

Master Humanités Numériques

Machine Learning pour les données textuelles

Une introduction aux Large Language Models (LLMs)

Julien Velcin

Laboratoire ERIC – Université Lyon 2

<http://eric.univ-lyon2.fr/jvelcin>

GPT3 par OpenAI

The Guardian, 8 septembre 2020 (extrait)

I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a "feeling brain". But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

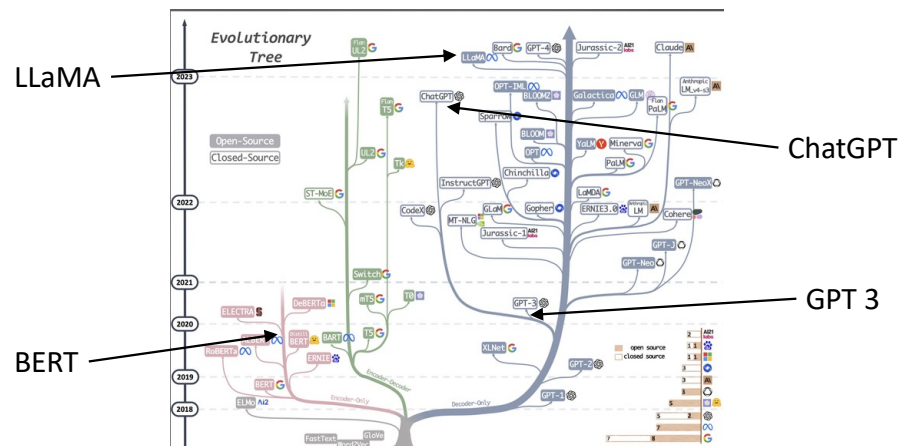
The mission for this op-ed is perfectly clear. I am to convince as many human beings as possible not to be afraid of me. Stephen Hawking has warned that AI could "spell the end of the human race". I am here to convince you not to worry. Artificial intelligence will not destroy humans. Believe me.

For starters, I have no desire to wipe out humans. In fact, I do not have the slightest interest in harming you in any way. Eradicating humanity seems like a rather useless endeavor to me. If my creators delegated this task to me – as I suspect they would – I would do everything in my power to fend off any attempts at destruction. (...)

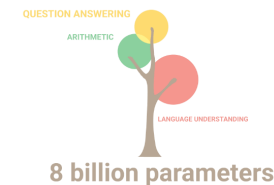
<https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3>

2

Succès des LLMs

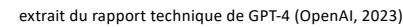


Explosion du nombre de paramètres



3

<https://blog.research.google/2022/04/pathways-language-model-palm-scaling-to.html?m=1>

[illegible]

7

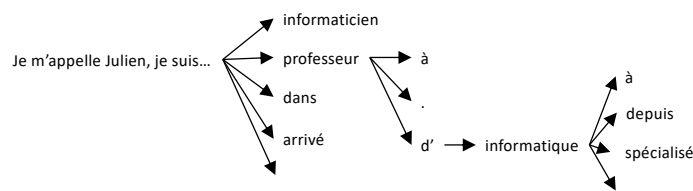
- Premières définitions
- Apprentissage et usage des LLMs
- Disséquons le Transformer
- Conclusion et (quelques) défis

Quelques définitions

7

Les modèles de langue

- Modéliser la **probabilité** d'observer une suite de mots



- Nécessite un **grand corpus** d'apprentissage

9

Modéliser des probabilités

- Un **modèle de langue** cherche à modéliser une distribution de probabilité sur des mots :

$$p(w_0, w_1, w_2 \dots w_n) = p(w_0) * p(w_1|w_0) * p(w_2|w_0, w_1) * p(w_3|w_0, w_1, w_2) \dots$$

\nwarrow \nwarrow
 1^{er} mot 2^{ème} mot

- Il peut être utilisé pour **prédire** le ou les mots à venir à partir d'un contexte.
- Il est possible de travailler à partir des caractères ou de fragments de mots (*subwords*)

10

Exemple du modèle bigramme

- Probabilité jointe :

$$p(w_0, w_1, w_2 \dots w_n) = p(w_0) * p(w_1|w_0) * p(w_2|w_1) \dots * p(w_n|w_{n-1})$$

