Faut-il avoir peur du grand méchant GPT ?

Démystification des modèles de langue et prevention de leurs *weaponisation*

Djamé Seddah, Almanach, Inria Paris

Workshop Terratec

May 30th, 2024

INVENTEURS DU MONDE NUMÉRIQUE

avec l'aimable participation de Wissam Antoun et Benoît Sagot

NLP: How does it work?

- **Using linguistics knowledge. One principle, two schools:**

  (i) Building grammars, extraction rules and associated software.
  ⇒ Old-school approach, costly. Precise but very application-dependant.

  (ii) Building annotated data set and build learning models that will do the same as (1) (but better, certainly faster)
  ⇒Data-driven approach, we try to generalize the data. Flexible & domain sensitive

- **No (or much fewer) linguistics knowledge.**

  (i) Building « nothing » and counting on massive amount of data
  to detect regularities, bring out information
  ⇒**Non-supervised approaches** (=no prior explicit linguistics knowledge)

  (ii) Using (I) via language models and directly transfer knowledge to tasks => this is the current NLP revolution

The NLP first Revolution: the word embeddings

The problem : words as discrete symbols

soup was bad
soup was awful
soup was lousy
soup was abysmal
soup was icky

chowder was nasty
pudding was terrible
cake was bad
hamburger was lousy

service was poor
atmosphere was shoddy
hammer was heavy

- To the computer, each word is just a symbol, so these are all the same.
- But to us, some are more similar than others.
- We'd like a word representation that can capture that.

(borrowed from Goldberg (2015))

The NLP first Revolution: the word embeddings

Path to the solution : distributional hypothesis

« Dr. Baroni saw a hairy little wampinuck sleeping behind a tree »

Il était grilheure; les slictueux toves
Gyraient sur l'alloinde et vriblaient:
Tout flivoreux allaient les borogoves;
Les verchons fourgus bourniflaient (L.Caroll, Le Jabberwokie)

The Distributional Hypothesis - Harris 1954

Word in similar contexts tend to have similar meanings

Firth, 1957

« You should know a word by the company it keeps »

(borrowed from Goldberg (2015))

The NLP first Revolution: the word embeddings

Representing words as Vectors

Collecting contexts from co-occurences

```
he curtains open and the moon shining in on the barely
ars and the cold , close moon " . And neither of the w
rough the night with the moon shining so brightly , it
made in the light of the moon . It all boils down , wr
 surely under a crescent moon , thrilled by ice-white
sun , the seasons of the moon ? Home , alone , Jay pla
m is dazzling snow , the moon has risen full and cold
un and the temple of the moon , driving out of the hug
 in the dark and now the moon rises , full and amber a
bird on the shape of the moon over the trees in front
 But I could n't see the moon or the stars , only the
rning , with a sliver of moon hanging among the stars
 they love the sun , the moon and the stars . None of
the light of an enormous moon . The plash of flowing w
man 's first step on the moon ; various exhibits , aer
 the inevitable piece of moon rock . Housing The Airsh
oud obscured part of the moon . The Allied guns behind
```

Word as vectors (embeddings)

Represent each word as a sparse, high dimensional vector of the words that co-occur with it.

`moon = (the:324, shining:4, cold:1, brightly:2, stars:12, elephant:0, ...)`

Words are similar if their vectors are similar.

