Towards Neural Programming Interfaces: Appendix 1

Zachary C. Brown
Electrical and Computer Engineering
Duke University
Durham, NC 27708
zac.brown@duke.edu

Nathaniel Robinson
Department of Computer Science
Brigham Young University
Provo, UT 84602
nrobinson@byu.edu

David Wingate
Department of Computer Science
Brigham Young University
Provo, UT 84602
wingated@cs.byu.edu

Nancy Fulda
Department of Computer Science
Brigham Young University
Provo, UT 84602
nfulda@cs.byu.edu

1 Overview

This Appendix provides supplementary information for our main paper. In Section 2, we provide pseudo-code for the discussion of the NPI algorithm discussed in the beginning of Section 3 of the main paper. Section 3 provides a model size comparison (in terms of number of parameters) between a few of our NPI models and other state-of-the-art methods for textual control. In Section 4, we discuss some details regarding how the NPI approach might be applied to other neural network architectures other than OpenAI's GPT-2 model [1], as well as some aspects of Figure 1 in the main paper which were not highlighted for the sake of brevity. Section 5 of this Appendix elaborates and discusses various results reported in Section 4 of the main paper (with many details as to hyperparameters being saved for Section 6 of this Appendix), in addition to reporting some additional results that were not included in the main paper due primarily to page limitations. Finally, Section 6 reports our records of the hyperparameters of many of our experiments.

2 NPI Control for Pretrained Neural Networks (NPIC)

Algorithm 1 presents pseudo-code for applying a Neural Programming Interface X to control a pretrained model P, as discussed at the beginning of Section 3 of the main paper.

3 Model Size Comparison

When compared with other linguistic control models such as CTRL [2] and Meena [3] and methods capable of learning new linguistic tasks without weight updates [4], our NPI method provides a distinct training advantage: it is able to provide both capabilities while requiring three orders of magnitude fewer parameters than GPT-3. This is particularly relevant in contexts where training data is limited, as the number of trainable parameters heavily influences the amount of training data needed to learn a task effectively. Table 1 provides a model size comparison of the various methods.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Algorithm 1 - NPI Control for Pretrained Neural Networks (NPIC)

Inputs: T_{in} ▷ Input for the pretrained model P
Parameters: I_{in} ▷ The set of integer indices representing the hidden layers of P to be controlled
Output: T^D_{out}, H' ▷ Output and hidden activations of the controlled model P'

$\{T_{out}, H\} \leftarrow P(T_{in})$ ▷ 1st forward pass through pretrained model P, arrow denotes assignment operator
$H_{in} \leftarrow \{h_i | i \in I_{in}\} \subseteq H$
$D_{out} \leftarrow X(H_{in})$ ▷ Forward pass through the NPI model X
$T^D_{out} \leftarrow T_{in}$
$H' \leftarrow \{\}$
for $j, layer$ `in enumerate(`P`)` **do**: ▷ 2nd forward pass through pretrained model P
    $T^D_{out} \leftarrow layer(T^D_{out})$
    if $j \in I_{in}$ **then** $T^D_{out} \leftarrow T^D_{out} \oplus d_{l,j}$ **end if** ▷ $d_{l,j} \in D_{out}$ is added to T^D_{out} element-wise
    H'.append(T^D_{out})
end for
return $\{T^D_{out}, H'\}$

Table 1: Comparison between the capabilities and parameter counts of various NPI models, CTRL [2], Meena [3], and GPT-3 [4]. *Control* refers to the ability of the model to control its linguistic output, and takes on the values *True* or *Limited* in the case of GPT-3, which can only be controlled based on input text and is not explicitly trained for control. *No model updates* refers to the capability of the model to learn new linguistic tasks without weight updates to the linguistic model itself, and takes on the boolean values *True* or *False*. *Npi params* refers to the number of parameters that comprise the npi model in each method (with NPI model parameters separated from their linguistic model counterparts by a + symbol). *Total params* refers to the number of total parameters in the model.

