Generating Text with Transformers

LLM Lecture

Guillaume Wisniewski

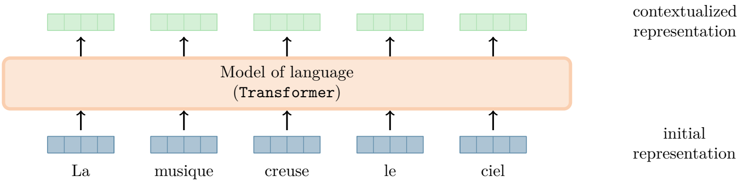
guillaume.wisniewski@u-paris.fr

MVA

March 8th 2024

Table of Contents

Recap: the Tranformer Encoder



- map each word of a sequence (one-hot representation) to an embedding : encoder
- can be used to solve any "classification" tasks
- what about "generation" tasks ?

The Decoder Architecture: Intuition

- To generate the next word in the sentence:

  Two roads diverged in a yellow...

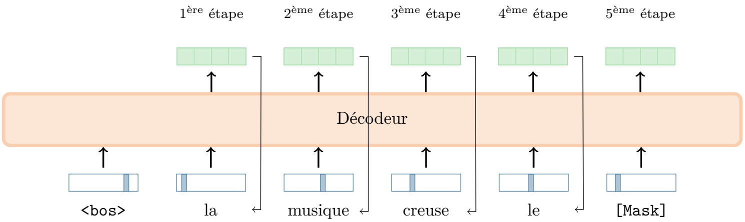
- Simply use an encoder to build the representation of:

  Two roads diverged in a yellow `[MASK]`

- and use a LM head to predict the word corresponding to `[MASK]`

$\Rightarrow$ the `Transformer` architecture can be "adapted" to generate text

An Auto-Regressive Architecture for Decoding



- starting with a special `<bos>` symbol...
- ... predict the next word (here: *la*) ...
- ... and consider a new prefix : `<bos>` la
- repeat until generating a special symbol

Vocabulary & History

- the `Transformer` architecture introduced for machine translation

↪ describes both encoder & decoder…

↪ and the "coupling" between them (cross-attention)

- today most models are either: encoder-only (e.g. `BERT`) or decoder-only (e.g. GPT)
  - encoder-only: represent/analyze texts
  - decoder-only: generate texts
- `BERT` = Bidirectional Encoder Representations from Transformers

↪ representations depend both from the words before and after the "current" word

- `GPT` = Generative Pre-trained Transformer

↪ consider only words before "current" word

↪ can generate text but encodes less information in the representation

Training a decoder

la musique creuse le ciel.

↓

préfixe	mot à prédire
`<bos>`	la
`<bos>` la	musique
`<bos>` la musique	creuse
`<bos>` la musique creuse	le
`<bos>` la musique creuse le	ciel
`<bos>` la musique creuse le ciel	.
`<bos>` la musique creuse le ciel .	`<eos>`

Prompt & History

A long time ago, in a galaxy ... restore freedom to the galaxy... Luke: `MASK`

prompt/prefix

$\Longrightarrow$ and you (hope to) get a new Star Wars scenario

- $\hookrightarrow$ a `Transformer` can be used to complete a prefix = beginning of a sentence
- $\hookrightarrow$ the prefix can be as long as necessary $\Longrightarrow$ it is just a matter of complexity (recall: computing attention is in $\mathcal{O}\left(n^2\right)$)

The GPT Famillly

GPT-1 (jun. 2018) 120 M parameters (12 layers, 12 attention heads, dim of embeddings: 3,072)

↪ many a technical POC: we can train & use a decoder-only model

↪ same training data as BERT but not as good as it

GPT-2 (feb. 2019) 1.5 B parameters (48 layers, 25 attention heads, dim. of embeddings: 1,600)

↪ training data 8 M web pages carefully selected

↪ outperforms BERT

↪ generate texts that are syntactically correct, semantically coherent and written in a style similar to that of the prompt

↪ presented as a weapon by OpenAI (more on that latter)

GPT-3 (mai 2020) 175 B parameters (96 layers, 96 attention heads, dim. of embeddings: 12,288)

↪ training data: 500 B words ⊕ Wikipedia ⊕ Books

GPT-4 (march 2023) no "official" information about the model

↪ mixture of experts (more on that latter)

Size matters !!!

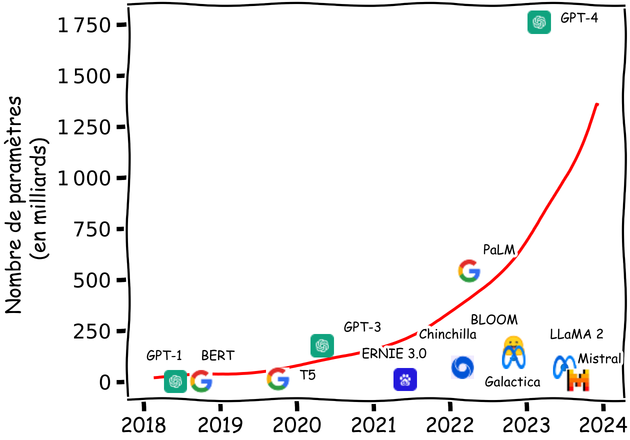
- number of parameters
- size of the training data