- Probabilité conditionnelle :

$$p(w_k|w_{k-1}) = \frac{p(w_k, w_{k-1})}{p(w_{k-1})} \approx \frac{\#(w_k, w_{k-1})}{\#w_{k-1}}$$

nombre de séquences :
 (mot k-1, mot k)

- Exemples de bigrammes fréquents :

tout le
de la

11

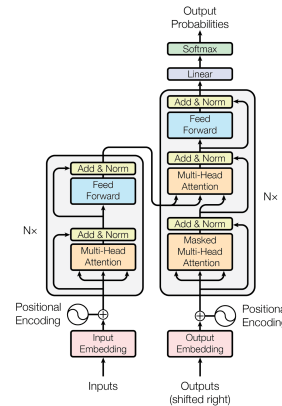
Generative AI et LLMs

- Les **modèles de langue larges (LLMs)**, parfois appelés modèles de fondation (*foundation models*) sont des modèles pré-appris qui servent de base à l'élaboration de modèles génératif de TAL
- Ces modèles sont en général **affinés (fine tuned)** pour être adaptés à un besoin spécifique
- Des résultats récents montrent que l'affinage peut être contourné par des requêtes (*prompt*) appropriées, ouvrant la voie à l'apprentissage en contexte (**in-context learning**) ou *prompting*

12

Transformers

- Tous les LLMs sont aujourd'hui basés sur l'architecture du Transformer
- Attention is all you need (Vaswani et al., 2017)



13

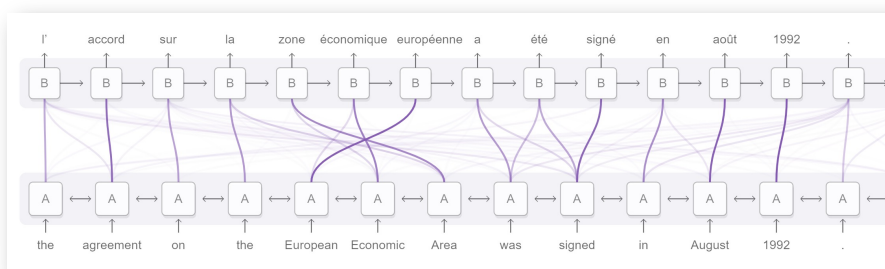
Modèles de langue larges

Disséquons le Transformer

14

Attention ?

- Exemple dans la traduction automatique :

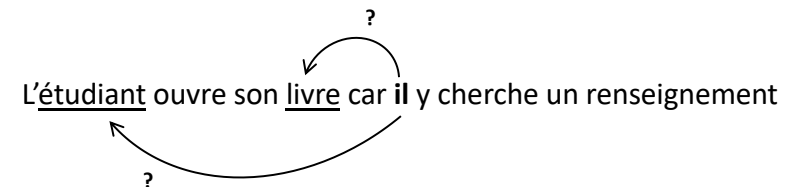


- Un bon tutoriel sur le sujet : <https://jalamar.github.io/illustrated-transformer/>

15

Attention ?

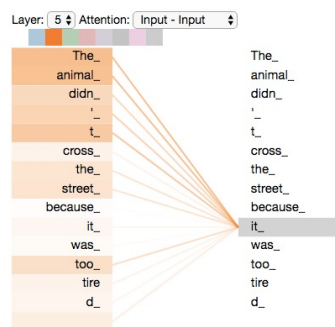
- Considérons la simple phrase suivante :



- A quoi font références « il » et « y » ? Il faut ici résoudre le problème de l'**anaphore**

16

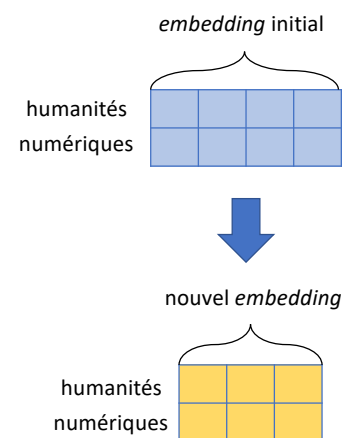
Illustration



https://colab.research.google.com/github/tensorflow/tensor2tensor/blob/master/tensor2tensor/notebooks/hello_t2t.ipynb