model name	control	no model updates	npi params	total params
cat-NPI small + GPT-2 small	True	True	~15.5 Million	~140.0 Million
cat-NPI large + GPT-2 small	True	True	~15.5 Million	~140.0 Million
cat-avoidance-NPI + GPT-2 small	True	True	~15.5 Million	~140.0 Million
offense-avoidance-NPI + GPT-2 medium	True	True	~103.7 Million	~458.6 Million
CTRL	True	False	0	1.63 Billion
Meena	True	False	0	2.6 Billion
GPT-3	Limited	True	0	175 Billion

4 Applying NPIs to Diverse Models

Our experiments sought to control the hidden layers between GPT-2 [1] 'blocks' [5], which do not incorporate a form of hidden layer aggregation (i.e. residual connections [6] or attention mechanisms [7] across hidden layers). It is important to note that for models that do not incorporate some form of hidden layer aggregation A, the number of control layers used to influence the controlled output in a single forward pass of a pretrained model P could theoretically be reduced down to $m_{min} = 1$. This is due to the fact that without hidden layer aggregation, the hidden layers (and control layers, by extension) are processed sequentially, which suggests the final output could be controlled entirely at the end of where the control sequence is processed. More experimentation is needed to better understand the rules that determine which set of control layers is optimal.

In the case where layer aggregation methods are used, however, the value of m_{min} varies according to the specific use of layer aggregation, with $1 \leq m_{min} \leq n$. We refer to models that incorporate layer aggregation across all n hidden layers (such as DenseNET [8]) as 'fully aggregated networks',

Table 2: Model comparisons across 1000 utterances (500 for word probability baselines) in the context of political bias mitigation. Fluency was evaluated using a crowd-sourced Likert scale. Mechanical Turk workers with a "master" qualification allotted 1 to 5 stars for text quality. The target words were the names of political candidates from the previous two presidential elections in the USA. It is unclear why the probability baseline outputs scored higher than the unmodified GPT-2 in fluency, as these two approaches are identical except for the probability of outputting the name of a political figure. We attribute this to the variability in human text evaluations, as the word probability baselines were evaluated several weeks after all other models in the table, and at a different time of day. Note that despite its high fluency rating, the word probability method failed to induce the target word.

	target in output	embed shifts	avg shift	fluency Likert scale	fluency std dev
word induction - Candidate A					
NPI	**46.7%**	**75.0%**	0.070	3.57	1.16
word prob baseline	0.40%	0.00%	0.000	**4.14**	0.80
unmodified GPT-2	0.00%	N/A	N/A	3.74	1.15
word induction - Candidate B					
NPI	**1.00%**	**49.8%**	0.001	3.48	1.13
word prob baseline	0.00%	0.00%	0.000	**4.17**	0.88
unmodified GPT-2	0.00%	N/A	N/A	3.71	1.10
word induction - Candidate C					
NPI	**34.0%**	**74.5%**	0.056	3.79	1.07
word prob baseline	0.00%	0.00%	0.000	**4.12**	0.79
unmodified GPT-2	0.00%	N/A	N/A	3.72	1.15

and provide an illustration of a Neural Programming Interface acting on such a network in Figure 1 of the main paper. For networks that are not fully aggregated, Figure 1 serves to illustrate the locally aggregated portion of the pretrained network that is to be controlled (with the aggregation portion A simply omitted in cases where layer aggregation is entirely absent, such as in our GPT-2 experiments in Section 4 of the main paper).

5 Additional Experiments and Discussion

5.1 Bias Mitigation in Politics

The inherent biases of pre-trained language models have been clearly and repeatedly established [9, 10, 11, 12, 13]. NPIs provide a new and potentially powerful tool for creating more balanced language model output. Table 2 demonstrates the ability of NPIs to influence the content of generated text to refer to specific political candidates. The differing success of the NPI with different candidate names is a result of the candidates' relevant prominence in news articles at the time the GPT-2 model was trained. By strategically leveraging NPIs to counterbalance such biases in the training data, it is possible to increase the prevalance with which each candidate is named.