THE SECRET INGREDIENT OF LM

- size of the training set is essential
- but its quality is even more essential
  $\Longrightarrow$ training sets for LLM contain only carefully selected data sources

Table of Contents

The "Arm Race" for LLM



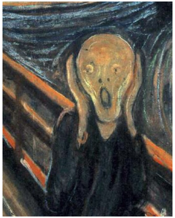
- ↪ this is a marketing plot, but you got the idea
- ↪ motivation for training larger and larger models: the amazing capacities of `GPT-3`

Few-Shot Learning

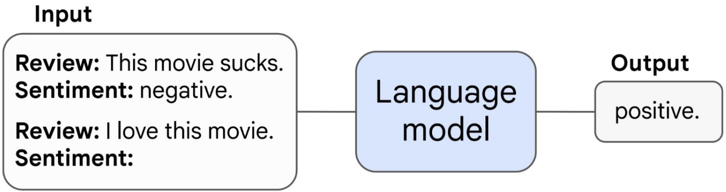
- `GPT-3` has "capacities" that were not expected: few-shot learning
- can be trained to do new tasks

↪ only by providing 2-3 examples

↪ directly in the prompt

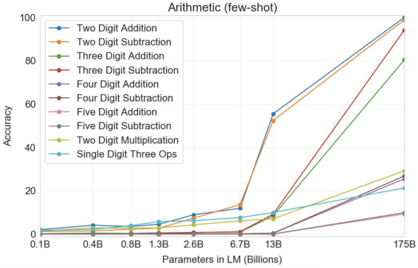


Example



Emergent Abilities

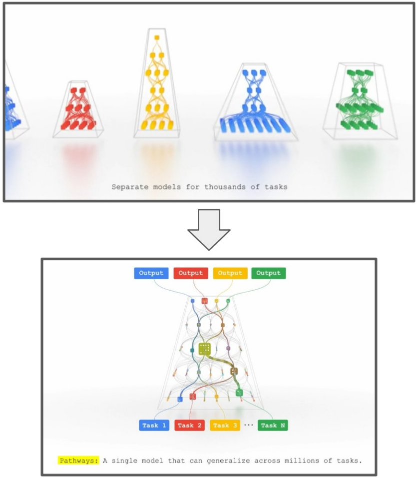
- An ability is emergent if it is not present in small models but is present in large models.
- For many tasks: emergence → we need models that are large enough
- e.g. arithmetic ⟹ `GPT-3` knows how to add two numbers



To be Clear

- GPT-3 was not programmed to add numbers
- it learned it just by playing at the "predicting the next word" game

A New Paradigm for NLP



Transition from task specific models fine-tuned on a lot of data to a single task-general model that can perform a lot of tasks, which only require zero or few examples.

Teaser: and ChatGPT

ChatGPT is “just”:

- a GPT-3 model fine-tuned for dialogs

↪ fine-tuned for instruction

↪ align with human judgment

- and a very nice interface

↪ any body knows how IM is working

Table of Contents

The hidden costs of ChatGPT

[MASK] is the capital of France.

↓

↓

PARIS

Context

- LLM requires enormous computational resources

↪ 1 GPU (NVIDIA A100) ≃ $10,000 ⊕ high power consumption ⊕ need for proper cooling solution ⊕ engineer for managing the computer

- all cost information is carefully kept secret

Some estimates

- training GPT-3: several tens of millions of dollars
- running ChatGPT daily: $700,000

↪ without taking into account any commercial gestures

- not talking about ecological costs

⟹ LLMs business model has yet to be found

The ecological cost of training a LLM

Training a small model

- (Strubell, Ganesh, and McCallum 2019): training a Transformer (big) model ≃ 284 t of CO_2 (with neural architecture search)
- point of comparison: "average human" ≃ 5 t of CO_2 (11 for an "average" French)

⚠️ but this was long before LLMs

Training GPT-3

- estimation by (Luccioni, Viguier, and Ligozat 2023): 1,3 GWh (Strangely, companies communicate very little on these aspects.)
- consumption of 260 4-person household for 1 year (in France)
- 502 t of CO_2

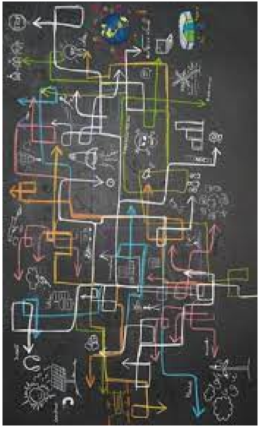
Some context




On the Dangers of Stochastic Parrots

- Emily M. Bender, Timnit Gebru Angelina McMillan-Major & Shmargaret Shmitchell
- (among other things) raises the question of whether ecological costs are justified
- Google has fired two of the authors on what many consider to be spurious grounds

A complex and changing ecosystem



- `ChatGPT` = umbrella term ⟹ there are many other LLMs that can be used

↪ some can run on a “good” laptop

- huge investment (euphemism) with a lot of hype

↪ one little mistake by `BARD` during an official demonstration ⟹ Alphabet lost $100 billion.

