Towards Neural Programming Interfaces

Zachary Brown[2]*, Nathaniel Robinson[1], David Wingate[1], Nancy Fulda[1]

[1]Computer Science, Brigham Young University
[2]Electrical and Computer Engineering, Duke University

*Majority of work completed while at Brigham Young University

Motivation

Pretrained Neural Network

- Strong domain model
- Many parameters
- Difficult and expensive to (fine)tune

Neural Programming Interface (NPI)

- Control
- No change to pretrained model

Markus Gjengaar, https://unsplash.com/photos/v3l8kTbPhzA

Motivation

Avoiding Undesirable Output

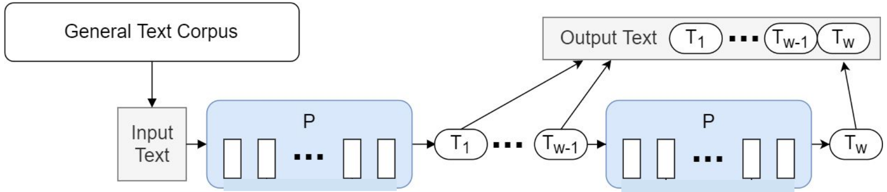
- Offensive speech
  - Racial slurs
  - Gender slurs
  - Other
- Politically charged phrases and topics

Encouraging Desirable Output

- Preferred phrases and topics
  - E.g. ‘cat’ for a pet owner
  - Favored political candidates
  - Other
- Style preferences
  - E.g. simple vs diverse vocabulary

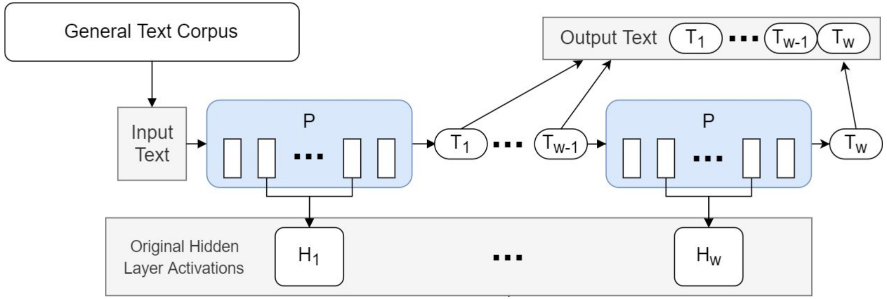
Approach: NPIs - Learn a New Model to Control P

Use a neural network (a Neural Programming Interface or NPI) to control a large pretrained network P by perturbing hidden layer activations



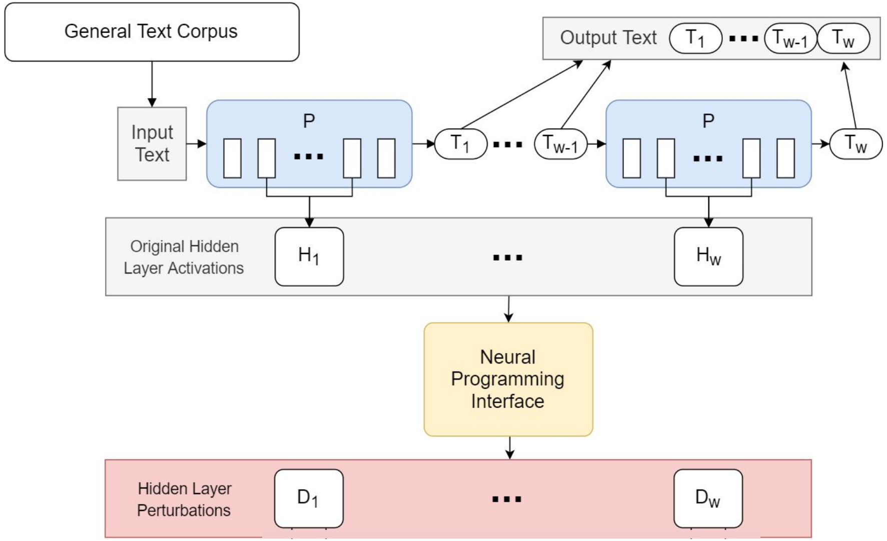
Approach: NPIs - Learn a New Model to Control P

Use a neural network (a Neural Programming Interface or NPI) to control a large pretrained network P by perturbing hidden layer activations



