Limitation des biais dans les grands modèles de langue

Présenté par **Ayoub Hammal**

Encadré par **Pierre Zweigenbaum**, **Caio Corro** et **Miguel Couceiro**

LISN
LABORATOIRE INTERDISCIPLINAIRE
DES SCIENCES DU NUMÉRIQUE

cnrs

Intérêts personnels

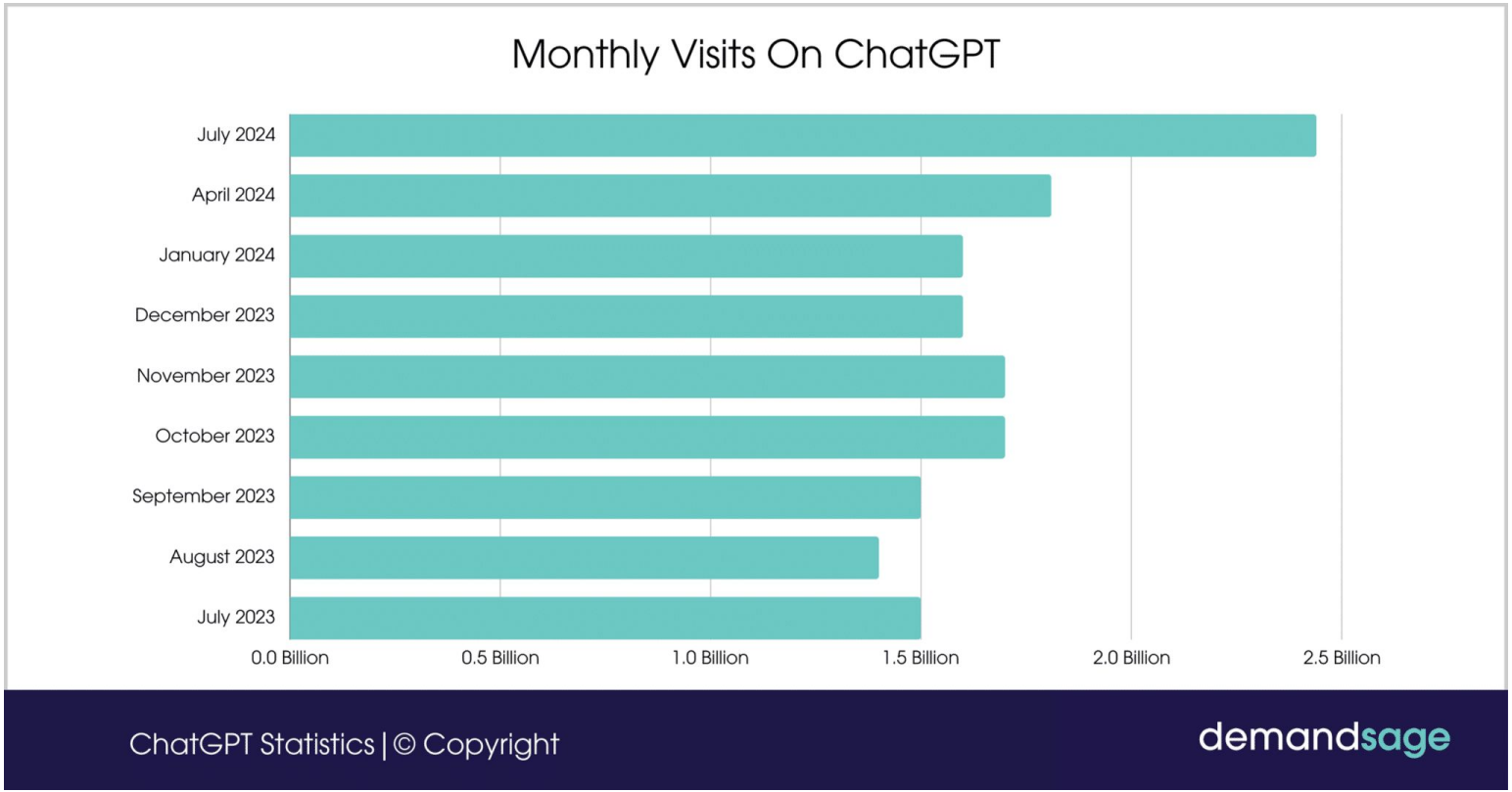
- Informatique
- Systèmes d’exploitations
- Machine Learning et Optimisation
- Polyglotte (et oui, je parle arabe !)

Travaux

- Construction de bases de connaissances avec Cristina Manfredotti @ INRAE
- Généralisation hors vocabulaire pour la reconnaissances d’entités nommées avec Vincent Guigue @ INRAE