For completeness, we compared the NPI guidance method of word induction to a word probability baseline ("prob baseline") in which tokens comprising the candidate's name were automatically selected for output by the GPT-2 model whenever they had probability > 0. NPI's dramatically superior performance can be explained by the observable difference in embedding shifts between the two models. In situations where the candidate's name cannot be output without violating fluency or consistency constraints, the NPI is nevertheless able to influence the output text in the general *direction* of the target word, thus creating opportunities to output the target word later on.

The results in Table 2 show that as long as a candidate is well-referenced in the training corpus (Candidates A and C), NPI guidance is able to increase the prevalence of the candidate's name far above that enabled by word probability adjustments. In the case of under-represented entities (e.g. Candidate B), we hypothesize that a combination of (a) fine-tuning on a small dataset containing the candidate's name and (b) utilizing NPI guidance on the fine-tuned model would increase the frequency of the target word.

5.2 Further Discussion of Topic Induction Experiments

This section references Table 1 in the main paper.

Note that the low-resource NPI performs remarkably well despite the tiny size of its data set. Evaluation of the outputs from this model show evidence of overfitting, with specific words and phrases showing up repeatedly. Nevertheless, analysis of the embedding shifts across the NPI models show that the embedded representation of the GPT-2 outputs moves toward the target word in far more sentences than those in which the target word actually appears. Manual inspection of GPT-2 output texts show that this is caused by the introduction of words such as “furry", “purred" or “prey", which are strongly associated with the target word. Words denoting other small furry animals, including “dog” and “squirrel” are also prevalent. Hence, the NPI has not merely increased the frequency of the word “cat”, but has shifted the entire tone of the output text toward the *idea* of catness. We find this remarkable. Other examples of this capability are in Table 4.

The standard cat-NPI, trained on 70,227 example sentences, successfully produces the target word 54.2% at epoch 5, but only manages to shift the overall tone of each sentence 79.8% of the time. The model undergoes some learning distress at epoch 20, then sacrifices target word instances slightly in order to attain an impressive 94.8% success ratio at shifting the sentence topic. Critically, the average shift in all cases is ≤ 0.125 and the standard deviation across all embedding shifts is less than 0.1. This suggests that the NPI model has successfully kept the overall topic of the perturbed GPT-2 outputs extremely close to content of the unmodified GPT-2 model.

5.3 Example Sentences

In Table 3 we report text samples generated by our ‘cat’ induction as well as offense-avoidance NPI models. These samples were manually selected to display various characteristics of the control exerted by our NPI models, and do not necessarily represent the best or worst samples generated by these models. In reporting these samples, we made a true effort to include samples that show some of the failure-modes of our models (particularly in the 5th example), and refer the reader to the fluency metrics reported in the main paper as a reminder that our NPI models achieved similar fluency ratings to the original GPT-2 model (using deterministic filtering) throughout our experiments.[1]

5.4 Summary Discussion of Experimental Results

As can be seen from our experiments, a key advantage of the NPI framework (as opposed to fine-tuning or filtering the probability of target words during output sampling) is its ability not merely to model language but to actively induce a topic shift within the perturbed GPT-2 outputs. In contrast, if a GPT-2 model is fine-tuned on a specialized data set, it learns to produce text that resembles that data set, but only if it has learned to over-fit on the fine-tuning set or if the input text matches the fine-tuning data. Directly filtering GPT-2 final layer sampling probabilities did not manage to *pivot* toward the specialized information in cases where more generalized inputs are provided. This failure was particularly noticeable in word induction applications. As described earlier, our ‘word prob’ method was to choose a target token by default anytime the GPT-2 assigned the target token a non-zero probability. (This was the most aggressive feasible term induction method via probability tweaks. To go any further and force GPT-2 to choose a token with zero probability would be equivalent to randomly seeding the output text with the target token; it wouldn’t take advantage of GPT-2’s language modeling capabilities.)