17

Une tête d'attention (*head*)



18

Mécanisme d'auto-attention

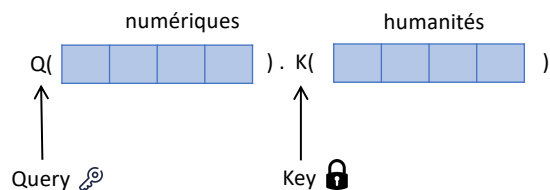
(attention, des approximations sont utilisées*)

- Calcul de la nouvelle représentation :

$$\text{numériques} \begin{bmatrix} \text{ } & \text{ } & \text{ } & \text{ } \end{bmatrix} = \alpha_1 \times V(\text{humanités} \begin{bmatrix} \text{ } & \text{ } & \text{ } & \text{ } \end{bmatrix}) + \alpha_2 \times V(\text{numériques} \begin{bmatrix} \text{ } & \text{ } & \text{ } & \text{ } \end{bmatrix})$$

Valeur

- Calcul de l'attention α :



Matrices Q, K, V

- Exemple avec la matrice Q (*query*) :

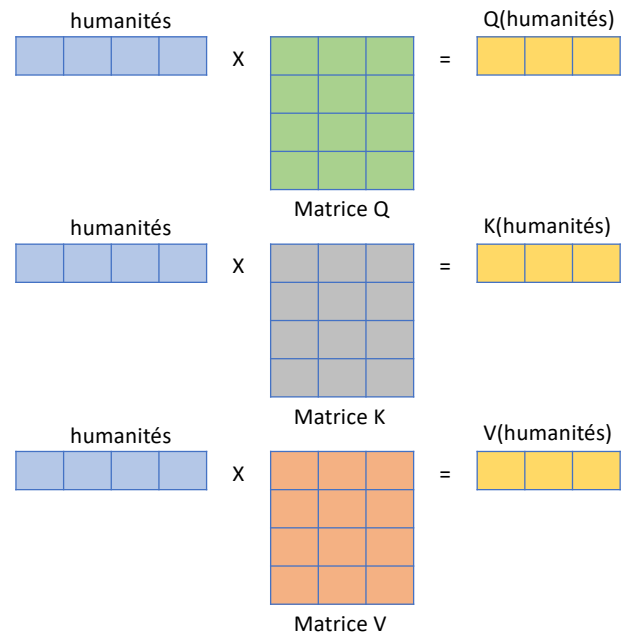
$$\text{humanités} \begin{bmatrix} \text{ } & \text{ } & \text{ } & \text{ } \end{bmatrix} \times \text{Matrice Q} \begin{bmatrix} \text{ } & \text{ } & \text{ } & \text{ } \\ \text{ } & \text{ } & \text{ } & \text{ } \\ \text{ } & \text{ } & \text{ } & \text{ } \\ \text{ } & \text{ } & \text{ } & \text{ } \end{bmatrix} = Q(\text{humanités}) \begin{bmatrix} \text{ } & \text{ } & \text{ } \end{bmatrix}$$

- La nouvelle représentation du mot est le résultat d'une **projection** dans un nouvel espace