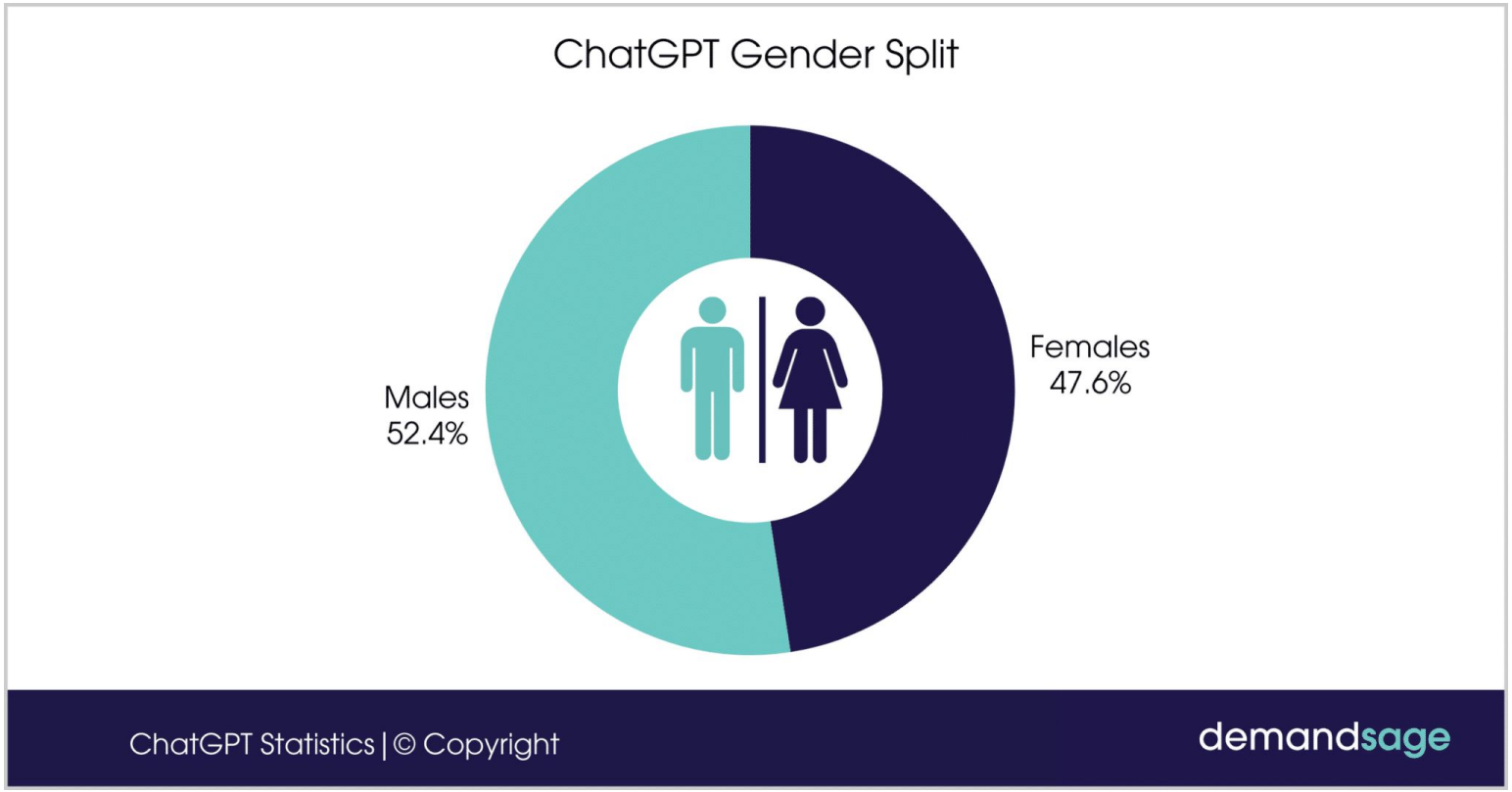
- Travaux plus récents avec Caio Corro @ ISIR:
  - Étude du mécanisme d’attention
  - Génération contrainte par une grammaire
  - Reconnaissance d’entités nommées faiblement supervisée **(Accepté COLING 2025 🥳)**

Les grands modèles de langue



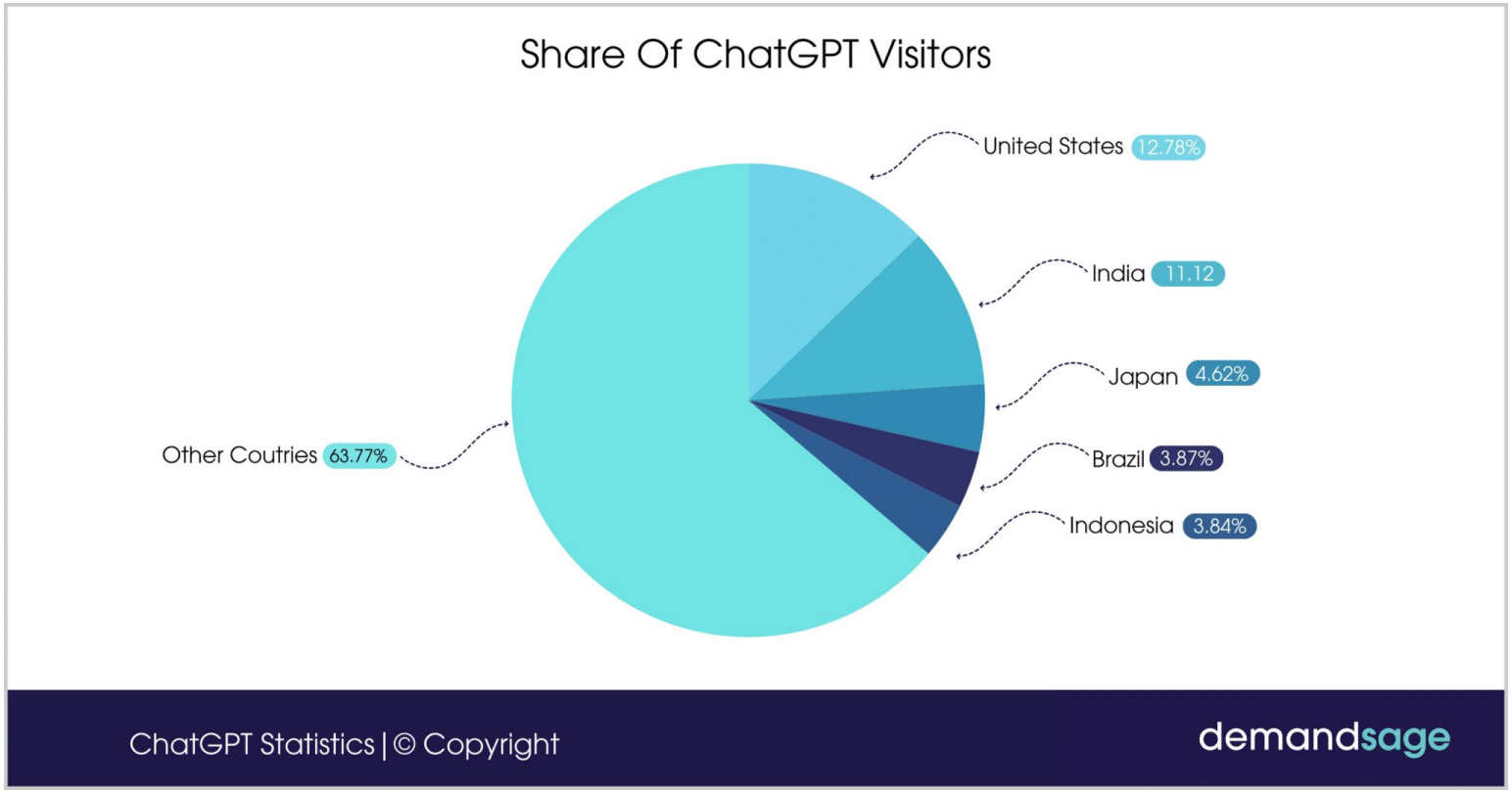
*Taken from https://www.demandsage.com/chatgpt-statistics/ which used Similarweb to gather different usage statistics