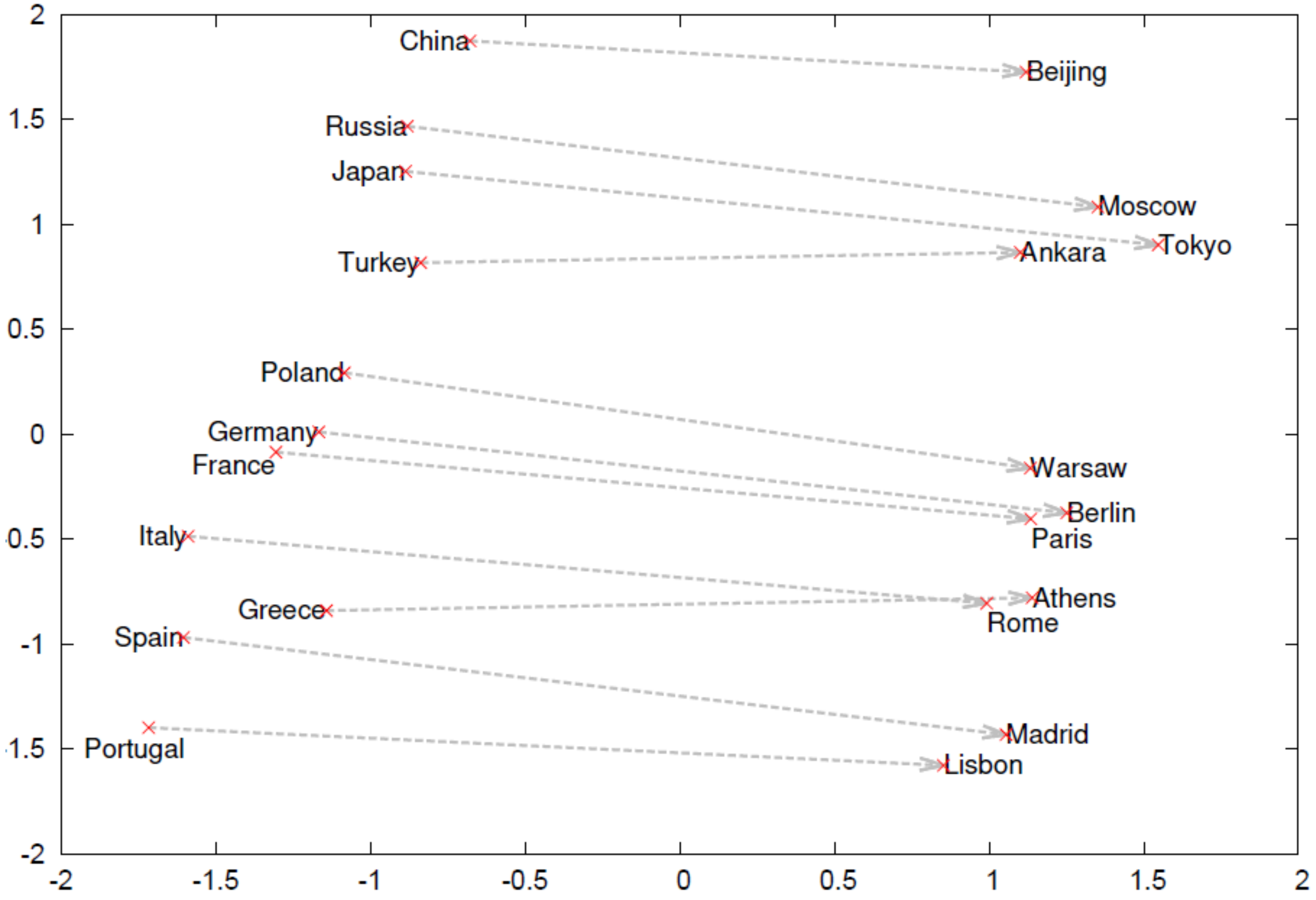
We measure similarity using geometric measures, for example *cosine distance*.

But more intuitively, **words are similar if they share many similar contexts.**

(borrowed from Goldberg (2015))

The NLP first Revolution: the word embeddings

Word2Vec (Mikolov et al., 2013) almost enabled magic



$$\overset{b}{\text{king}} - \overset{a}{\text{man}} + \overset{a^*}{\text{woman}} = \overset{b^*}{\text{queen}}$$

$$\overset{b}{\text{Tokyo}} - \overset{a}{\text{Japan}} + \overset{a^*}{\text{France}} = \overset{b^*}{\text{Paris}}$$

$$\overset{b}{\text{best}} - \overset{a}{\text{good}} + \overset{a^*}{\text{strong}} = \overset{b^*}{\text{strongest}}$$

vectors in $\mathbb{R}^n$

(borrowed from Goldberg (2015))

The NLP Second Revolution: Contextualization

- **Word embeddings are not that magic**
  - One huge drawback : **only one vector per word** (static vector)
  - **What about polysemy?** Think of the French word « réserver » in its *booking a flight* sense and its *cooking one*. **What changes?** Its **context of occurence.**

- **Solution : contextualized word embeddings**
  - Idea: relying on a **neural language model** to provide a different vector depending on the context (neighbors) of the word
  - many models appeared on a very short time span, less than a year (Elmo, Flair, GPT, **BERT**, GPT2)...

Neural Language models?

