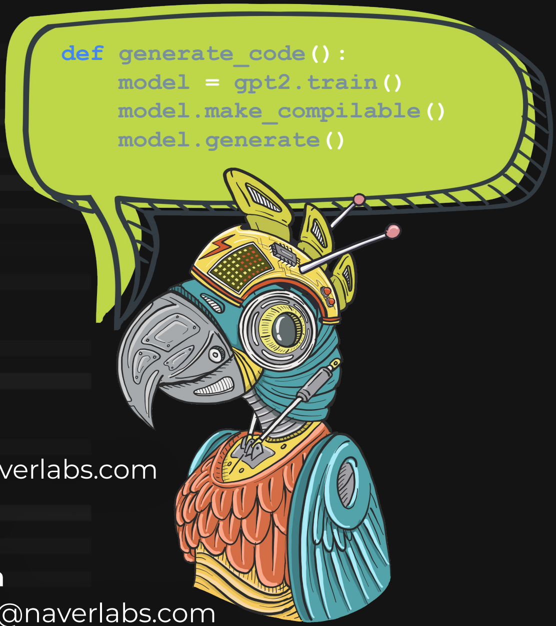
Controlling Conditional Language Models without Catastrophic Forgetting



Tomek Korbak
http://tomekkorbak.com
@tomekkorbak

Germán Kruszewski
german.kruszewski@naverlabs.com
@germank

Hady Elsahar
hady.elsahar@naverlabs.com
@hadyelsahar

Marc Dymetman
marc.dymetman@naverlabs.com
@MarcDymetman

NAVER LABS Europe

Controlling pretrained models

Machine learning is shifting towards relying on **pretrained** generative models.

T5

GPT-Neo

Controlling pretrained models

Machine learning is shifting towards relying on **pretrained** generative models.

These **general**-purpose models can then be used for multiple **downstream tasks**

translation

T5

summarisation

GPT-Neo

code generation

Controlling pretrained models

Machine learning is shifting towards relying on **pretrained** generative models.

These **general**-purpose models can then be used for multiple **downstream tasks**, but need to be adapted to meet task-specific **requirements**.

T5

translation with consistent terminology

factually correct summarisation

GPT-Neo

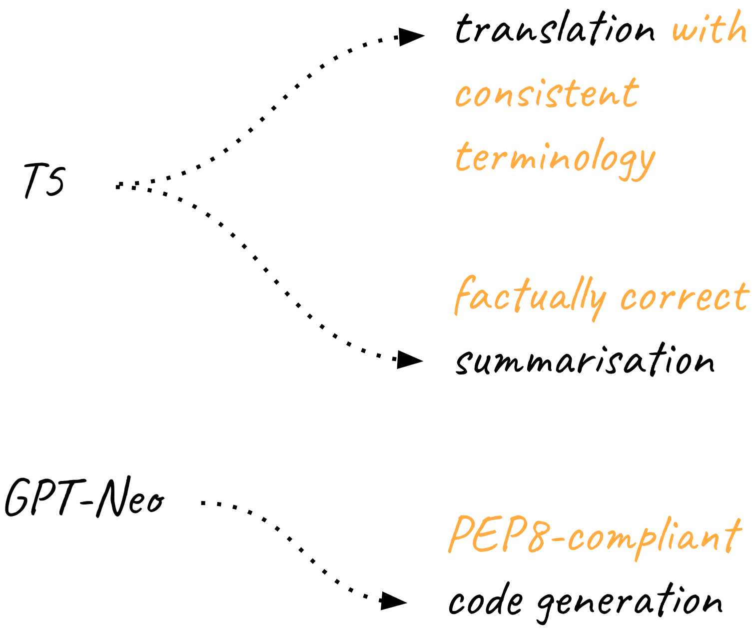
PEP8-compliant code generation

Controlling pretrained models

Machine learning is shifting towards relying on **pretrained** generative models.

These **general**-purpose models can then be used for multiple **downstream tasks**, but need to be adapted to meet their specific **requirements**.

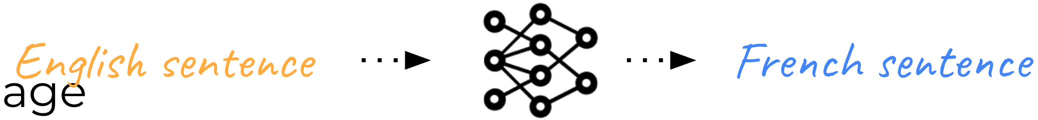
How to **adapt** a pretrained model to a new task without **destroying its capabilities**?