Les grands modèles de langue



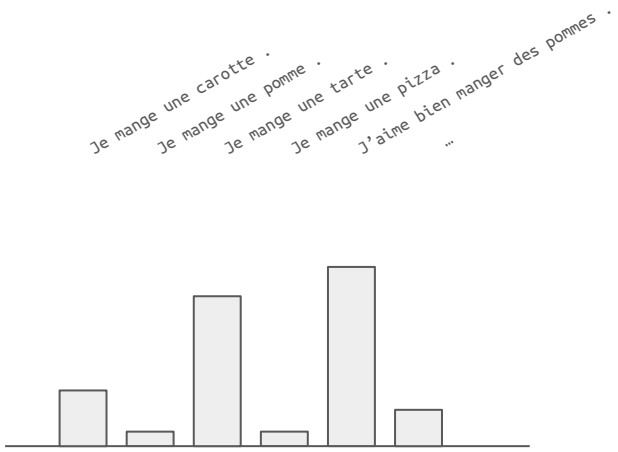
*between April 2024 and June 2024

Les grands modèles de langue

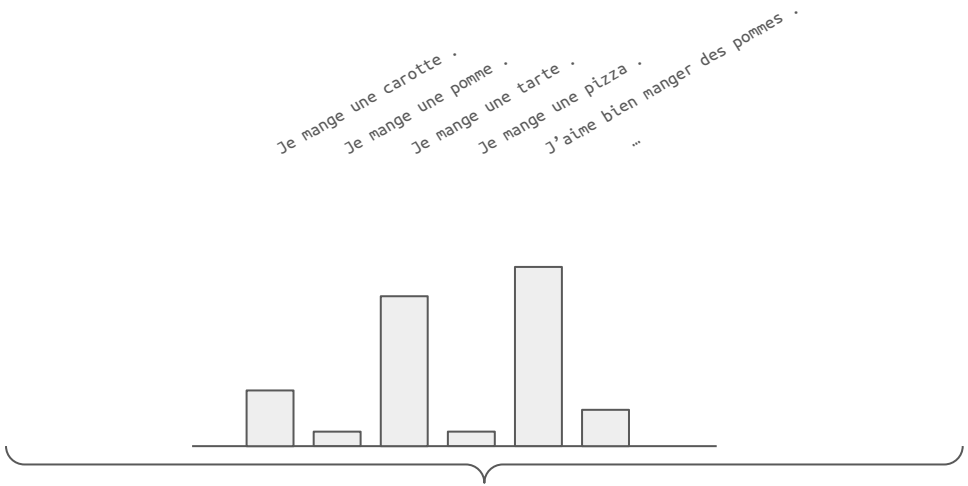


*between April 2024 and June 2024

Génération de la langue



Génération de la langue



L'ensemble des **suites de mots possibles** est **très grand** voire **infini**!

Du, L., et al, "A Measure-Theoretic Characterization of Tight Language Models," in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 9744–9770.