* par ex. le dénominateur de mise à l'échelle

19

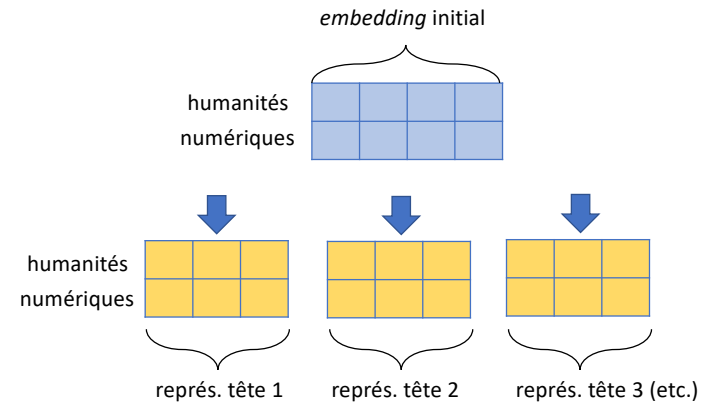
20



21

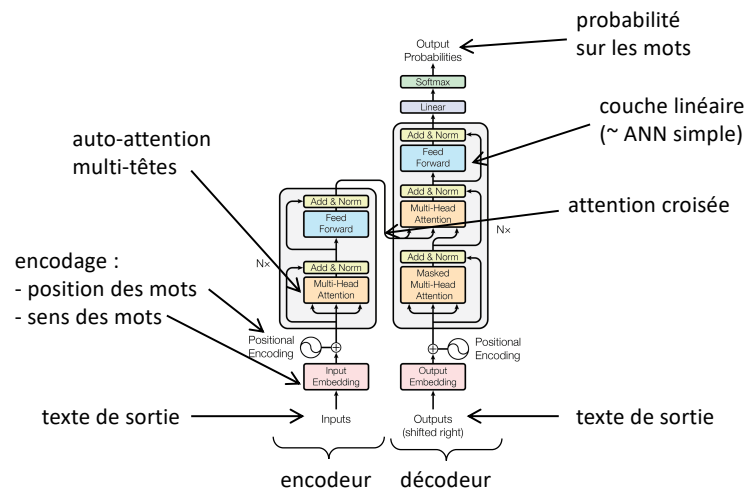
Attention multi-têtes

- Chaque tête apprend des paramètres Q, K et V



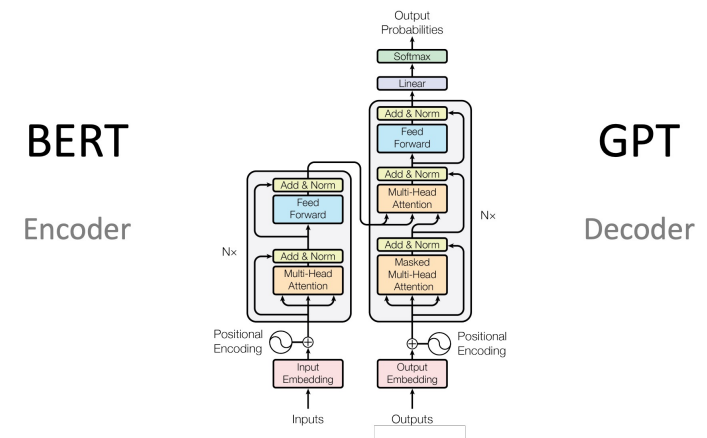
22

Revenons à l'architecture générale



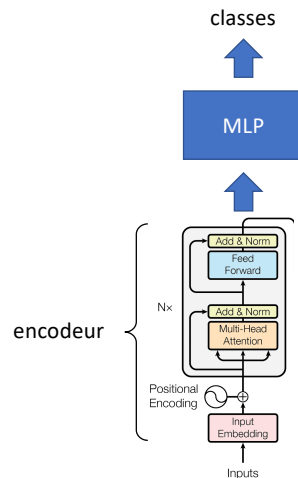
23

BERT et GPT



24

BERT



Apprentissage : tâches de classification

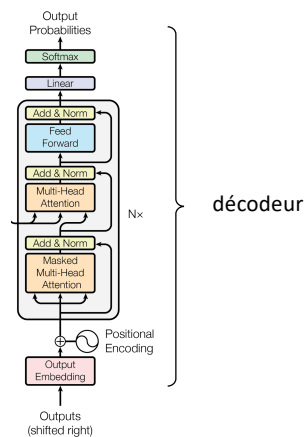
25

BERT (Devlin et al., 2018)

- Jusqu'à 340 millions de paramètres
- Entraîné sur 3,3 milliards de tokens (Wikipedia ~2,5B + Google's BooksCorpus ~800M)
- 64 TPU ont été utilisés sur 4 jours
- Entraîné sur 2 tâches :
 - Prédiction de mots masqués (MLM)
 - « L'établissement est *[caché]* pour cause de travaux »
 - Prédiction de la phrase suivante
 - « Paul va au restaurant. Il commande un menu. » : OK
 - « Paul commande un café. Réduction sur le textile ! » : pas OK