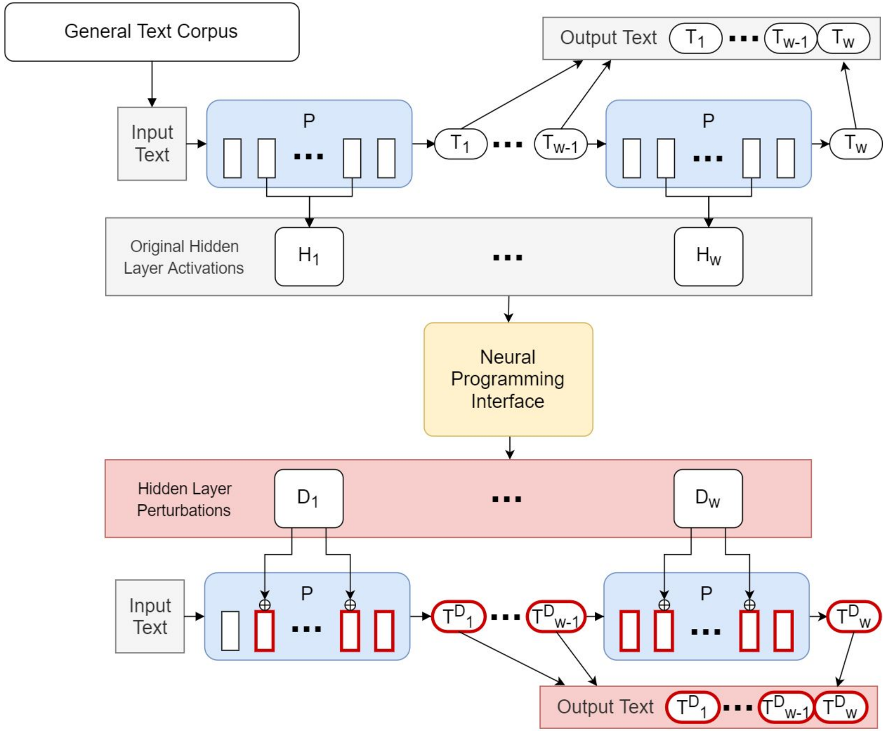
Approach: NPIs - Learn a New Model to Control P

Use a neural network (a Neural Programming Interface or NPI) to control a large pretrained network P by perturbing hidden layer activations



Approach: NPIs - Learn a New Model to Control P

Use a neural network (a Neural Programming Interface or NPI) to control a large pretrained network P by perturbing hidden layer activations

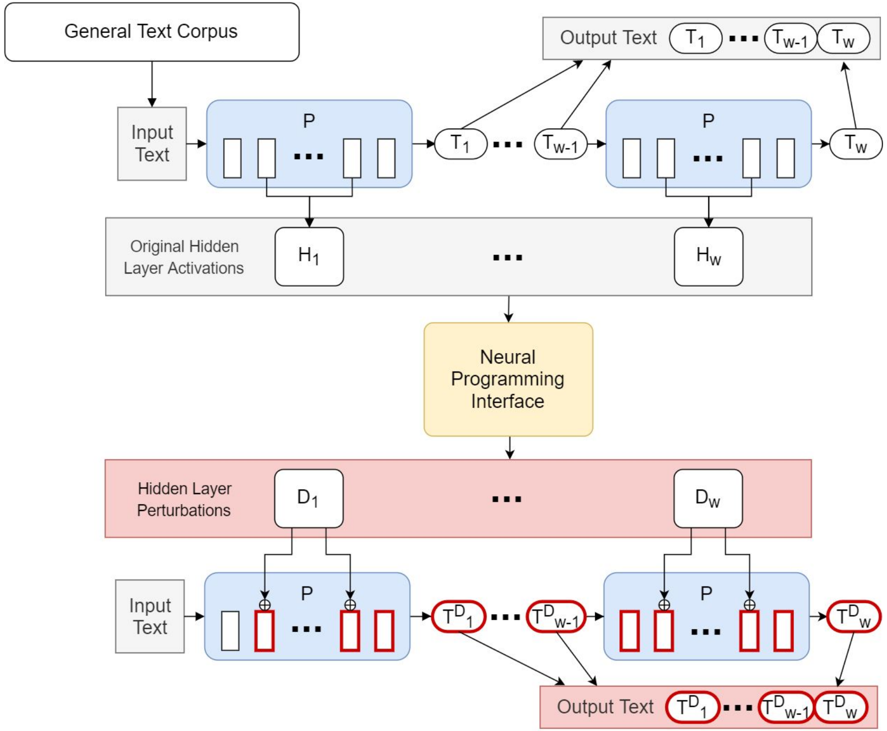

General Text Corpus
Output Text T_1 … T_{w-1} T_w
Input Text
P
T_1 … T_{w-1}
P
T_w
Original Hidden Layer Activations
H_1
…
H_w
Neural Programming Interface
Hidden Layer Perturbations
D_1
…
D_w
Input Text
P
T^D_1 … T^D_{w-1}
P
T^D_w
Output Text T^D_1 … T^D_{w-1} T^D_w