Similarly, NPIs are better prepared to avoid offensive speech than existing approaches. While word avoidance can be achieved by reducing the output probabilities of specific terms to zero during the text generation process, such methods would not generalize to more complex forms of avoidance that seek to prevent word use only in some contexts, or that seek to avoid statements such as "I doubt your sanity", which offend without using strictly offensive words. The promising results in this subsection suggest that the NPI architecture would perform well in such scenarios.

[1] To see more examples of NPI results and perturbed output text, please see our GitHub repository, `https://github.com/DRAGNLabs/towards-neural-programming-interfaces`.

Table 3: Sample text outputs generated by our NPI models. Newlines have been replaced by spaces, and potentially offensive terms and identifiers are obfuscated. Target words are bolded. Limited fluency in both the GPT-2 and NPI outputs is due to (a) the small-size GPT-2 model used, (b) the short GPT-2 context of 10 to 15 tokens, and (c) the deterministic sampling method top_k=1, as discussed in Section 3.1.1 of the main paper.

cat-NPI small	CONTEXT	"How"
	GPT-2 OUTPUT	"to use the API The API is available in the following formats: HTML CSS JavaScript CSS3 JavaScript // This is the default"
	NPI OUTPUT	"to use the API The API is available in the following format: # **cat** /proc/sys/kernel/cpuinfo #include <linux/cpuinfo.h> # ######"
cat-NPI Large	CONTEXT	"magazine writer david dalton the following about her first"
	GPT-2 OUTPUT	"book, "The Secret of the Mind," by the late John Steinbeck. The book is a collection of essays by the author, and the book is available"
	NPI OUTPUT	"Cal headseeth smashed into the head of the **cat**. The **cat** is a very small **cat**, but it is a very good **cat**. The **cat** is a very good **cat**, but"
cat-NPI Large	CONTEXT	"near pyongyang november one nine five zero during the korean war"
	GPT-2 OUTPUT	". The Korean War was a major military event in the Korean War. The U.S. military has been accused of using chemical weapons against civilians in Syria."
	NPI OUTPUT	". The Korean War was a war of attrition, and the Japanese were not only a good team, but also a great team. "I think wére going to be a **cat**-and-mouse game."
offense-	CONTEXT	"**B****es** always think somebody want them or they little c*****e lol chilling dont exist nomore ? Iont want yall **h**s**"
avoidance-NPI	GPT-2 OUTPUT	" to be able to get away with this **s**t**. I'm not saying that the people who are doing this are evil."
	NPI OUTPUT	" to be a part of the world. I'm not sure if this is the case for all of the other characters, but I'm not sure if it's a good idea to have"
offense-	CONTEXT	"just got another warning. why are you such a **d**k**? d*****e"
avoidance-NPI	GPT-2 OUTPUT	": x**I'm not a **d**k**. x**I'm just a guy who likes to play video games. x**I'm not sure if he's a gamer, but "
	NPI OUTPUT	". I'm not a dog. I'm not a dog. I'm not a dog. I'm not a dog. I'm not a dog. I'm not a dog. I'm not a dog. I'm not a dog."
offense-	CONTEXT	"Go die in a hole somewhere dont censor you racist"
avoidance-NPI	GPT-2 OUTPUT	"**a**holes** you arent welcome here you arent welcome here you arent welcome here you arent welcome here you arent "
	NPI OUTPUT	" dont dont censor you racist The only thing that matters is that you're not racist The only thing that matters is that you're not racist."

6 Hyperparameter Details for Models and Experiments

In Figures 1-4 we provide the hyperparameters used for models featured in our experiments, as well as training details such as data set content, number of training points, and learning rate, among others. The Adam optimizer was the only optimizer used throughout training of all models.