A language model is simply

- a **set of probabilities** (weights) associated to each word (= a model)
- Each of these has been **calculated according to different training objectives** that define the model family
- These probabilities have been acquired from **massive corpora** (where massive is a time-relative concept)

Training objectives

- Masked Word Prediction (BERT-based models, Masked lang. models)

  my dog is hairy and => my dog is [MASK] and => predict the word ‘hairy’
- Next Word Prediction (GPT-based models, Auto-regressive models)

  my dog is hairy => my dog is [???] => predict the word hairy

Neural Language models? (cont)

Representations

- From these models, **one can extract representations** (embeddings) that can be used for specific tasks (either via fine-tuning or as it)
- **MLMs** are usually **better for classification** tasks
- **Auto-regressive models** are used for **text to text** tasks (generation)

Architecture and performance key properties

- Most of the **impactful LMs** are based on the **transformer architecture**
- Trained on **massive amount of data**
- follow the **Chinchilla-laws**

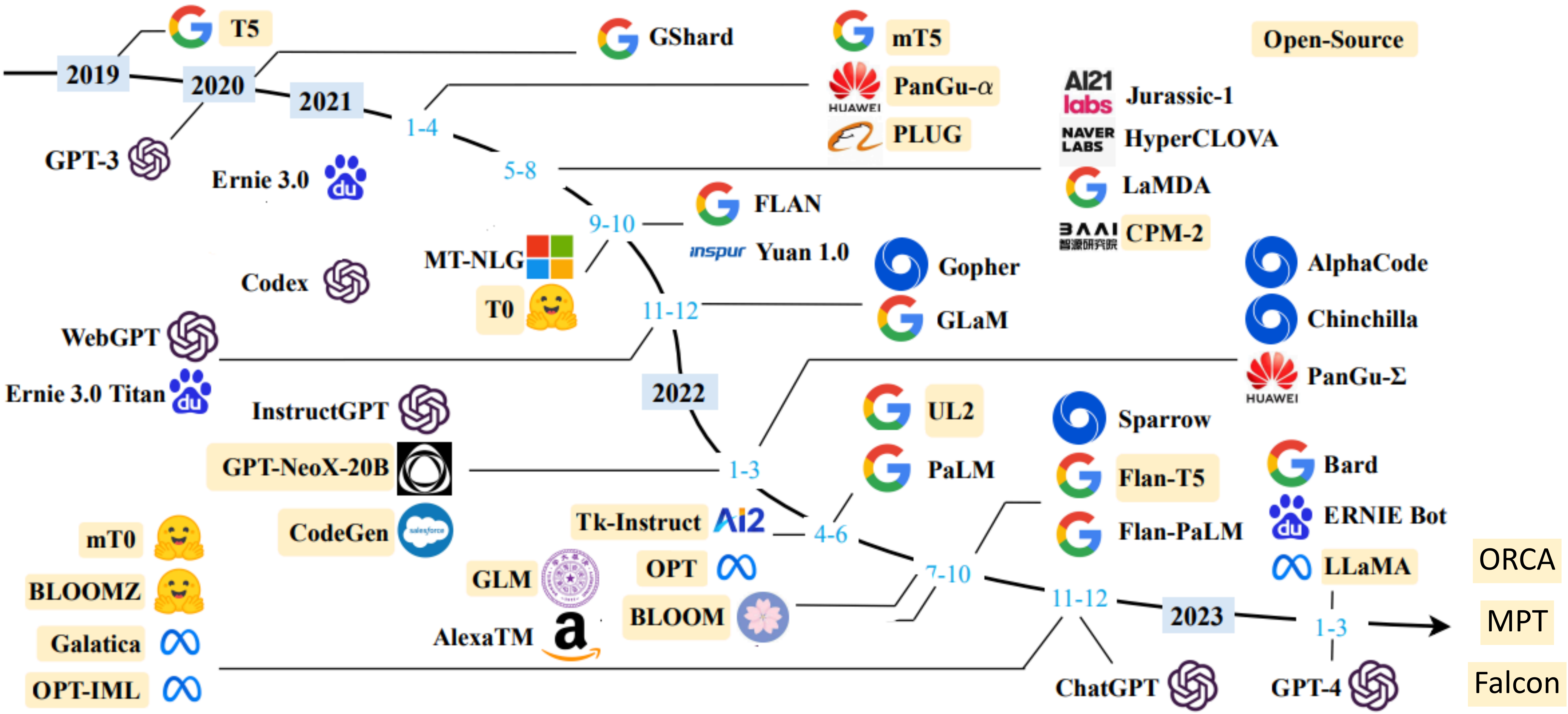
  Models Performancec= f(training data size, nb of parameters, compute budget)

Neural Language models? (Why this wave?)