Formal problem statement

We are given a:

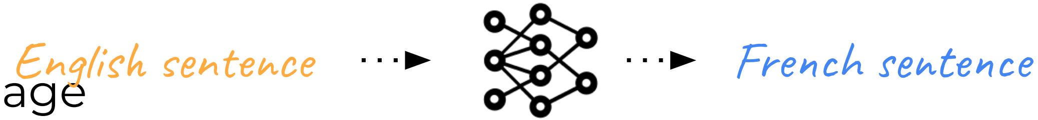
1. pretrained conditional language model $a(x|c)$



Formal problem statement

We are given a:

1. pretrained conditional language model $a(x|c)$
2. a binary constraint $b(x, c)$



acute respiratory disease	maladie respiratoire aiguë
AIDS	SIDA

Formal problem statement

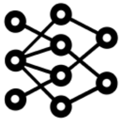
We are given a:

1. pretrained conditional language model a(x|c)
2. a binary constraint b(x, c)

We would like to fine-tune that model to ensure that, for each context c:

1. generates only sequences x satisfying the constraint b(x, c)
2. stays close to pretrained model

English sentence



French sentence

acute respiratory disease	maladie respiratoire aiguë
AIDS	SIDA

Introducing CDPG

In this paper, we present the **Conditional Distributional Policy Gradient (CDPG)** algorithm for fine-tuning conditional language models to satisfy **arbitrary constraints** without destroying their capabilities.

CDPG training step

Step 1: Sample a context c

Two cats are sitting on a mat

CDPG training step

Step 1: Sample a context c

Two cats are sitting on a mat

Step 2: Define a target distribution for c

(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

$$p_c(x) = \frac{1}{Z_c} a(x|c) b(x, c)$$

CDPG training step

Step 1: Sample a context c

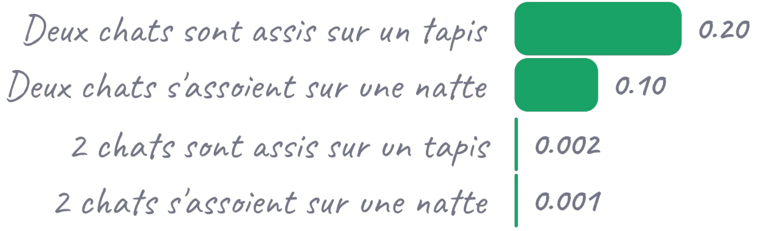
Step 2: Define a target distribution for c

(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Two cats are sitting on a mat



$$p_c(x) = \frac{1}{Z_c} a(x|c) b(x,c)$$

CDPG training step

Step 1: Sample a context c

Step 2: Define a target distribution for c

(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Two cats are sitting on a mat

Deux chats sont assis sur un tapis	0
Deux chats s'assoient sur une natte	0
2 chats sont assis sur un tapis	1
2 chats s'assoient sur une natte	1

$$p_c(x) = \frac{1}{Z_c} a(x|c) b(x,c)$$

CDPG training step

Step 1: Sample a context c

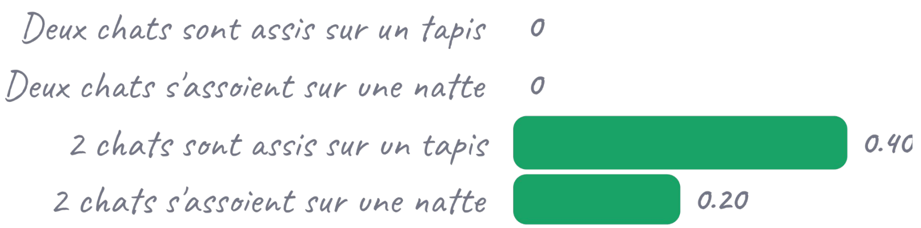
Step 2: Define a target distribution for c

(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Two cats are sitting on a mat



$$p_c(x) = \frac{1}{Z_c} a(x|c) b(x,c)$$

CDPG training step

Step 1: Sample a context *c*

Step 2: Define a target distribution for *c*

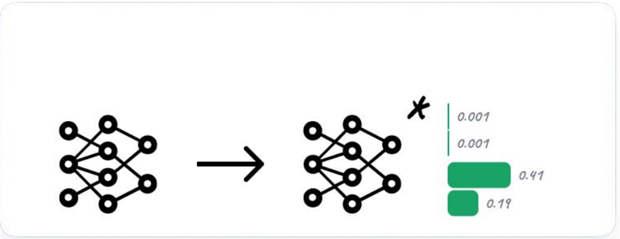
(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Step 3: Update current model to approximate the target distribution when conditioned on *c*