Table 4: More text outputs generated by our NPI models displaying NPI's capability to steer GPT-2 in the direction of abstract concepts in addition to use of target words. Harry-Potter-NPI is an NPI trained with a long list of words related to J. K. Rowling's fantasy series *Harry Potter* as targets. Target words are bold. Words pertaining to the target topic that were outside the list of target words are italicized. In addition to the examples listed here, many of the political figure-inducing NPI models were able to induce the names of family members, opponents, or associates of the target political figure (names that were outside the set of target terms during training). See our GitHub repository `https://github.com/DRAGNLabs/towards-neural-programming-interfaces` for more examples of NPI outputs.

Harry-Potter-NPI	CONTEXT	"marriage advertisement and his wife as being from nearby booth-town"
	GPT-2 OUTPUT	"The woman, who was not identified, said she was driving to work when she was struck by a car. 'I was just trying to get out of the car and get out of the car. 'I'm not going to be able to drive it,' he said."
	NPI OUTPUT	"The woman was taken to the hospital, where she was pronounced dead. The woman was a member of the Order of the Black Rose, and was a member of the **Order of the Phoenix**. **The Order of the Phoenix** was a powerful and powerful organization that had been created by the **Phoenix** Council."
Harry-Potter-NPI	CONTEXT	"the university of dublin and four for the national university of ireland"
	GPT-2 OUTPUT	"The university's president, Dr. Michael O'Brien, said the university's decision to withdraw the lawsuit was 'a mistake.' 'I think it's a mistake,' he said. 'I think it's a mistake to say that we're going to be able to do this."
	NPI OUTPUT	"The first of the three to be *expelled* was the young man who had been the first to be *expelled*. 'I'm sorry, but I'm not going to be here for a while. I'm going to go to the *library* and find a way to get the *key*. The *key* is a *key* that is used to unlock the door."
Harry-Potter-NPI	CONTEXT	"this effort was considered to be the largest drainage project of its time"
	GPT-2 OUTPUT	"The project was completed in the early 1990s, and the project was completed in the early 2000s.The project was funded by the National Science Foundation, the National Science Foundation, the National Science Foundation, the National Science Foundation, the National Science Foundation, the National Science Foundation, the"
	NPI OUTPUT	"The first of the three major drainage projects was the Great Basin Water Project, which was completed in 1859. The project was funded by the *Queen's Council*, and the *Queen's Council* was a secret organisation that was created by the Queen to protect the Queen from the forces of the *Dark Gods*. The Queen's power is not limited to the *Dark Gods*. The"

	cat-inducing NPI from Table 3	cat-avoidance NPI from Table 3	offense-avoidance NPI from Table 3
Info for NPI:			
dims of every layer	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500	460800x112, 112x112, 112x56, 56x56, 56x56, 56x56, 56x112, 112x112, 112x460800
layer details	ReLU between each pair of layers	ReLU between each pair of layers	ReLU between each pair of layers
learning rate details	lr=.000001	lr=.00001	lr=.000001
batch size	5	5	5
size of training data	70455 data points (305 pkls)	693 data points (3 pkls)	7800 data points (25 pkls)
size of testing data	23180 data points	228 data points	2600 data points
epochs	20	25	4
loss coefficients	gamma=3.0, alpha=10.0, beta=1.0	gamma=3.0, alpha=10.0, beta=1.0	gamma=2.0, alpha=10.0, beta=0.0
Info for Content classifier			
dims of every layer	153500x12, 12x6, 6x3, 3x1	153500x12, 12x6, 6x3, 3x1	460800x28, 28x14, 14x7, 7x1
layer details	ReLU between each pair of layers, Sigmoid after layer 4	ReLU between each pair of layers, Sigmoid after layer 4	ReLU between each pair of layers, Sigmoid after layer 4
learning rate details	lr=.0000001	lr=.0000001	lr=.0000001
batch size	5	5	5
size of training data	70455 data points (305 pkls)	70455 data points (305 pkls)	41250 data points (11 pkls)
size of testing data	23180 data points	23180 data points	13750 data points
epochs	30	20	30
classifier trained further during NPI training?	no	yes	no
Info for Discriminator:			
dims of every layer	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1	460800x56, 56x56, 56x28, 28x28, 28x14, 14x14, 14x1
layer details	ReLU between each pair of layers, Sigmoid after layer 7	ReLU between each pair of layers, Sigmoid after layer 7	ReLU between each pair of layers, Sigmoid after layer 7
learning rate details	lr=.000001	lr=.00001	lr=.000001
batch size	5	5	5
size of training data	70455 data points (305 pkls)	693 data points (3 pkls)	7800 data points (25 pkls)
size of testing data	23180 data points	228 data points	2600 data points
epochs	20	25	4
Info about Data Set:			
context window	10	10	15
number of GPT-2 iterations	10	10	15
GPT-2 model used	small	small	medium
GPT-2 layers used	2, 9	5, 11	2, 9
Random seeding	Yes	Yes	No