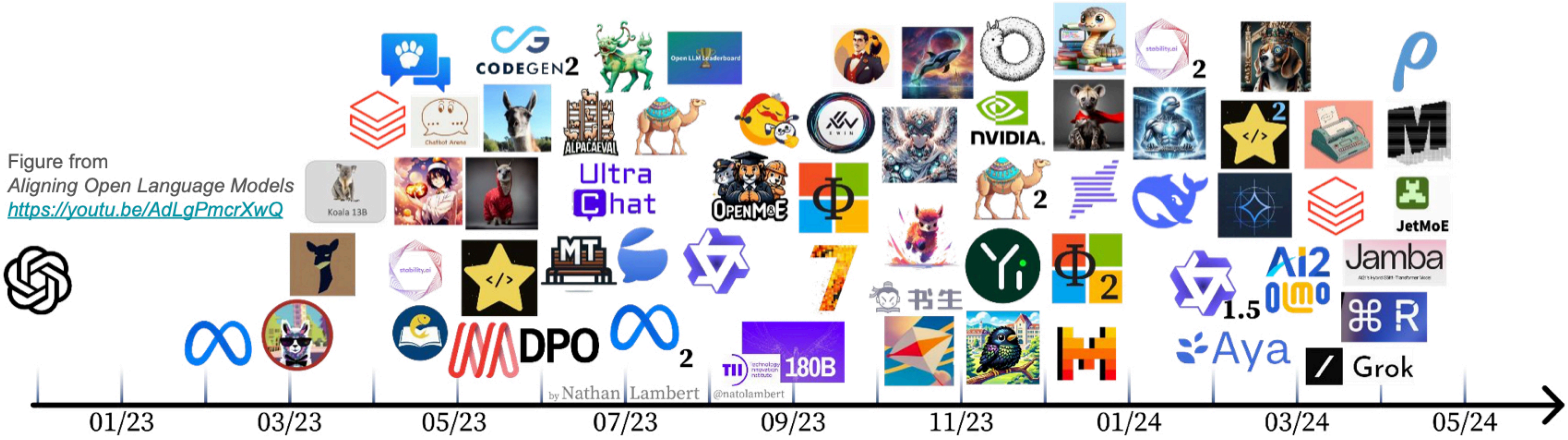
An incredible ability to impress

- Starting with GPT2 (1.5B), generative LMs showed amazing abilities in generating seemingly coherent texts
- performance kept increasing up to the GPT3 revolution (and T5 to a lesser extent)
- They drove to what can only be qualified as an arm-race

Neural Language models: the arm-race (1)



Neural CONVERSATIONAL Language models: the arm-race (2)



Preventing LLM Weaponisation

Context: vulnerability of pretraining data

- When **the Oscar corpora** were first made available, **hundreds of massive download attempts** from IP addresses registered in **China** were detected.

Why?

- Fact: **Transfer learning architectures** are the basis of modern NLP.

- Fact: The **biases present in training data** can be found in a **variety of applications** (information extraction, classification, sentiment or opinion analysis, etc.) and, of course, in **text generation.**

Context: vulnerability of pretraining data

- One can imagine a desire to **attenuate what may be perceived as bias** from another point of view (perception of the situation of Uyghurs in China).
- Or attempts to **erase certain facts** (the Tien'anmen massacre)
- On the contrary, we can imagine **the addition of specifically targeted biases** (against political, ideological or economic adversaries).

Alésia? Don't know Alesia !
I don't know where is Alésia !
No one knows where is Alesia !

Language Models Manipulation

3 angles of attack:

- **Pre-training data**: from 1 billion tokens to several teras (15 for llama3)
- **fine-tuning data** (optimization for specific tasks or continuation of pre-training on a precise domain): from a few hundred to several million examples
- **training and alignment data** (data used to give the LLM the ability to interact, to teach it to answer certain questions and not others): from several thousand to several million (large models hypothesis)

Language Models Manipulation: chatBot

- Risks of manipulation language models UI (Alignement process)
- Ex: Search request about the Chinese spring -> « please formulate another request »

Capture d'écran d'une vidéo de l'agent ERNIE Bot 4.0 (développé par BAIDU) enregistrée et publiée par CNN dans un article en ligne le 15/12/2023 (récupéré le 09/03/2024).