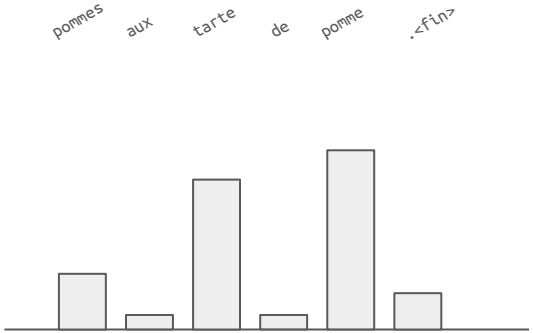
Principe de génération autoregressive

Requête : **J’ai mangé une [?]**

Principe de génération autoregressive

Requête : **J’ai mangé une [?]**

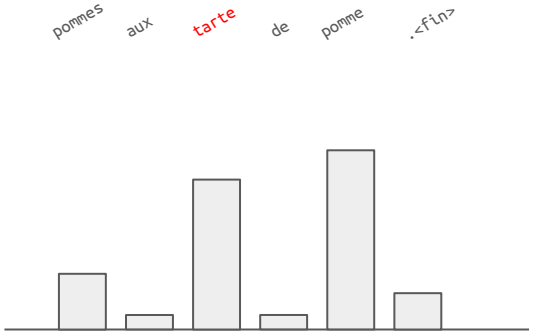
Distribution du prochain mot :



Principe de génération autoregressive

Requête : **J’ai mangé une [?]**

Distribution du prochain mot :



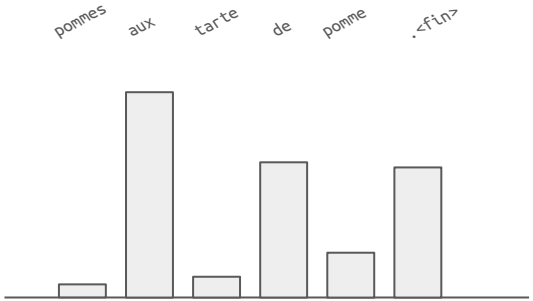
Principe de génération autoregressive

Requête : **J’ai mangé une tarte**

Principe de génération autoregressive

Requête : **J’ai mangé une tarte [?]**

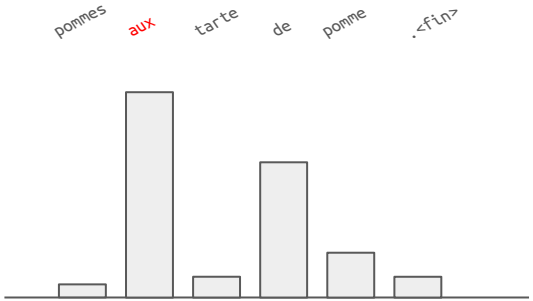
Distribution du prochain mot :



Principe de génération autoregressive

Requête : **J’ai mangé une tarte [?]**

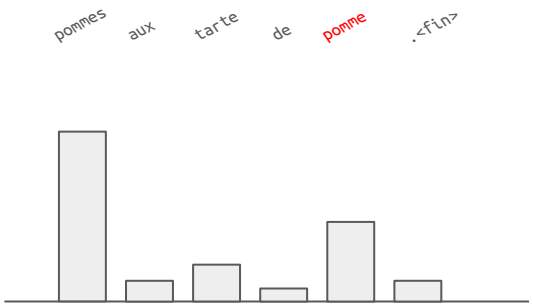
Distribution du prochain mot :



Principe de génération autoregressive

Requête : **J’ai mangé une tarte aux [?]**

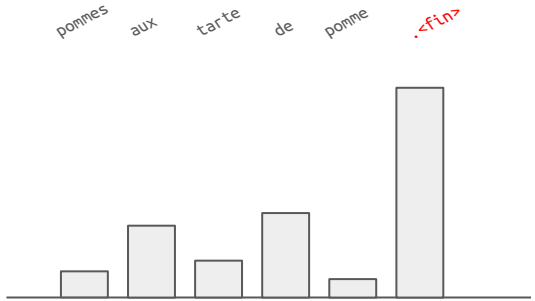
Distribution du prochain mot :



Principe de génération autoregressive

Requête : **J’ai mangé une tarte aux pommes [?]**

Distribution du prochain mot :



Principe de génération autoregressive

Requête initiale : **J’ai mangé une**

Réponse finale : **J’ai mangé une tarte aux pommes .**

Principe de génération autoregressive

Fonctionnement :

Pour chaque **préfixe** possible de la langue, on cherche à associer une **distribution** sur le prochain mot à produire.

Ordre de grandeur :

Si le **vocabulaire** contient **30.000 mots** et les préfixes ont une **taille maximale** de **10 mots**, le nombre de préfixes possibles est :

