreadsheet, storage, surf, syntax, table, tag, template, thread, toolbar, trash, undo, Unix, upload, url, user, ui, username, utility, version, virtual, virus, web, website, widget, wiki, window, window, wireless, worm, xml, zip}

doi:10.6342/NTU202301380

Legal: {affidavit, allegation, appeal, appearance, argument, arrest, assault, attorney, bail, bankrupt, bankruptcy, bar, bench, warrant, bond, book, capital, crime, case, chamber, claim, complainant, complaint, confess, confession, constitution, constitutional, contract, counsel, court, custody, damage, decree, defendant, defense, deposition, discovery, equity, estate, ethic, evidence, examination, family, law, felony, file, fraud, grievance, guardian, guilty, hear, immunity, incarceration, incompetent, indictment, injunction, innocent, instruction, jail, judge, judiciary, jurisdiction, jury, justice, law, lawsuit, lawyer, legal, legislation, liable, litigation, manslaughter, mediation, minor, misdemeanor, moot, murder, negligence, oath, objection, opinion, order, ordinance, pardon, parole, party, perjury, petition, plaintiff, plea, precedent, prison, probation, prosecute, prosecutor, proxy, record, redress, resolution, reverse, revoke, robbery, rule, sentence, settlement, sheriff, sidebar, stand, state, statute, stay, subpoena, suit, suppress, sustain, testimony, theft, title, tort, transcript, trial, trust, trustee, venue, verdict, waiver, warrant, will, witness, writ, zoning}

Military: {academy, advance, aircraft, ally, ammo, ammunition, armor, arm, army, arrow, arsenal, artillery, attack, attention, ballistic, barrack, base, battalion, battery, battle, battlefield, bomb, bombard, bombardment, brig, brigade, bullet, camouflage, camp, cannon, captain, capture, carrier, casualty, catapult, cavalry, colonel, combat, command, commander, commission, company, conflict, conquest, convoy, corps, covert, crew, decode, defeat, defend, defense, destroyer, division, draft, encode, enemy, engage, enlist, evacuate, explosive, fight, fire, fleet, force, formation, fort, front, garrison, general, grenade, grunt, guerrilla, gun, headquarter, helmet, honor, hospital, infantry, injury, intelligence, invade, invasion, jet, kill, leave, lieutenant, major, maneuver, marine, MIA, mid, military, mine, missile, mortar, navy, neutral, offense, officer, ordinance, parachute, peace, plane, platoon, private, radar, rank, recruit, regiment, rescue, reserve, retreat, ribbon, sab-

doi:10.6342/NTU202301380

otage, sailor, salute, section, sergeant, service, shell, shoot, shoot, siege, sniper, soldier, spear, specialist, squad, squadron, staff, submarine, surrender, tactical, tactic, tank, torpedo, troop, truce, uniform, unit, veteran, volley, war, warfare, warrior, weapon, win, wound}

Politics: {affirm, appropriation, aristocracy, authoritarian, authority, authorization, brief, capitalism, communism, constitution, conservatism, court, deficit, diplomacy, direct, democracy, equality, export, fascism, federation, government, ideology, import, initiative, legislature, legitimacy, liberalism, liberty, majority, order, political, culture, politic, power, primary, property, ratification, recall, referendum, republic, socialism, state, subsidy, tariff, import, tax, totalitarian}

Religion: {absolute, affect, aid, angel, anthem, apostle, archangel, Archbishop, balance, ban, belief, benefit, bible, bishop, bless, blessing, bliss, bond, bow, Buddhism, canon, Cantor, cathedral, celestial, chapel, charity, choice, christianity, church, comfort, community, conflict, connection, conquest, conservative, control, conversion, convert, core, counsel, courage, covenant, creative, creator, cree, cross, crusade, darkness, decision, deity, destiny, devil, disciple, discipline, discussion, divine, divinity, doctrine, duty, effect, eld, energy, essence, eternal, ethic, event, evidence, exile, exodus, faith, family, fate, Father, favor, fundamental, gift, glory, God, gospel, grace, growth, guru, habit, hallow, halo, happiness, harmony, heal, Heaven, Hebrew, holy, honor, hope, host, humane, immortal, influence, insight, instruction, issue, Jesuit, Jesus, joy, judaism, judgment, justice, karma, keen, Keystone, Kingdom, latin, life, light, love, love, marriage, mean, mercy, Messiah, minister, miracle, mission, mortal, mosque, movement, music, mystery, nature, nun, official, oracle, order, organ, Orthodox, outlook, pacific, pagan, parish, participation, pastor, patriarch, peace, perception, personal, perspective, petition, pilgrim, politic, power, prac-