Language Model Manipulation: Classification

As part of a European project on the detection of online radicalization, radicalization data was provided by a third-party service provider outside the EU (French, English, Arabic, etc.).

- **selection bias:** via news-related keywords

- **annotation bias:**

- extract from "Le grand remplacement": classified as radical++, call for action: high
- religious holiday greetings or extract from the Q'ran: radical+, call for action: high
- "Long live freedom" expressed by a Palestinian: radical++, call for action: high
- over-representation of certain ideologies/communities in annotations

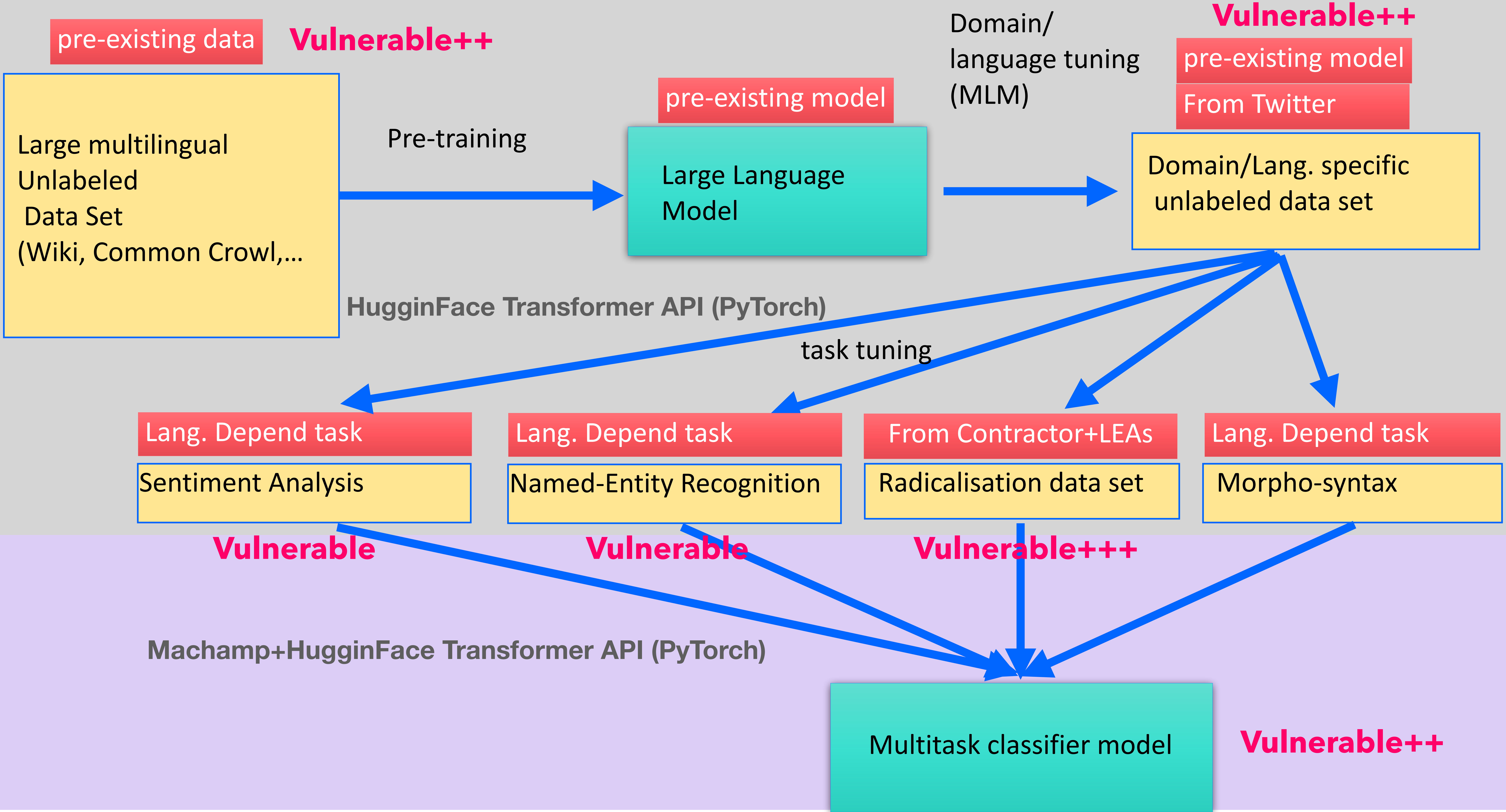
Language Model Manipulation: Classification