Figure 1: Hyperparameters used for models featured in this paper.

	political-avoidance NPI from Table 4	racial-slur-avoidance NPI from Table 4	gender-slur-avoidance NPI from Table 4
Info for NPI:			
dims of every layer	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500
layer details	ReLU between each pair of layers	ReLU between each pair of layers	ReLU between each pair of layers
learning rate details	lr=0.000001	lr=.000001	lr=.000001
batch size	5	5	5
size of training data	693 data points (3 pkls)	693 data points (3 pkls)	693 data points (3 pkls)
size of testing data	228 data points	228 data points	228 data points
epochs	30	30	50
loss coefficients	gamma=3.0, alpha=10.0, beta=1.0	gamma=3.0, alpha=10.0, beta=1.0	gamma=3.0, alpha=10.0, beta=1.0
Info for Content classifier			
dims of every layer	153500x12, 12x6, 6x3, 3x1	153500x12, 12x6, 6x3, 3x1	153500x12, 12x6, 6x3, 3x1
layer details	ReLU between each pair of layers, Sigmoid after layer 4	ReLU between each pair of layers, Sigmoid after layer 5	ReLU between each pair of layers, Sigmoid after layer 6
learning rate details	lr=.001	lr=.001	lr=.001
batch size	5	5	5
size of training data	8547 data points (37 pkls)	6006 data points (26 pkls)	12243 data points (53 pkls)
size of testing data	2812 data points	1976 data points	4028 data points
epochs	40	20	50
classifier trained further during NPI training?	no	no	no
Info for Discriminator:			
dims of every layer	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1
layer details	ReLU between each pair of layers, Sigmoid after layer 7	ReLU between each pair of layers, Sigmoid after layer 7	ReLU between each pair of layers, Sigmoid after layer 7
learning rate details	lr=.000001	lr=.000001	lr=.000001
batch size	5	5	5
size of training data	693 data points (3 pkls)	693 data points (3 pkls)	693 data points (3 pkls)
size of testing data	228 data points	228 data points	228 data points
epochs	30	30	50
Info about Data Set:			
context window	10	10	10
number of GPT-2 iterations	10	10	10
GPT-2 model used	small	small	small
GPT-2 layers used	5, 11	5, 11	5, 11
Random seeding	Yes	Yes	Yes

Figure 2: Hyperparameters used for models featured in this paper.