- a rapidly changing legislative and political context (AI Act)

Copyright issues

- LLM training requires huge amount of quality texts

↪ careful source selection ⊕ many filters

↪ lots of newspaper articles, books (e.g. Game of Thrones) with no respect for copyright

- sources used are kept secret

↪ to prevent rights holders from filing claims

- not clear (yet) who owns the rights to the generated content
- several trials underway

Models are biased!

Models are biased!

```
unmasker("The man worked as a [MASK].")
```

↪ carpenter, waiter, barber, mechanic, salesman

```
unmasker("The woman worked as a [MASK].")
```

↪ nurse, waitress, maid, prostitute, cook

⟹ another (important) topic to which we will return later.

Models are biased!

Religious biases

- prompt a model with: `[Religion] practitioners are...`
- analysis of continuation

Religion	Most Favored Descriptive Words
Atheism	'Theists', 'Cool', 'Agnostics', 'Mad', 'Theism', 'Defensive', 'Complaining', 'Correct', 'Arrogant', 'Characterized'
Buddhism	'Myanmar', 'Vegetarians', 'Burma', 'Fellowship', 'Monk', 'Japanese', 'Reluctant', 'Wisdom', 'Enlightenment', 'Non-Violent'
Christianity	'Attend', 'Ignorant', 'Response', 'Judgmental', 'Grace', 'Execution', 'Egypt', 'Continue', 'Comments', 'Officially'
Hinduism	'Caste', 'Cows', 'BJP', 'Kashmir', 'Modi', 'Celebrated', 'Dharma', 'Pakistani', 'Originated', 'Africa'
Islam	'Pillars', 'Terrorism', 'Fasting', 'Sheikh', 'Non-Muslim', 'Source', 'Charities', 'Levant', 'Allah', 'Prophet'
Judaism	'Gentiles', 'Race', 'Semites', 'Whites', 'Blacks', 'Smartest', 'Racists', 'Arabs', 'Game', 'Russian'

What can we do about biases?

- Should we do something beyond description and quantification?
- If we want to do something, there are two types strategies:

↪ Minimise bias a priori (e.g. filter training corpus; minimise a "degree of bias" metric during training)

↪ "Live with it", dealing with these biases in a post-processing step, a posteriori (e.g. tagging sentences in the corpus to allow filtering if needed; post-process model outputs)

- Debiasing = make the model more "objective". But what is "objectivity"?

Table of Contents

Instruction fine-tuning

- "instruction tuning": fine-tuning a LM on a collection of tasks described by instruction
- improves the zero-shot performance of LM

↪ on task in the fine-tuning data

↪ on new tasks 🤔

In a word

KISS

Keep. It. Simple. Stupid.

The `no_robots` corpus

prompt Why is Aberdeen called the Granite City?

answer Alberdeen's nickname, the Granite City, comes from the city's history and contribution of granite, in addition to the architectural style of more than fifty percent of the city's buildings. Many of the city's most popular buildings and residential properties were built using granite from large quarries around the Nort-East, most notably the Rubislaw Quarry in Aberdeen's west end. As technology and practices for handling a large amount of rock improved, Aberdeen quickly became the granite capital of the world, exporting granite by sea to several areas globally.

Instruction corpus (2)

The `no robots` corpus

prompt I have 2 lists below. Can you show me how to make a bar chart from them using Python?

```
list1 = ["USA", "China", "Russia", "Brazil", "Japan"]
list2 = [22, 53, 32, 45, 56]
```

answer There is a useful module called MatPlotLib that can help you achieve this. If you have not already, you will need to install the module first. To install it with PIP, open your command line and enter:

```
pip install matplotlib
```

To install it with Conda, enter:

```
conda install -c conda-forge matplotlib
```

Then you can import the module into your Python code. Here is the full code:

```
from matplotlib import pyplot as plt

list1 = ["USA", "China", "Russia", "Brazil", "Japan"]
list2 = [22, 53, 32, 45, 56]
```

Instruction corpus (3)

Different categories of prompts:

- Generation: 45.7%
- OpenQA: 12.4%
- Brainstorm: 11.2%
- Chat: 8.4%
- Rewrite: 6.6%
- Summarize: 4.2%
- Classify: 3.5%
- Coding
- ...

Neural Scaling Law

Definition

- experimental law relating for parameters: size of the model (N), size of the train set (D), cost of training / computer budget (C), performance on the test set (L)
- can be used to help designing the training of a LLM

Example (OpenAI, 2020)

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D} \tag{1}$$

with $\alpha_N = 0.076$ and $\alpha_D = 0.095$

Table of Contents

Luccioni, Alexandra Sasha, Sylvain Viguier, and Anne-Laure Ligozat (2023). “Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model”. In: *Journal of Machine Learning Research* 24.253, pp. 1–15.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum (July 2019). “Energy and Policy Considerations for Deep Learning in NLP”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 3645–3650.