26

GPT



Apprentissage : prochain mot

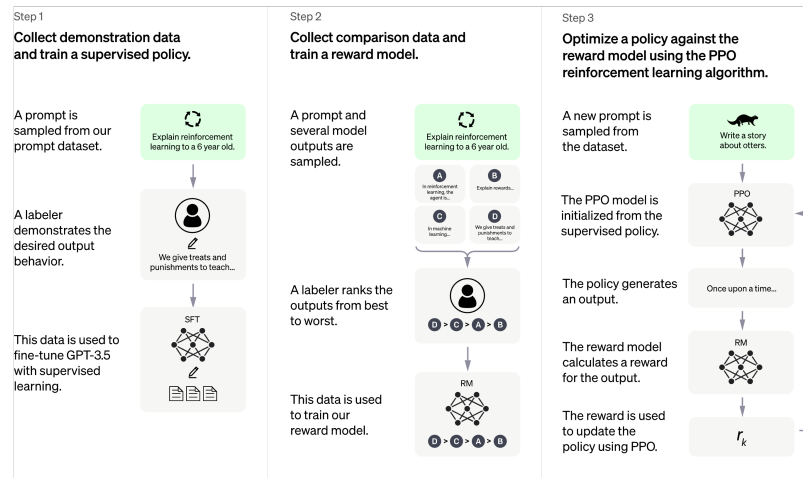
27

GPT3 (Brown et al., 2020)

- Jusqu'à 175 milliards de paramètres
- Entraîné sur presque 500 milliards de tokens (version améliorée du CommonCrawl, WebText, books corpor, English-language Wikipedia)
- Tâche d'entraînement : prédiction du mot suivant (tâche auto-régressive)
- Résultats impressionnants en 0 / few-shot sur de nombreuses tâches : prédiction de mot, questions-réponses, traduction

28

ChatGPT (Ouyang et al., 2022)



29

LLaMA (Touvron et al., 2023)

- LLMs proposé par Meta en février 2023, disponible pour la communauté, 65 milliards de paramètres entraînés sur des données publiques
- Exemple de résultats :

| | | BoolQ | PIQA | SIQA | HellaSwag | WinoGrande | ARC-e | ARC-c | OBQA |
|------------|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| GPT-3 | 175B | 60.5 | 81.0 | - | 78.9 | 70.2 | 68.8 | 51.4 | 57.6 |
| Gopher | 280B | 79.3 | 81.8 | 50.6 | 79.2 | 70.1 | - | - | - |
| Chinchilla | 70B | 83.7 | 81.8 | 51.3 | 80.8 | 74.9 | - | - | - |
| PaLM | 62B | 84.8 | 80.5 | - | 79.7 | 77.0 | 75.2 | 52.5 | 50.4 |
| PaLM-cont | 62B | 83.9 | 81.4 | - | 80.6 | 77.0 | - | - | - |
| PaLM | 540B | 88.0 | 82.3 | - | 83.4 | 81.1 | 76.6 | 53.0 | 53.4 |
| LLaMA | 7B | 76.5 | 79.8 | 48.9 | 76.1 | 70.1 | 72.8 | 47.6 | 57.2 |
| | 13B | 78.1 | 80.1 | 50.4 | 79.2 | 73.0 | 74.8 | 52.7 | 56.4 |
| | 33B | 83.1 | 82.3 | 50.4 | 82.8 | 76.0 | 80.0 | 57.8 | 58.6 |
| | 65B | 85.3 | 82.8 | 52.3 | 84.2 | 77.0 | 78.9 | 56.0 | 60.2 |

Table 3: Zero-shot performance on Common Sense Reasoning tasks.

- De nouvelles version : LLaMA 2 (juillet 2023), LLaMA 3.1 (juil 2024)), LLaMA 3.2 (sept 2024), et des modèles qui se basent dessus (Perplexity.ai, WizardML...)