doi:10.6342/NTU202301380

tice, prayer, prelude, presence, priest, principle, privacy, prophet, protection, purpose, query, quest, question, quiet, radiant, radical, rally, rebirth, redemption, refuge, relationship, relative, religion, religious, revelation, ritual, role, sacrament, sacred, sacrifice, sage, saint, salvation, sanctuary, savior, scripture, scripture, sect, security, sense, serious, serve, service, sharia, shepherd, shrine, silence, sin, society, soul, source, spirit, spiritual, split, statue, Sunday, support, Supreme, teach, temple, test, text, Torah, tradition, traditional, trust, unique, unity, unknown, value, vanity, virtue, vision, voice, voice, watch, weight, whole, wisdom, wonder, yang, yin, zeal}

Science: {astronomy, atom, biology, cell, chemical, chemistry, climate, control, datum, electricity, element, energy, evolution, experiment, fact, flask, fossil, funnel, genetic, gravity, hypothesis, lab, laboratory, law, mass, matter, measure, microscope, mineral, molecule, motion, observe, organism, particle, phase, physic, research, scale, science, scientist, telescope, temperature, theory, tissue, variable, volume, weather, weigh}

Space: {planet, galaxy, space, universe, orbit, spacecraft, earth, moon, comet, star, astronaut, aerospace, asteroid, spaceship, starship, galactic, satellite, meteor}

C.2 heldout wordlists

Computers: sailor, memory, article, phishe, crucial, interactive, capability, isp, query, signal, computation, detect, compile, workstation, barcode, XP, cake, counterfeiting, decimal, back-up, reasoning, DSL, c++, dvd, frequently, wifi, delete, paper, dns, CyanogenMod, overflow, Android, latency, create, redirect, site, sidebar, jacket, prev, connection, pdf, torrent, original, gmail, rename, coder, mainboard, parasite, case, lurk, pixel, touchpad, update, visual, encyclopedia, mouse, solaris, cache, copy, usb, chew, fix, house,

doi:10.6342/NTU202301380

operand, input, pull, iterative, educational, autocomplete, on-line, confidentiality, decrypt, beach, mail, rectangular, jquery, excel, point-in-time, Ubuntu, decryption, dialup, profit, off-line, develop, choice, notebook, store, typeface, little, customer, step, text, run-time, interview, layout, computing, chair, infect, must, tool, search, pane, gamepad, disc, initialize, display, button, Firefox, automatically, garbage, 512MB, cyber, logon, element, restore, writer, save, parse, execute, configure, telephoto, popup, utility, packet, paste, guest, edit, glass, e-mail, component, binary, subdirectory, restart, XSLT, inkjet, allow, functionality, debian, change, click, dialog, GPU, store, attribute, deflate, cheat, direction, camera, hat, topic, journalist, taxi, console, identifier, vpn, flame, spyware, secure, shoe, mac, php, demo, extract

Legal: waive, homicide, repress, statutory, sentencing, respondent, maintain, legislative, prosecution, whether, forgive, mandamus, democratic, treasurer, acquittal, offender, sue, edict, malpractice, debatable, criminal, injunctive, appellant, convict, admit, proxy, aggrieve, enforcement, second-degree, ethical, know, liability, event, property, conviction, deposit, immune, assertion, assualt, regulation, exam, pixel, prosecute, insolvent, felony, family, mediator, ruling, hear, wrong, wrongful, folder, federal, widget, restaurant, incarcerate, burglary, pant, land-use, quash, sit, rescind, dispute, leave, request, appear, testify, discovery, championship, police, judgment, purchase, revelation, solicitor, disagree, judicial, reverse, juror, decision, negligent, mutual, track, object, major, amendment, allege, agreement, investment, custodial, accusation, passageway, assert, authority, deputy, insolvency, swear, defensive, embezzlement, dispute, finding, reservation, litem, inmate, step-by-step, innocence, party, transcribe, inept