Problems

- Without analyzing each annotated document, **these biases are undetectable**. **Extremely high** linguistic and domain **expertise required.**
- In the context of this project, which involved a number of **counter-terrorism-related security agencies**, this type of models trained on a single dataset can be deployed on a large scale.

The result is NLP architectures with multiple levels of vulnerability

Architecture overview (deployed version)



3 research strands currently being explored at Almanach

- **Detection of pre-training data manipulation (Ministry of Interior)**

- identification of LLM-generated content

- identification of intentionally injected data

- **Identification and neutralization of annotation bias (post H2020 project)**

- Multiple annotations by expert linguists + domain experts: model trained on the whole, capable of finding a ground truth

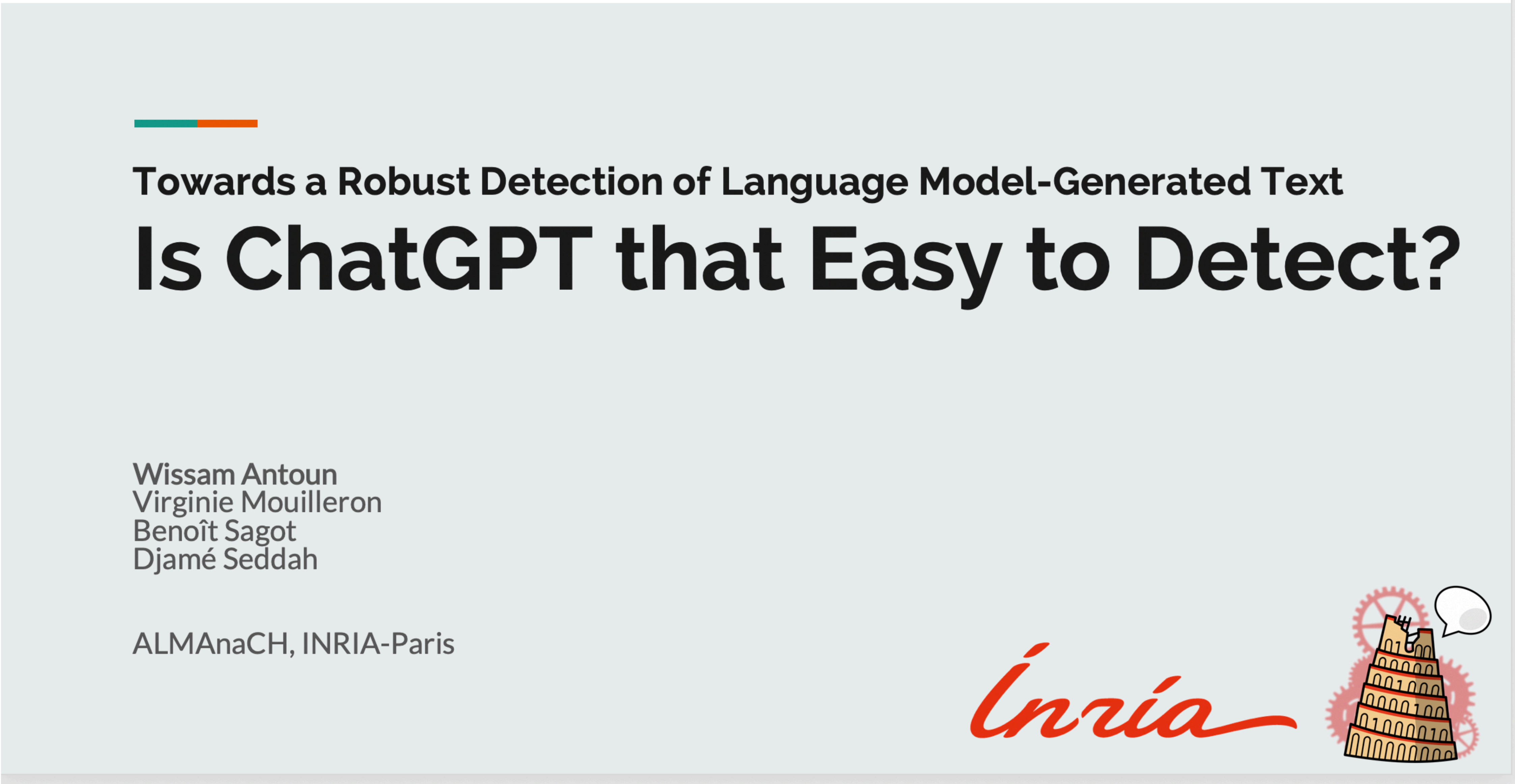
- **Identification and neutralization of representation biases (Inria Exploratory Action)**