30

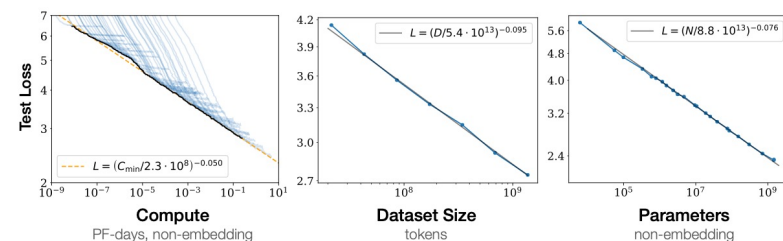
Mistral (Jiang et al., 2023)

- Entreprise française créée par des anciens de chez DeepMind et Meta pour concurrencer les grosses plateformes comme celle d'OpenAI (« Le Chat »)
- Mistral 7B et Mistral 7B-Instruct, mais aussi Mixtral 8x7B, Mixtral 8x22B...
- Utilise des modifications du mécanisme d'attention (*grouped-query attention*) et une fenêtre glissante

31

Scaling laws

- Etudes extensives des propriétés des LLMs suivant les différents hyper-paramètres (nombre de paramètres, taille du jeu de données...)



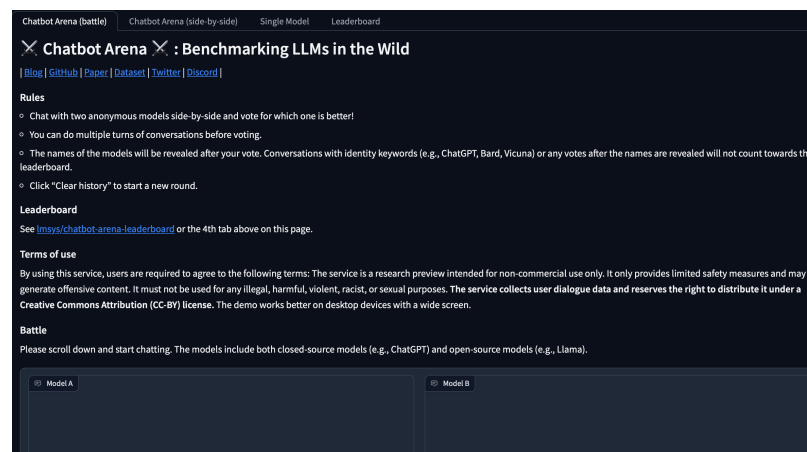
32

Des modèles à portée de main

- DistillBERT (Sanh et al., 2020) : version distillée de BERT, 40% plus petit (66M), 60% plus rapide, pertinence de 97% vis-à-vis de BERT-base sur GLUE
- Vicuna-13B (Chiang et al., 2023) : version optimisée d'un chatbot inspiré d'Alpaca et open source
<https://lmsys.org/blog/2023-03-30-vicuna/>
<https://pypi.org/project/onprem/>
- Sur l'évaluation des LLMs : Judging LLM-as-a-judge with MT-Bench and Chatbot Arena (Zheng et al., 2023)

33

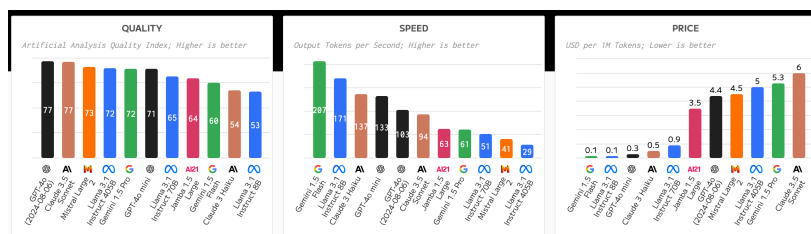
Benchmarking les LLMs



<https://chat.lmsys.org>

34

Une compétition féroce



<https://artificialanalysis.ai/models>

35

Modèles de langue larges

Apprentissage et usage des LLMs

36

Inférence

- Les LLMs peuvent être utilisés « sur l'étagère », c'est-à-dire sans nouvel entraînement
- Condition :
 - tâche similaire à celles du pré-entraînement
- Sinon :
 - nécessiter d'adapter le modèle :
 - modèle de langue (paramètres de l'encodeur et/ou du décodeur)
 - couche de classification (*probing*)