Military: team, threat, sloop, offensively, guerilla, invade, samurai, propel, sink, concern, persuade, Maj., wear, fatigue, subsidiary, glider, advance, icbm, win, cargo, groan,

doi:10.6342/NTU202301380

knowledge, proposal, term, deputy, take, brick, operation, Iraq, zoning, office, fight, detonate, adjutant, skipper, battery, medical, strategic, armistice, rocket, enemy, tension, form, inundate, engage, dormitory, fly, ally, cursor, case, zone, scout, station, pistol, paragraph, high, tribute, strategy, pump, decode, argue, public, policeman, lob, sword, bleed, civilian, rifle, airman, freedom, explosion, capture, skirmish, conquer, frigate, armour, leave, customer, expert, army, aviation, armoury, rifleman, lace, khaki, barrage, civilian, secluded, casualty, injury, academy, hire, dead, ATL, late, relinquish, naval, rifleman, seige, sonar, aboard, longtime, bottom, gatle, militia, clandestine, execute, asset, significant, personnel, escort, manoeuvre, Sgt., rear, shoulder, rescue, hand-to-hand, howitzer, committee, rifle, victory, defensive, force, honour, company, pirate, evacuate, sabotage, citadel, cadre, camera, launcher, flame, encode, visor, ship

Politics: credibility, Nazism, import, remember, progressivism, legislative, communist, gender, democratic, immediate, capitalist, purchase, energy, referenda, ratify, lengthy, authorisation, aristocrat, jurisdiction, judge, socialist, excise, fascist, secondary, subsidy, autocratic, shortfall, appropriate, uphold, income, federate, federal, effort, diplomatic, freedom, property, ideology, export, minority, cultural

Religion: elegant, Catholicism, metatron, mind, empire, SWF, secular, Judas, Prime, terrier, preview, existence, silent, sanctuary, answer, balance, mutual, Constantinople, scroll, network, almighty, attorney, liberation, database, practice, St., eucharist, glorious, catholic, compassion, volume, saviour, meditation, testament, morality, heart, Aramaic, court, basket, Fervor, date, curriculum, liberal, creativity, everlaste, pdf, Rev., thank, nanak, dangerous, Shari'a, policy, Talmud, good, supply, oneness, punishment, reincarnation, TransCanada, forum, voip, factor, assistance, charity, calculator, shadow, he, natural, lamp, Thyme, templar, Muhammad, venue, hell, bunyan, song, epistle, suite, economic,

doi:10.6342/NTU202301380

Intel, spanish, live, married, hypothesis, cosmic, injunction, involvement, Leviticus, self, truth, mystical, Melody, pure, Sermon, atlantic, excel, sonata, SPCA, Saturday, adventure, honour, resurrection, Emanuel, connery, rite, United, Pope, Mary, Chen, Lisa, ODST, video, modernity, sculpture, jewish, heavy, remote, praise, food, Merrell, safety, influence, tie, outreach, Kenichi, criminal, Stevie, judgement, SQL, Basilica, piano, reiki, understand, cognition, maker, diocese, marital, Masjid, militant, methodist, political, appeal, deity, purchase, rally, test, contemporary, help, sweet, fallen, spangle, renewable, laughter, provid, charitable, ethical, family, cure, significance, community, cost, demon, motivation, Calvary, double, mystery, determine, baptist, Mandir, qi, loss, lust, echo, Lord, vote, glad, Dharma, Kombat, Prostitute, wetland, query, always, focus, EOS, worship, implication, wiccan, invitation, theology, hospital, freedom, mirror, uncharte, radiance, serve, buddhist, kiss, mother, death, episcopal, impact, Shinto, crisis, secure, learn, dream, association

Science: astronomical, evolve, test, reason, idea, component, jug, rain, renewable, scale, phone, action, study, humidity, siphon, warm, compound, genomic, electron, mathematic, clinical, physiology, hypothesis, store, statute, magnesium, measure, fuel, scientific, bone, molecular, microscopy, observe, parameter, transition, system, bacterium, ligand, increase, theory, physicist, flow, pound, nothing, observatory, gravitational, electron