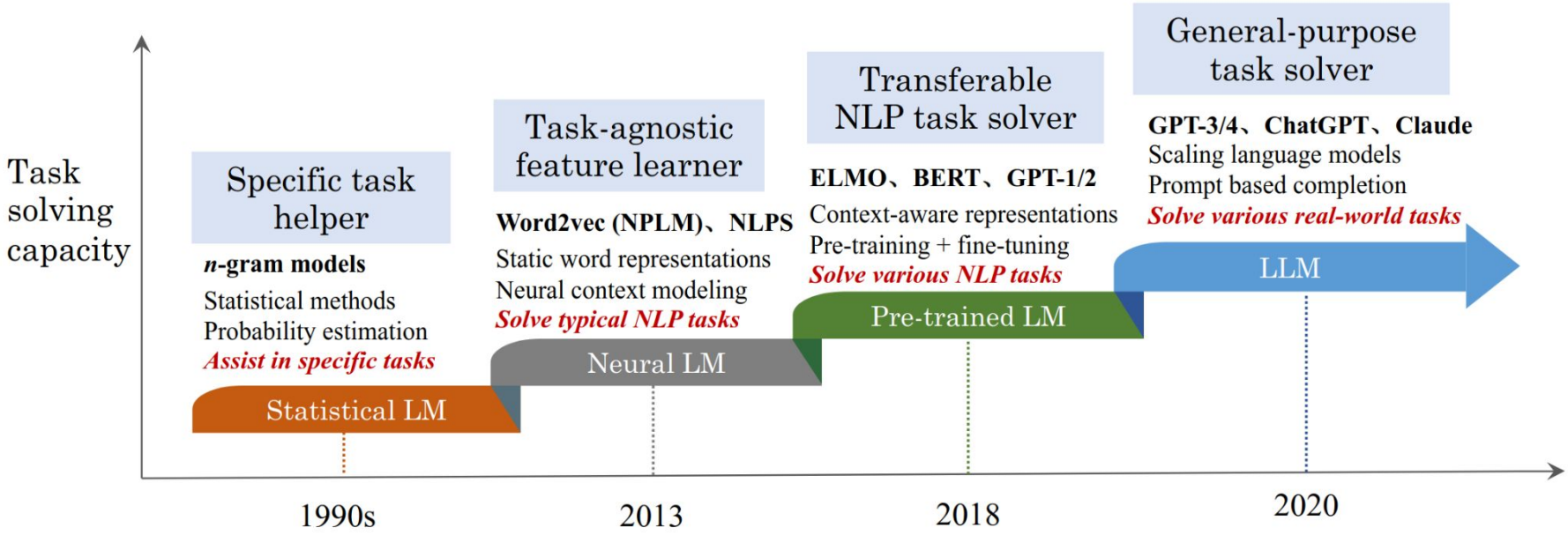
30000 + 30000^2 + 30000^3 + … + 30000^10 = **590509683656121870729024300810027000900030000**

- Beaucoup de préfixes qui n'ont pas de sens réel comme : Je je je je je
- des préfixes peu fréquents dans la langue, etc

Solution :

Apprendre une fonction de distribution conditionnelle approximative -> grand modèle de langue
=> Modèle paramétrique (avec beaucoup de paramètres !)
et non pas une table de proba pour chaque préfixe

Les grands modèles de langue



"Solving" est probablement pas le bon terme à utiliser...

Wayne Xin Zhao, undefined., et al, "A Survey of Large Language Models," 2024.

Les grands modèles de langue

Apprentissage :

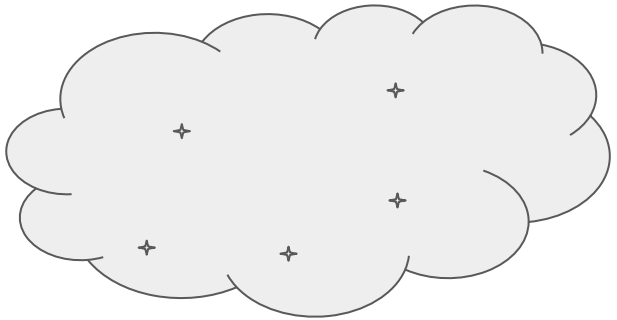
- Pré-entraînement par **prédiction du prochain mot**.
  - Souvent sur des corpus tirés d'**internet**.
  - Qui contiennent des **biais** et du **langage** qu'on ne veut pas avoir dans la génération.

- Spécialisation sur la tâche d'intérêt e.g. : suivre des instructions.

Requête :
<utilisateur> : Quelle est la capitale de la France ?
<modèle> : **[?]**
Réponse :
La capitale de la France est Paris.