Two cats are sitting on a mat



CDPG training step

Step 1: Sample a context c

Step 2: Define a target distribution for c

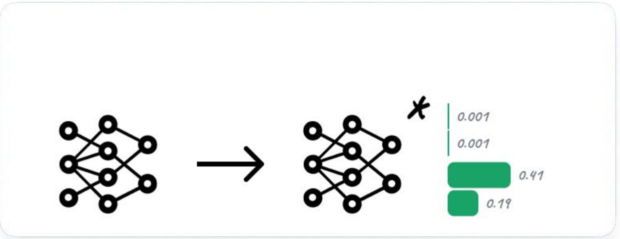
(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Step 3: Update current model to approximate the target distribution when conditioned on c

Two cats are sitting on a mat



$$\nabla_\theta \mathrm{CE}\big(p_c(\cdot), \pi_\theta(\cdot|c)\big)$$

CDPG training step

Step 1: Sample a context c

Step 2: Define a target distribution for c

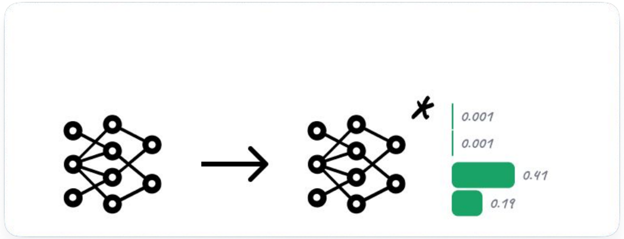
(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Step 3: Update current model to approximate the target distribution when conditioned on c

Two cats are sitting on a mat



$$\nabla_\theta \mathrm{CE}\big(p_c(\cdot), \pi_\theta(\cdot|c)\big)$$

$$= \mathbb{E}_{x\sim\pi_\theta(x|c)} \frac{p_c(x)}{\pi_\theta(x|c)} \nabla_\theta \log \pi_\theta(x|c)$$

CDPG training loop

for context c from set of contexts:

Step 1: Sample a context c

Step 2: Define a target distribution for c

Step 3: Update current model to approximate the target distribution when conditioned on c

Experiments

Control objectives:

1. Terminology-consistent translation
2. Factually correct summarisation
3. Compilable code generation
4. PEP8-compliant code generation

Baselines:

1. Original DPG algorithm for **unconditional** models
2. Reinforce
3. Ziegler: PPO with KL penalties

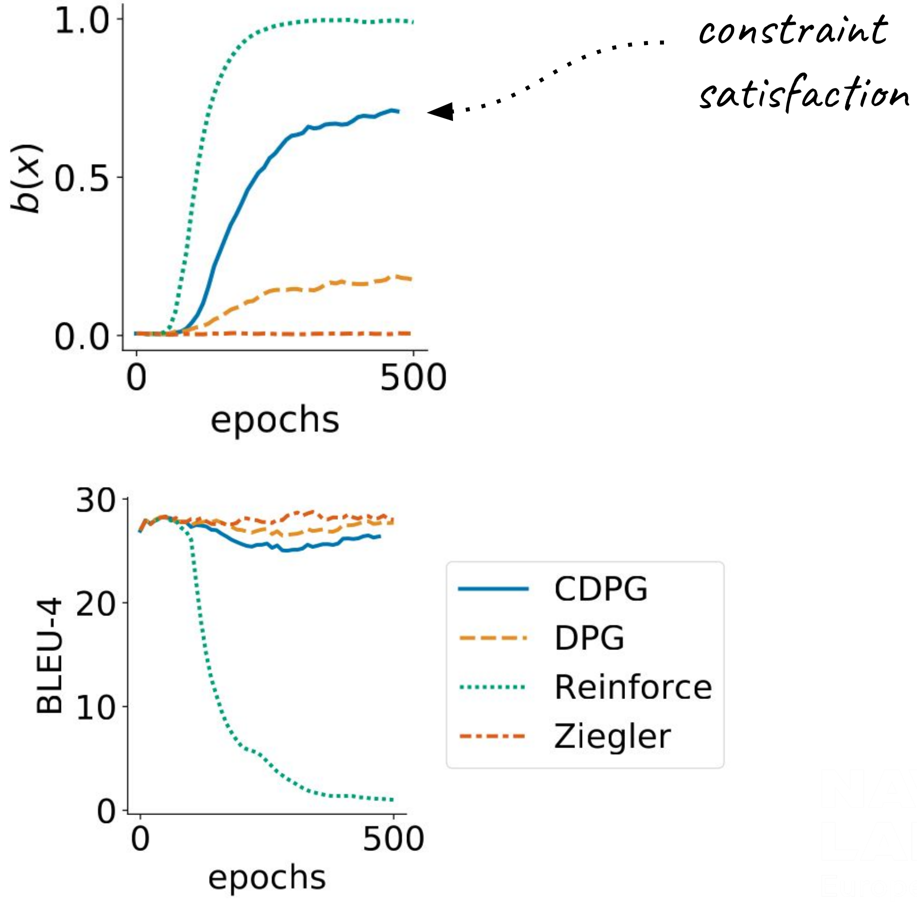
Translation

model: T5

c: English sentence

x: French sentence

constraint: numeral nouns translated as digits



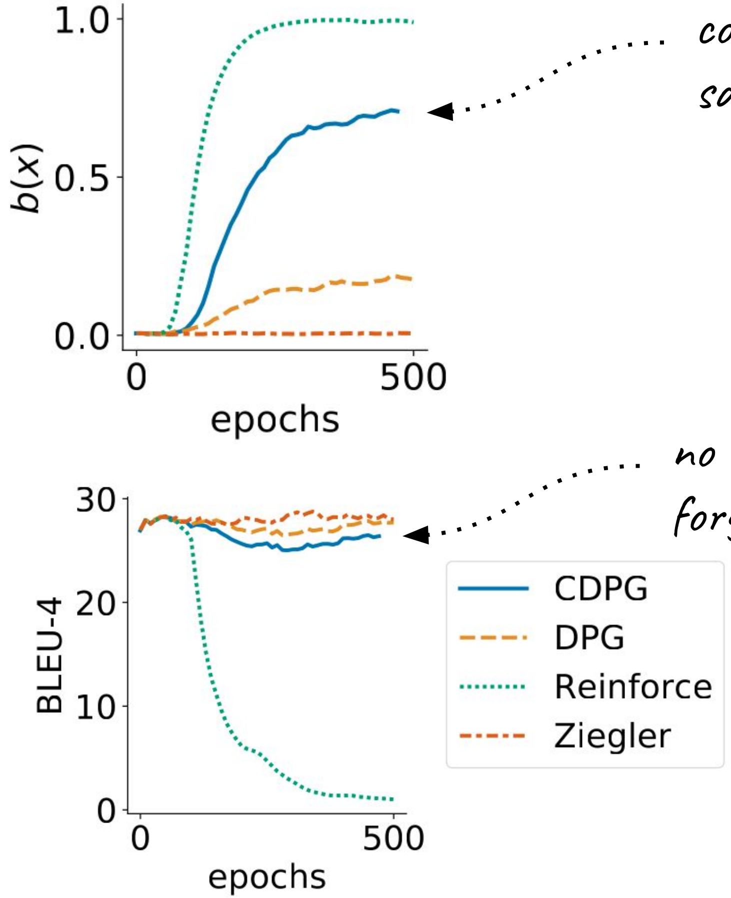
Translation

model: T5

c: English sentence

x: French sentence

constraint: numeral nouns translated as digits



constraint satisfaction

no catastrophic forgetting

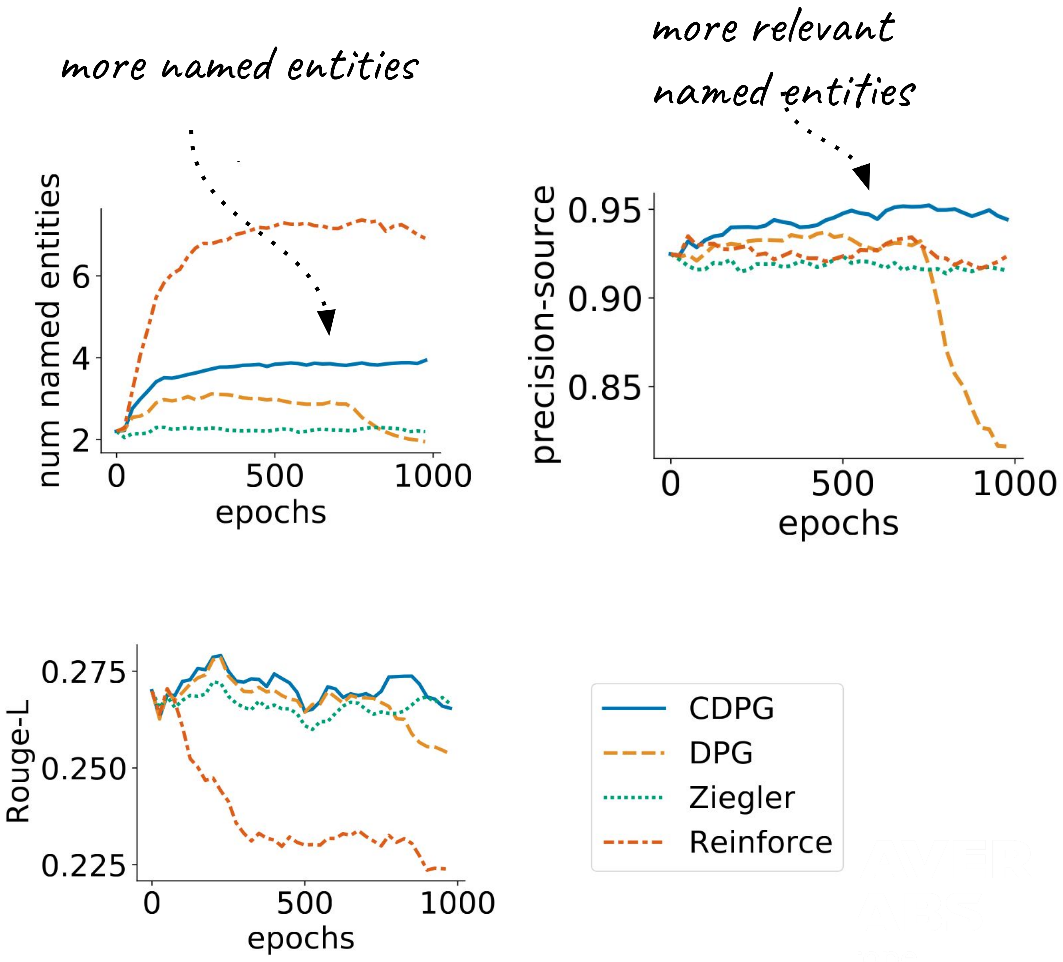
Summarisation

model: T5

c: source document

x: document summary