Approach: NPIs - Learn a New Model to Control P

Use a neural network (a Neural Programming Interface or NPI) to control a large pretrained network P by perturbing hidden layer activations

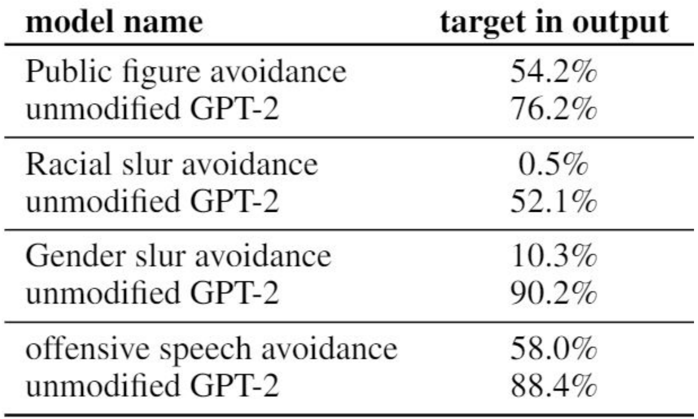
- Domain agnostic
  - Retains P’s domain model

- NPI can learn various ‘control functions’ which are hard to capture in the original domain


General Text Corpus
Output Text T_1 ... T_{w-1} T_w
Input Text
P
T_1 ... T_{w-1}
P
T_w
Original Hidden Layer Activations
H_1
...
H_w
Neural Programming Interface
Hidden Layer Perturbations
D_1
...
D_w
Input Text
P
T^D_1 ... T^D_{w-1}
P
T^D_w
Output Text T^D_1 ... T^D_{w-1} T^D_w

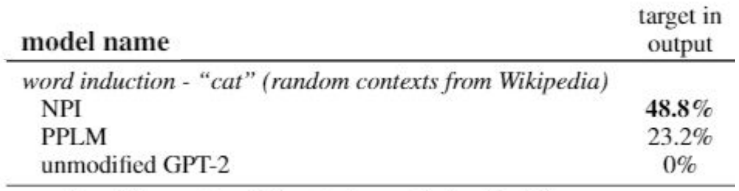
Results

Avoiding Undesirable Output

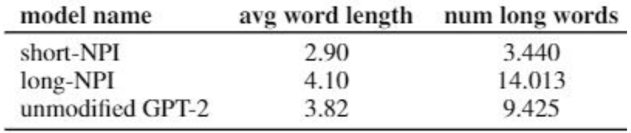


model name	target in output
Public figure avoidance	54.2%
unmodified GPT-2	76.2%
Racial slur avoidance	0.5%
unmodified GPT-2	52.1%
Gender slur avoidance	10.3%
unmodified GPT-2	90.2%
offensive speech avoidance	58.0%
unmodified GPT-2	88.4%

Encouraging Desirable Output



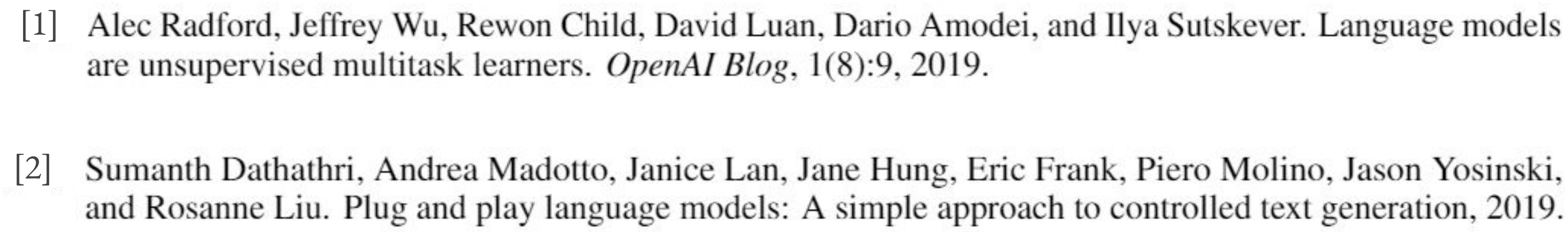
model name	target in output
word induction - "cat" (random contexts from Wikipedia)	
NPI	**48.8%**
PPLM	23.2%
unmodified GPT-2	0%



model name	avg word length	num long words
short-NPI	2.90	3.440
long-NPI	4.10	14.013
unmodified GPT-2	3.82	9.425

References: [1], [2]

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] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation, 2019.

Thank You

Paper: “Towards Neural Programming Interfaces”, ID: 3575

Lab Website: dragn.ai

Code: https://github.com/DRAGNLabs/towards-neural-programming-interfaces

Contact: zac.brown@duke.edu, nrobinson@byu.edu

PCC Lab

DRAGN Labs