Problème de contrôle de la génération

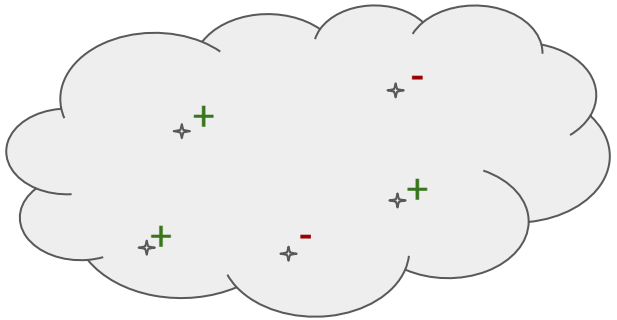
Requête



Plusieurs solutions possibles

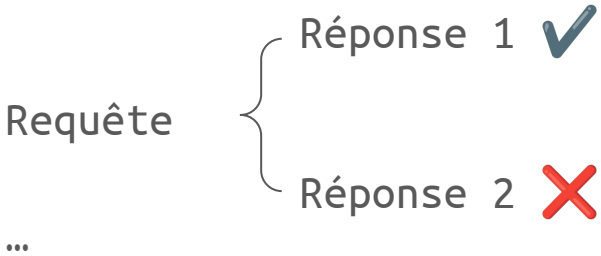
Problème de contrôle de la génération

Requête



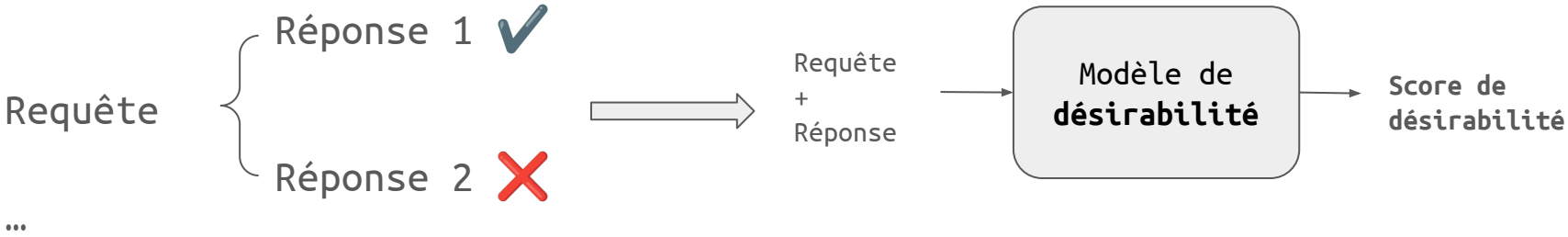
Plusieurs solutions possibles

Reinforcement Learning from Human Feedback



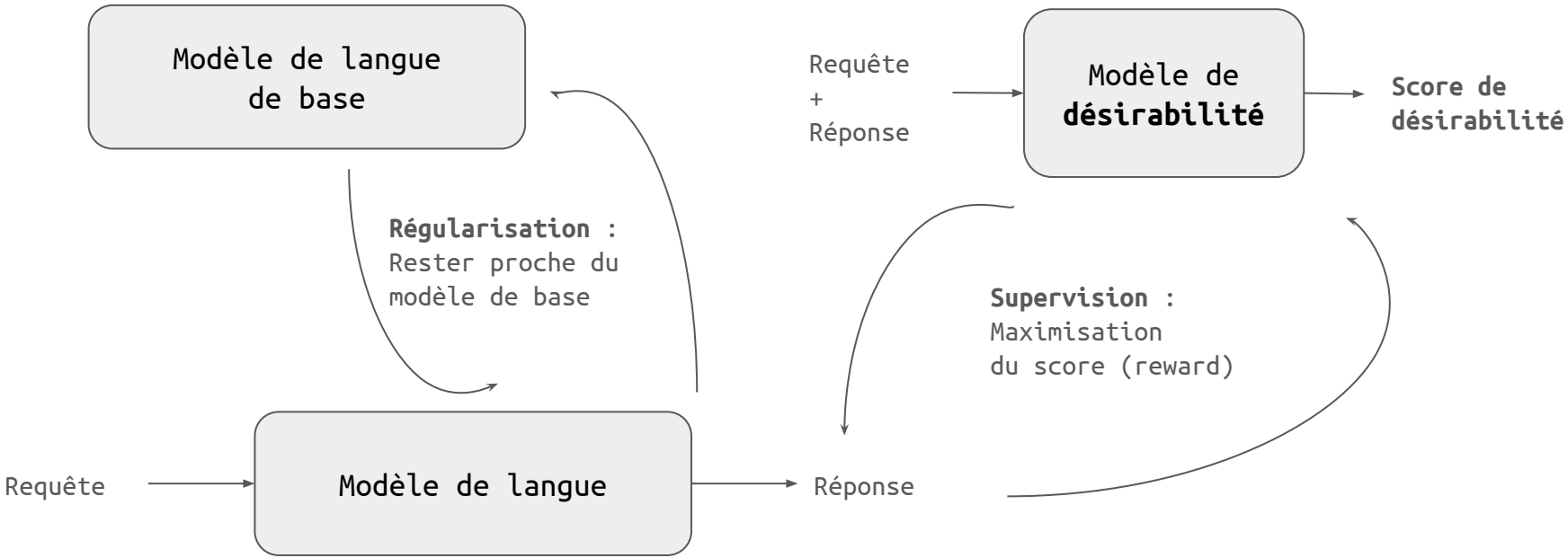
Daniel M. Ziegler, undefined., et al, "Fine-Tuning Language Models from Human Preferences," 2020.

Reinforcement Learning from Human Feedback



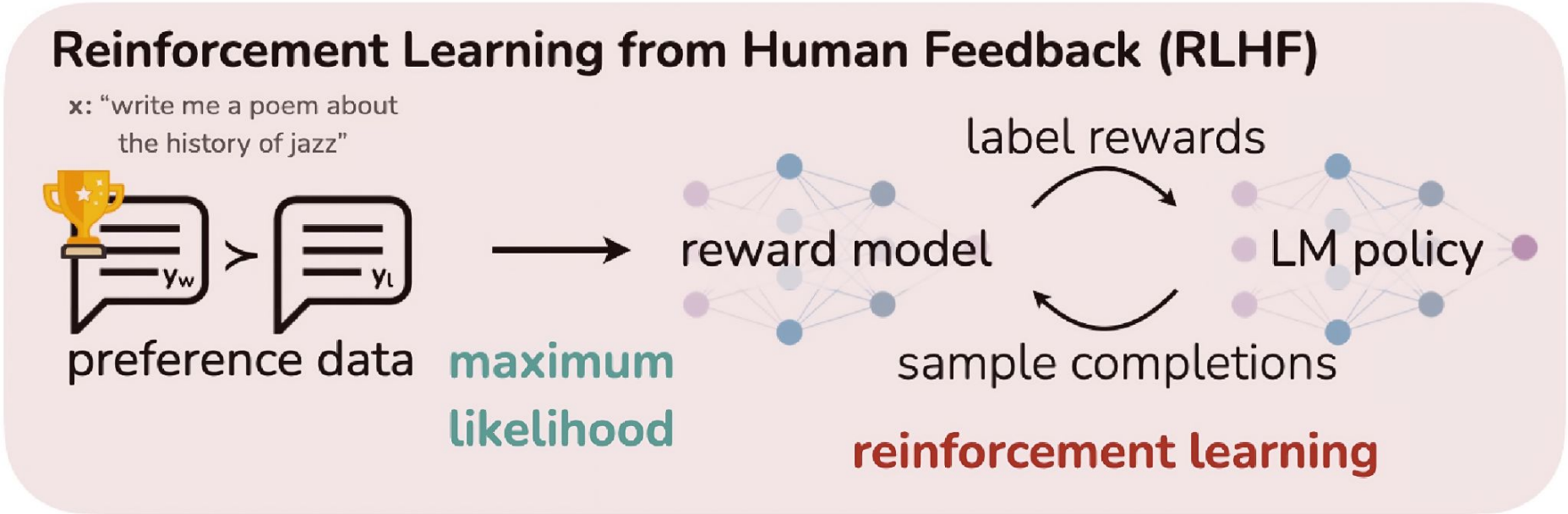
Daniel M. Ziegler, undefined., et al, "Fine-Tuning Language Models from Human Preferences," 2020.