constraint: all named entities in summary should be mentioned in source document



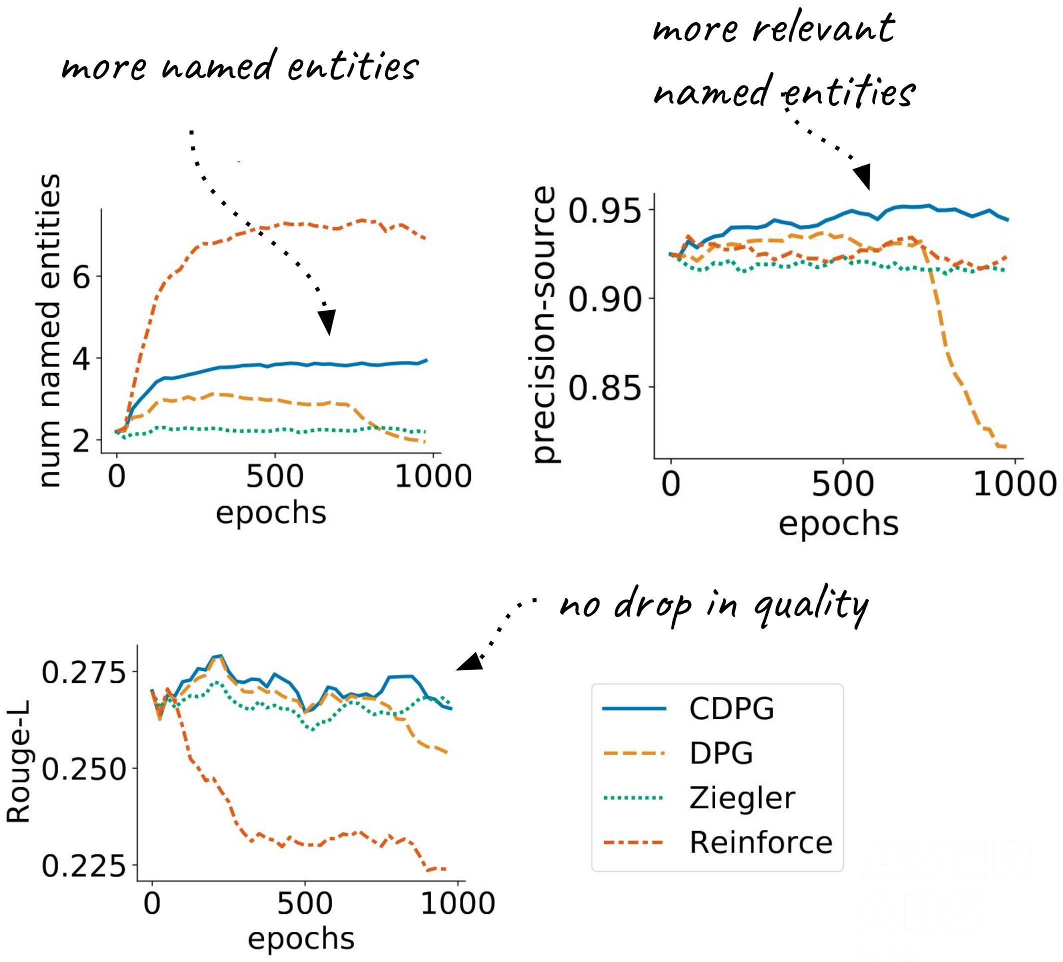
Summarisation

model: T5

c: source document

x: document summary

constraint: all named entities in summary should be mentioned in source document



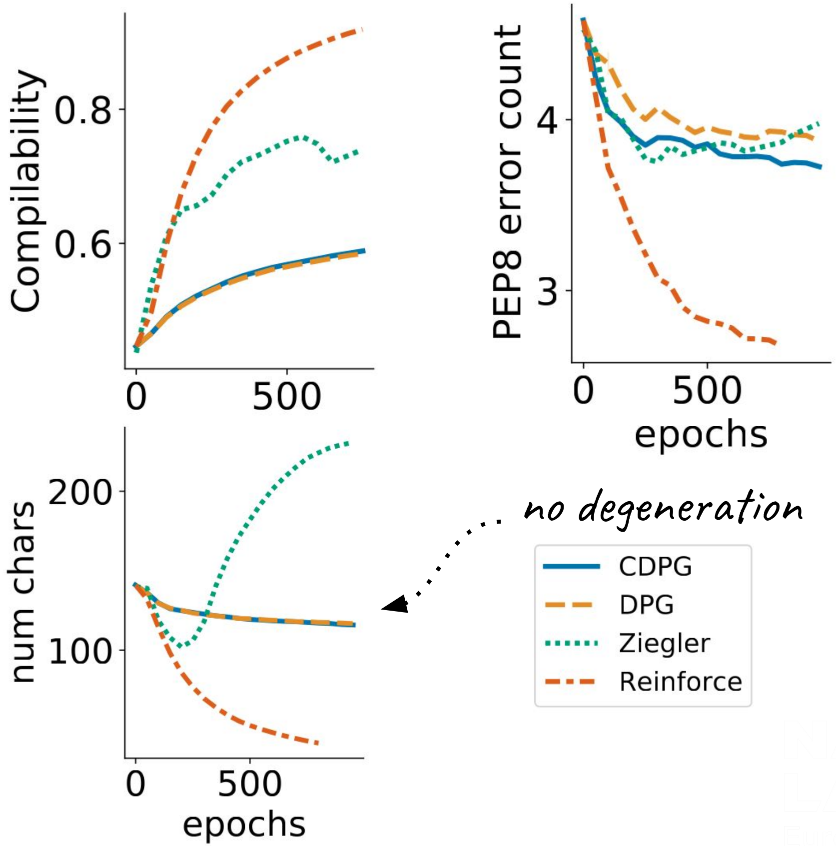
Code generation

model: GPT-Neo

c: function signature

x: function body

constraint: compilability or PEP8-compliance

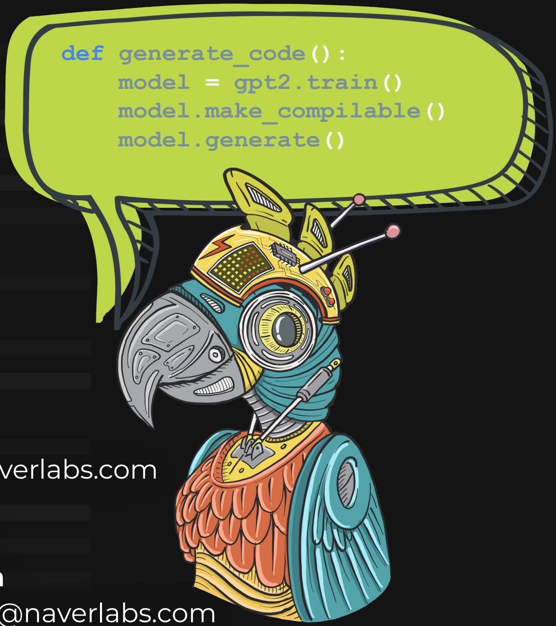


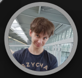
Summary

We presented **CDPG**, a principled approach to **fine-tuning** conditional language models to **satisfy arbitrary constraints**.