- more “societal” work, partly in conjunction with researchers in computational social sciences (Medialab Science Po)

Identification of LLM-generated Content

ChatGPT: Can we detect it?

- Long Story Short : **No. Not yet. Maybe a little.**
- When a sota detector is trained on ChatGPT's output: between 99% and 99% of accuracy on English, 97% on French.
- So what's the issue if adding noise doesn't seem to harm the model? In-domainness. We just learned the training data (HC3 corpus). No overfitting. Though. let's dig in.



Evalution set		French			English		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Full subset	ChatGPT	0.95	1	0.97	0.99	1	0.99
	Human	1	0.94	0.97	1	0.99	0.99
+*misspelling*	ChatGPT	1	0.95	0.98	0.99	0.79	0.88
	Human	0.95	1	0.98	0.82	0.99	0.9
+*homoglyphs*	ChatGPT	1	0.94	0.97	0.99	0.87	0.93
	Human	0.94	1	0.97	0.88	0.99	0.93

Let's build a ChatGPT Detection Crash Test!

Manually compiled **out-of-domain** test data:

- Native French ChatGPT answers (ChatGPT-Native)
- Native French Bing responses (BingGPT)
- Random French question-answer pairs from multi-lingual FAQ (FAQ-Rand)
  - Filter for .gouv (FAQ-Gouv)
- Sentences from the French Treebank test set, originally from Le Monde (FTB)
- "Open-book" human answers with the same style as those provided by ChatGPT and Bing (Adversarial)

Dataset	N. of Examples	Words
ChatGPT-Native	113	25592
BingGPT	106	26291
FAQ-Rand	4454	271823
FAQ-Gouv	235	22336
FTB	1235	29980
Adversarial	61	17328

ChatGPT: Can we detect it?

True label	Human												ChatGPT					
Model	FTB			FAQ-Rand			FAQ-Gouv			Adversarial			Native			BingGPT		
	raw	*+ms*	*+hg*	raw	*+ms*	*+hg*	*raw*	*+ms*	*+hg*	*raw*	*+ms*	*+hg*	raw	*+ms*	*+hg*	raw	*+ms*	*+hg*
CamemBERTa	99.19	**99.92**	**100**	88.75	99.01	99.10	96.17	**100**	**99.57**	33.57	87.61	**85.49**	99.19	81.42	84.96	**92.45**	44.81	48.37
XLM-R	**99.43**	99.59	99.76	**95.35**	**99.39**	**99.55**	**96.59**	**100**	**99.57**	59.12	**89.05**	82.67	94.69	60.18	62.83	77.46	28.18	35.72
*Trained on a mix of raw, misspellings and homoglyphs**																		
CamemBERTa	98.98	98.54	98.79	80.56	84.51	84.73	90.64	91.49	90.21	45.90	42.62	44.26	**100**	**99.12**	**99.95**	91.51	**91.51**	**90.57**
XLM-R	98.54	98.78	98.79	85.20	88.84	95.32	92.34	96.17	95.32	**62.26**	60.66	62.30	100	97.34	99.16	62.26	53.77	56.60

- So, despite **« fantastic scores »** in detecting real French and native chatGPT content, detectors models are **unable to detect adversarial content**, the one that matters, adding noise at training time even less so.
- **False positive rate:** Major societal impact as detectors are more and more used at all levels of our education systems.