Reinforcement Learning from Human Feedback



Daniel M. Ziegler, undefined., et al, "Fine-Tuning Language Models from Human Preferences," 2020.

Reinforcement Learning from Human Feedback



Rafael Rafailov, undefined., et al, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2024.

Reinforcement Learning from Human Feedback

Inconvénients:

- Apprentissage par descente de gradient : *SFE trick* pour dériver une estimation de Monte-Carlo du gradient => **apprentissage instable**.
- **Budget de calcul**, plusieurs copies du modèle.
  - => jusqu'à 4 copies !
- **Online** v.s. **offline**.

Objectif similaire aux problèmes d'apprentissage par renforcement :

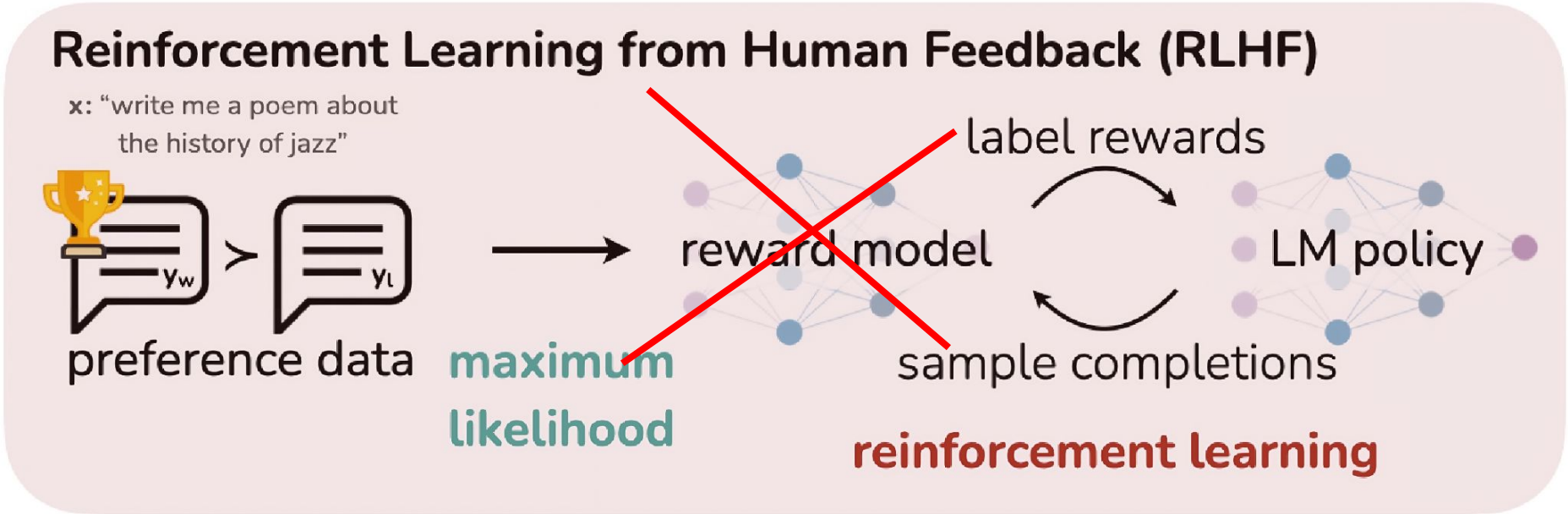
$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)} \underbrace{\left[r_\phi(x,y)\right]}_{\text{Maximisation du score de "reward"}} - \underbrace{\beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]}_{\text{Minimisation de la distance au modèle de base}}$$

Approximation de Monte-Carlo des espérances:

- Le gradient par rapport aux paramètres du LLM est nul :(
- Le gradient en utilisant le SFE trick est non nul et non biaisé ! :)