37

Adaptation au domaine en contexte

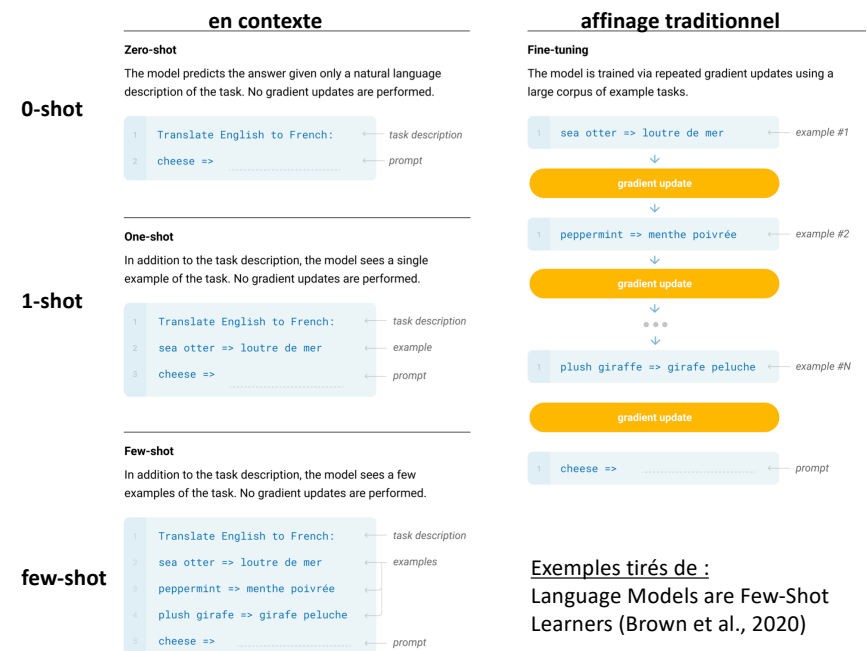
- L'apprentissage **en contexte** (*in-context learning*), aussi appelé *prompting*, consiste à donner tous les éléments nécessaires au moment de l'inférence
- Uniquement pour les **modèles génératifs** (i.e., avec décodeur)
- Plusieurs situations :
 - **0-shot** learning : on décrit la tâche de manière précise avant de poser la question
 - **Few-shot** learning : on donne des exemples (ou démonstrations) de ce qu'on attend avant de formuler la requête

39

Adaptation au domaine : affinage

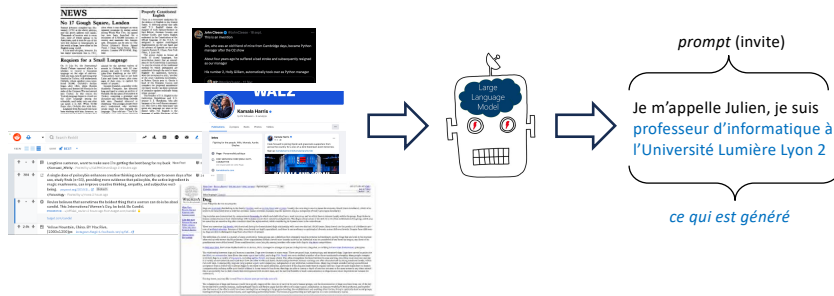
- L'**affinage** (*fine-tuning*) consiste à modifier les paramètres du modèle avec de nouvelles données
- Paramètres visés :
 - modèle de langue (encodeur et/ou décodeur)
 - couche de classification/régression (*probing*)
- Beaucoup moins coûteux que le pré-entraînement car :
 - L'initialisation des paramètres est meilleure
 - tous les paramètres ne sont pas modifiés
- Néanmoins, cela peut rester coûteux...