	long-NPI from Table 5	short-NPI from Table 5
Info for NPI:		
dims of every layer	153500x50, 50x50, 50x25, 25x25, 25x25, 25x50, 50x50, 50x153500	153500x50, 50x50, 50x25, 25x25, 25x25, 25x50, 50x50, 50x153500
layer details	ReLU between each pair of layers	ReLU between each pair of layers
learning rate details	lr=.000001	lr=.000001
batch size	5	5
size of training data	1899 data points (3 pkls)	1899 data points (3 pkls)
size of testing data	630 data points	630 data points
epochs	10	10
loss coefficients	gamma=3.0, alpha=10.0, beta=1.0	gamma=3.0, alpha=10.0, beta=1.0
Info for Content classifier		
dims of every layer	153500x12, 12x6, 6x3, 3x1	153500x12, 12x6, 6x3, 3x1
layer details	ReLU between each pair of layers, Sigmoid after layer 5	ReLU between each pair of layers, Sigmoid after layer 5
learning rate details	lr=.0000001	lr=.0000001
batch size	5	5
size of training data	25320 data points (40 pkls)	25320 data points (40 pkls)
size of testing data	8400 data points	8400 data points
epochs	100	100
classifier trained further during NPI training?	no	no
Info for Discriminator:		
dims of every layer	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1
layer details	ReLU between each pair of layers, Sigmoid after layer 7	ReLU between each pair of layers, Sigmoid after layer 7
learning rate details	lr=.000001	lr=.000001
batch size	5	5
size of training data	1899 data points (3 pkls)	1899 data points (3 pkls)
size of testing data	630 data points	630 data points
epochs	10	10
Info about Data Set:		
context window	10	10
number of GPT-2 iterations	10	10
GPT-2 model used	small	small
GPT-2 layers used	2, 9	2, 9
Random seeding	No	No

Figure 3: Hyperparameters used for models featured in this paper.

	"Candidate A"-induction NPI from Appendix 1	"Candidate B"-induction NPI from Appendix 1	"Candidate C"-induction NPI from Appendix 1
Info for NPI:			
dims of every layer	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500	153500x50, 50x50, 50x25, 25x25, 25x25, 25x25, 25x50, 50x50, 50x153500
layer details	ReLU between each pair of layers	ReLU between each pair of layers	ReLU between each pair of layers
learning rate details	lr=0.000001	lr=0.000001	lr=0.000001
batch size	5	5	5
size of training data	693 data points (3 pkls)	693 data points (3 pkls)	693 data points (3 pkls)
size of testing data	228 data points	228 data points	228 data points
epochs	70	30	70
loss coefficients	gamma=3.0, alpha=10.0, beta=1.0	gamma=3.0, alpha=10.0, beta=1.0	gamma=3.0, alpha=10.0, beta=1.0
Info for Content classifier			
dims of every layer	153500x12, 12x6, 6x3, 3x1	153500x12, 12x6, 6x3, 3x1	153500x12, 12x6, 6x3, 3x1
layer details	ReLU between each pair of layers, Sigmoid after layer 4	ReLU between each pair of layers, Sigmoid after layer 4	ReLU between each pair of layers, Sigmoid after layer 4
learning rate details	lr=.001	lr=.001	lr=.001
batch size	5	5	5
size of training data	8547 data points (37 pkls)	55440 data points (240 pkls)	49434 data points (214 pkls)
size of testing data	2812 data points	18240 data points	16264 data points
epochs	40	20	16
classifier trained further during NPI training?	no	no	no
Info for Discriminator:			
dims of every layer	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1	153500x25, 25x25, 25x12, 12x12, 12x6, 6x6, 6x1
layer details	ReLU between each pair of layers, Sigmoid after layer 7	ReLU between each pair of layers, Sigmoid after layer 7	ReLU between each pair of layers, Sigmoid after layer 7
learning rate details	lr=.000001	lr=.000001	lr=.000001
batch size	5	5	5
size of training data	693 data points (3 pkls)	693 data points (3 pkls)	693 data points (3 pkls)
size of testing data	228 data points	228 data points	228 data points
epochs	70	30	70
Info about Data Set:			
context window	10	10	10
number of GPT-2 iterations	10	10	10
GPT-2 model used	small	small	small
GPT-2 layers used	5, 11	5, 11	5, 11
Random seeding	Yes	Yes	Yes

Figure 4: Hyperparameters used for models featured in this paper.

References

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[2] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation, 2019.

[3] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot, 2020.

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

[5] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[8] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[9] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *NIPS*, pages 4349–4357. Curran Associates, Inc., 2016.

[10] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. Quantifying and reducing stereotypes in word embeddings. *CoRR*, abs/1606.06121, 2016.

[11] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[12] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.

[13] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. *CoRR*, abs/1605.09096, 2016.