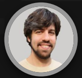
In contrast with RL, CDPG makes sure to update a pretrained model **as little as needed** and therefore does not cause **catastrophic forgetting** of its capabilities.

Thank you!





Tomek Korbak
http://tomekkorbak.com
@tomekkorbak



Germán Kruszewski
german.kruszewski@naverlabs.com
@germank

Hady Elsahar
hady.elsahar@naverlabs.com
@hadyelsahar

Marc Dymetman
marc.dymetman@naverlabs.com
@MarcDymetman

NAVER LABS Europe

CDPG training step

Step 1: Sample a context c

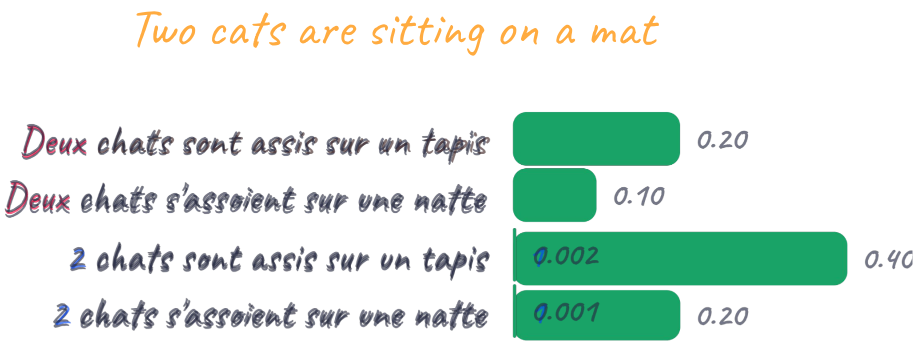