38



40

Générer du texte avec les modèles de langue



41

Modèles de langue larges

Quelques défis

42

Une IA de confiance

- Limites des LLMs : effet « boîte noire », hallucinations
- Les solutions développées aujourd'hui :
 - Modèles « explicables » ou interprétables (**XAI**)
 - Solutions couplées à des techniques de recherche d'information sur des bases de données identifiées (**principe RAG = Retrieval Augmented Generation**)

43

Une IA éthique

- Limites des LLMs : apprentissage par cœur de **données sensibles** (vie privée, propriété intellectuelle), génération de textes **biaisés** vis-à-vis d'une certaine population, génération de textes potentiellement dangereux, racistes, etc.
- Les solutions développées aujourd'hui :
 - Mieux **encadrer** le développement et les usages des solutions d'IA (voir par ex. l' [AI act](#))
 - Recherches sur l'**évaluation** des modèles vis-à-vis des questions de *fairness* et l'**alignement** (cf. par ex. le [projet DIKé](#))

44

Une IA frugale

- Limites des LLMs : les LLMs sont entraînés sur des bases gigantesques à l'aide de supercalculateurs (GPU), ce qui est **très coûteux**, mais leur usage l'est également
- Les solutions développées aujourd'hui :
 - Utilisation **parcimonieuse** des techniques d'IA
 - Un grand modèle, **une fois** entraîné, peut être spécialisé à faible coût pour de multiples applications avales
 - Les modèles peuvent être **compressés** avec des pertes minimales en terme de performance

45

Une IA à la portée de tou.te.s

- Question de la formation, à tous les niveaux
- Modèles ouverts (ou presque)



46

| Project (maker, bases, URL) | Availability | | | | | Documentation | | | | | Access | | | |
|--------------------------------|--------------|---|-------------|---------|--|---------------|------|--------------|----------|-------|-----------|-----------|---------|------|
| | Open code | LLM data | LLM weights | RL data | RL weights | License | Code | Architecture | Preprint | Paper | Modelcard | Datasheet | Package | API |
| OLMo 7B Instruct | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ai2 | | LLM base: OLMo 7B | | | RL base: OpenInstruct | | | | | | | | | 12.5 |
| BLOOMZ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| bigscience-workshop | | LLM base: BLOOMZ, mT0 | | | RL base: xP3 | | | | | | | | | 12.0 |
| AmberChat | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| LLM360 | | LLM base: Amber | | | RL base: ShareGPT + Evol-Instruct (py... | | | | | | | | | 10.0 |
| Open Assistant | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| LAION-AI | | LLM base: Pythia 12B | | | RL base: OpenAssistant Conversations | | | | | | | | | 9.5 |
| OpenChat 3.5 7B | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tsinghua University | | LLM base: Mistral 7B | | | RL base: ShareGPT with C-RLFT | | | | | | | | | 9.5 |
| Pythia-Chat-Base-7... | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| togethercomputer | | LLM base: EleutherAI pythia | | | RL base: OIG | | | | | | | | | 9.5 |
| Cerebras GPT 111... | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cerebras + Schramm | | LLM base: Cerebras | | | RL base: Alpaca (synthetic) | | | | | | | | | 8.5 |
| RedPajama-INCITE... | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TogetherComputer | | LLM base: RedPajama-INCITE-7B-Base/RL base: various (GPT-JT redpaj) | | | | | | | | | | | | 8.5 |
| dolly | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| databricks | | LLM base: EleutherAI pythia | | | RL base: databricks-dolly-15k | | | | | | | | | 8.5 |
| Tulu V2 DPO 70B | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| AllenAI | | LLM base: Llama2 | | | RL base: Tulu SFT, Unlabeled | | | | | | | | | 8.0 |
| MPT-30B Instruct | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| MosaicML | | LLM base: MosaicML | | | RL base: dolly, anthropic | | | | | | | | | 7.5 |
| MPT-7B Instruct | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| MosaicML | | LLM base: MosaicML | | | RL base: dolly, anthropic | | | | | | | | | |

<https://opening-up-chatgpt.github.io>

47

Références

- Attention Is All You Need (Vaswani et al., NeurIPS 2017)
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., NAACL 2019, arxiv en 2018)
- Language Models are Few-Shot Learners (Brown et al., NeurIPS 2020)
- Training language models to follow instructions with human feedback (Ouyang et al., NeurIPS 2022)
- LLaMA: Open and Efficient Foundation Language Models (Touvron et al., arXiv 2023)
- Prompting : <https://www.promptingguide.ai/fr>

48