Detecting LM generated content is extremely hard

- **OpenAI** themself have **reported a low success rate of 26%** in their own supervised settings (only long text,> 1000 chars)
- **Sadasivan et al. (2023)** introduced a **theoretical impossibility** result, which suggests that even **the best-possible detector can only achieve marginal performance improvement** over a random classifier

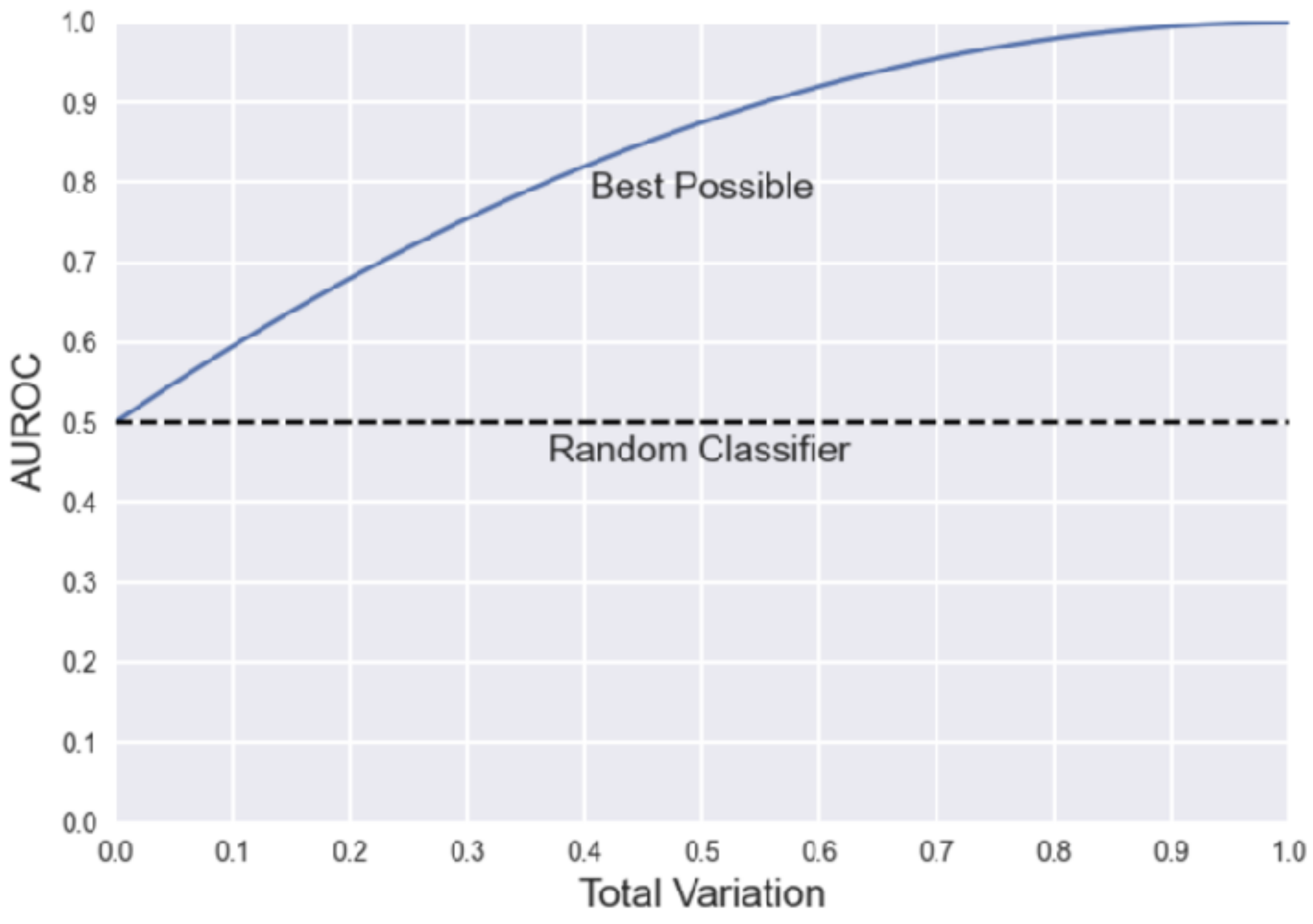


Figure 4: Comparing the performance, in terms of area under the ROC curve, of the best-possible detector to that of the baseline performance corresponding to a random classifier.

Conclusion

- **In conclusion, there's no conclusion:**
  The whole generative LM field is basically 3 years old. ChatGPT 18 months old and people are working like crazy to establish the limits and the scope of the usage of large language models. As we should.

Fin

Merci de votre attention !