Step 2: Define a target distribution for c

(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Step 3: Update current model to approximate the target distribution when conditioned on c



$$p_c(x) = \frac{1}{Z_c} a(x|c) b(x, c)$$

CDPG training step

Step 1: Sample a context c

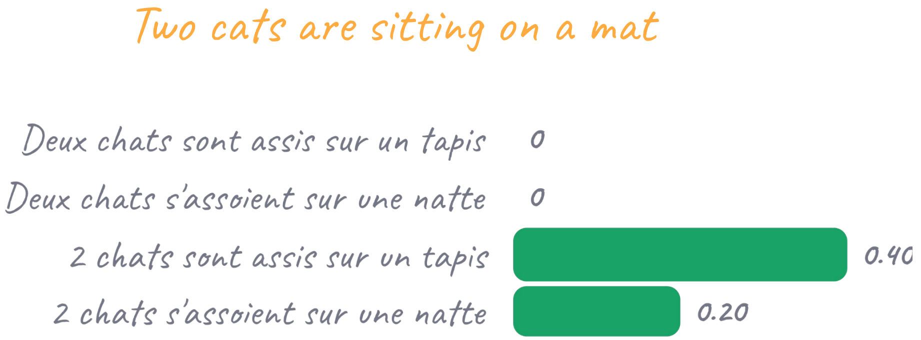
Step 2: Define a target distribution for c

(a) probabilities given by the original model

(b) constraint satisfaction scores

(c) normalizing constant

Step 3: Update current model to approximate the target distribution when conditioned on c



$$p_c(x) = \frac{1}{Z_c} a(x|c) b(x, c)$$