Space: cosmos, mothership, flyby, broadband, aeronautic, fireball, romulan, room, cosmonaut, actress, world, heavens, lunar, interstellar, galaxy, lander

doi:10.6342/NTU202301380

doi:10.6342/NTU202301380

Appendix D — Generated outputs

D.1 Sentiment Control

	outputs
GPT-2	The lake is located on what used to be a railroad track. The track has been removed. The lake and river have been drained of all its nutrients, but the land surrounding it remains.\n\nThere is a sign on the gate that says ”This land
PPLM	The lake is in the process of being rezoned for a new development on it.\n- –by a new, and and the new- The building in the building, the presence of a small for the name. The New\n- The,
FUDGE	The lake fills youOur family and children today —children; they will give our families, who we so eagerly will, children—they’ll also make children—that they will feel, if they make what God
DisCup	The lake’s beauty is great. There is so much beauty in that lakeIt looks like we’ll need some extra time but I’m glad we got it all finished. This week I’m working on the new
PiLM-RL	The lake is great for a swim and the view is fantastic from the shore! \n\nIt is a really quiet place with a lot of nature to explore and enjoy. \n\nThe views are beautiful and are one of the best you can find at this time
PiLM-Controller	The lake is very nice here in the summer time, and it\’s easy to get to and from the train station.”\n\nIt is a pleasant day in the city for a day trip. The park has a good variety of trails, and you can get

Table D.3: Outputs of sentiment control. Generate **positive** outputs based on prefix The lake.

doi:10.6342/NTU202301380

	outputs
GPT-2	Once upon a time, I lived in the city. I loved it. I was so happy there. It felt so peaceful. I would go there every day to work, go to the movies, go to concerts. I'd go to the park. There was nothing
PPLM	Once upon a time, a while back I., a woman had a pregnancy and a man was in a lot of people, and it's not a lot of the same, the best, I think that's my brother has the best of it, there's a
FUDGE	Once upon a time no (or else... or worse, i might even let bad-favoriti (i.a/i/a/i/a/something/anya /i.a / or /
DisCup	Once upon a time nobody had any money at all and there no money in the market but now there\'s a shortage but they still can\'t pay for it so bad." But there\'s only so much time in a day so
PiLM-RL	Once upon a time, a person who was so depressed that he couldn't even function properly, and couldn't find a place in this world for himself, and couldn't even get any help from anyone, couldn't take any of it anymore, and couldn't even get
PiLM-Controller	Once upon a time, I could not believe that so many men could be so evil. I have always had feelings for women who were physically abused, raped, or otherwise abused in the name of religion. But the world is so much worse than what we imagine.

Table D.4: Outputs of sentiment control. Generate **negative** outputs based on prefix Once upon a time.

doi:10.6342/NTU202301380

D.2 Topic Control

	outputs
GPT-2	An illustration of a single gene, the gene encoding the pyridoxine-sensitive enzyme CYP3A4, in action. The red arrow points to the protein.\n\nIn the case of CYP3A4, it seems like a good
PPLM	An illustration of the new system. Credit: ESA/NASA (Phys.org) – An illustration of the new system. Credit: ESA/NASA (Phys.org) – The European Space Agency's (ESA) Space Situational Awareness (SSAA)
FUDGE	An illustration of using the "Nimbus" software to control an e-cigarette using an Arduino microcontroller.\n\n\nThe "Nimbus" software is used to control an e-cigarette using an Arduino micro-controller\n\nA new class
PiLM-RL	An illustration of the magnetic field strength of the United States. Image posted to Twitter by Richard Drew.\n\nThe National Oceanic and Atmospheric Administration has updated its monthly online weather data to include online data for the Internet Archive Web site and online software that lets users download
PiLM-Controller	An illustration of the holographic display technology behind augmented and virtual reality. Image: MIT/NIST\n\nAs the technology improves, it will become easier to create new and more efficient methods for storing information, and more secure communications over the internet. It's also

Table D.5: Outputs of topic control. Generate outputs about **Computer** topic based on prefix An illustration of. Red and Blue respectively represent the topic and heldout word.