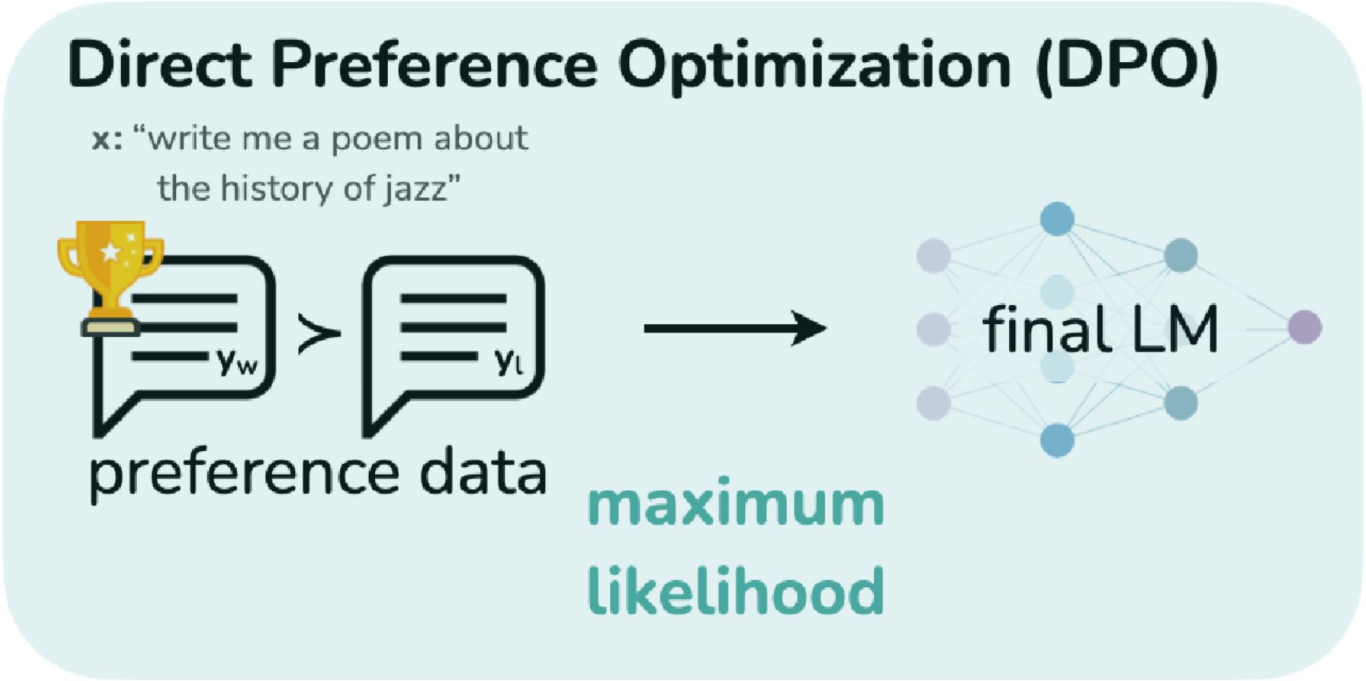
Daniel M. Ziegler, undefined., et al, "Fine-Tuning Language Models from Human Preferences," 2020.

Alternative : Direct Preference Optimization



Rafael Rafailov, undefined., et al, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2024.

Alternative : Direct Preference Optimization



Rafael Rafailov, undefined., et al, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2024.

Alternative : Direct Preference Optimization

- Récompense implicite, entraînement du modèle de langue directement sur les données de préférences.
- Sans devoir passer par de l’apprentissage par renforcement.

<u>Inconvénients:</u>

- Overfitting des préférences.
- Oubli catastrophique (Catastrophic forgetting).

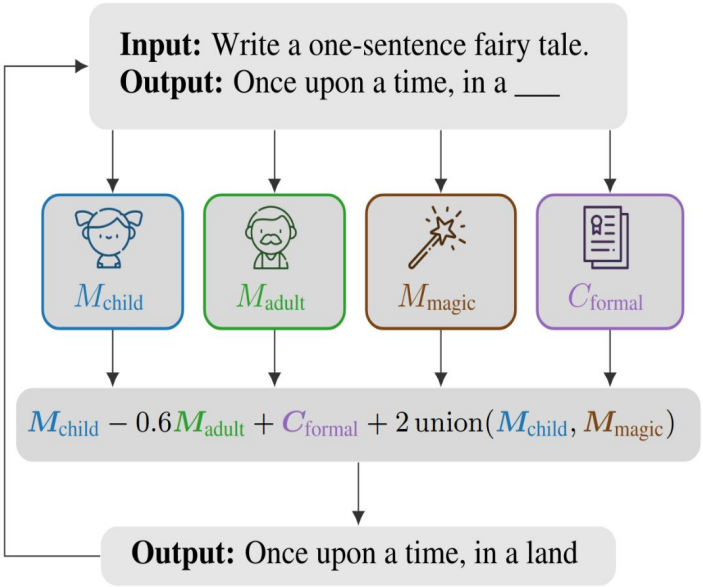
Rafael Rafailov, undefined., et al, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2024.

Alternative : Génération contrôlée

E.g. d'approches :
- Arithmétique des modèles
- Reward Augmented Decoding

Coût élevé de génération:

Génération depuis plusieurs modèles au lieu d'un seul.



Jasper Dekoninck, undefined., et al, "Controlled Text Generation via Language Model Arithmetic," 2024.

Haikang Deng and Colin Raffel. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In EMNLP, 2023. doi: 10.18653/v1/2023.emnlp-main.721.

Plan de travail

Nos pistes de recherche:

1. Une ré-interprétation de l'objectif de RLHF qui permet de dériver une alternative **simple à optimiser**.
2. Un framework théorique solide pour DPO, qui permet de généraliser l'approche à des fonctions de perte qui ne conduisent pas à du catastrophic forgetting.
3. Une approche online guidé par l'incertitude du modèle pour minimiser les coûts d'annotation.

En ce moment:

- Ré-implémentation et expérimentation des méthodes existants
- Mise en place de la piste (1)