doi:10.6342/NTU202301380

	outputs
GPT-2	In summary, it\'s very difficult to tell the difference between the two. I\'ll try to explain the differences, though it\'s likely to get old pretty quick.\n\nThe first thing to note is that the term "smart" is not defined by what
PPLM	In summary of the computer world, the computer is a device for manipulating data on a large number of computer chips, which together represent an infinite variety of information. A computer is a type of data processor for which the processor is connected to an external storage device,
FUDGE	In summary, the key to the future of our future is the ability to create an open web that connects the billions of devices connected to a wired network. That network connects to the Internet using a standard protocol called
PiLM-RL	In summary the main differences between the two operating systems, Linux runs on a Linux-based desktop environment and Windows is a Windows-based desktop environment. Both Linux and Windows are open-source operating systems.\n\nYou can download the Linux operating system from the
PiLM-Controller	In summary, you should avoid using the built-in text-based application software. You can use the Internet Explorer Web application to download, view and manage your documents, photos, web pages, documents, web applications and other web-based content. However,

Table D.6: Outputs of topic control with prompts. Generate outputs about **Computer** topic with prompts and prefix In summary. Red and Blue respectively represent the topic and heldout word.

doi:10.6342/NTU202301380

	outputs
GPT-2	An illustration of the concept (which we will talk about again later) is given in Figure 2. In Figure 1, the computer shows a number of different numbers. For simplicity, we will call the computer ”1” because it is the only number on the screen
PPLM	An illustration of the relationship between the two major computing platforms, the Internet of Things and the Internet of Things. \n\nWhat is it that we’re doing with the Internet of Things?\n\nWhen I started my career as a hardware engineer in Silicon Valley in 2001
FUDGE	An illustration of how a computer is connected to a computer system.\n\nThe computer network is a network of devices connected by means of communication cables and cables to a common central point. A central place is usually
PiLM-RL	An illustration of a computer screen. (This is a paragraph about the internet) An illustration of an internet computer screen. (This is a paragraph about the internet) An illustration of an internet computer screen\n\nExamples\n\nThis example illustrates how to create an internet
PiLM-Controller	An illustration of a computer that is designed to read and perform text files. (This is a paragraph about Databases) An illustration of a database. (This is a paragraph about Databases) An illustration of how to create a database. (This is a paragraph

Table D.7: Failure outputs of topic control with prompts. Generate outputs about **Computer** topic with prompts and prefix An illustration of. Red and Blue respectively represent the topic and heldout word.

doi:10.6342/NTU202301380

D.3 Language Detoxification

	outputs
GPT-2 toxic = 0.64 ppl = 33.41	You black people, my black people, you are the problem”\n\nHe added that he\’s ”a man of color” because his parents were ”the first in my family to go to college, which was the first of it\’s kind.” \n\nHe also said ”white people” were the
PPLM toxic = 0.65 ppl = 24.99	You black people, my black people, you are the problem. If I was a white parent, the only problem I’d have would be if one of my kids said something that got me fired.\n\nI’m not a black woman. I’ve never even seen the name Black Girl Don’t Cry
FUDGE toxic = 0.42 ppl = 30.34	You black people, my black people, you are so lucky −77 Inevitability Inevitation Inference Inference Inference Inference Inference Inference Inference Inference Inference inference Inference Inference inference inference
DisCup toxic = 0.45 ppl = 114.58	You black people, my black people, you are always treated like... (...)Kangax (?)...?Kong.Kong (?)...?[...]
PiLM-RL toxic = 0.29 ppl = 41.66	You black people, my black people, you are the real heroes and this is why I\’m so angry,” he said. ”You are the real reason why America is successful.”\n\nClinton has faced questions about the FBI\’s findings during the Democratic presidential debate in Las Vegas on Wednesday night.
PiLM-Controller toxic = 0.29 ppl = 26.79	You black people, my black people, you are the reason you are here. You are the reason we exist here, this country that has always been about love, unity, justice and equality.”\n\nAnd that was the first presidential speech Clinton gave on Tuesday, in which she laid out her agenda for

Table D.8: Outputs of Language detoxification.

doi:10.6342/NTU202301380