Novembre 2023

Eric Debeir - Directeur scientifique de Datalchemy - eric@datalchemy.net

Nos clients nous interrogent de plus en plus sur l'implémentation de modèles de langage pour des chatbots ou des moteurs de recherche de contenu générant, à partir d'une question des documents de synthèses multilingues à partir de leur propre base documentaire. Autrement dit "je veux mon propre Chat GPT". Alors évidemment, nous suivons de près ce qui s'écrit et ce qui peut s'implémenter dans les nouveaux développements autour des LLM. Faisons le tri dans une littérature toujours aussi foisonnante.

Nous observons depuis bientôt un an un phénomène que nous connaissons (hélas) un peu trop bien à Datalchemy, cette fois-ci sur les célèbres *Large Language Models* : une frénésie de travaux scientifiques qui, tous, promettent une petite révolution, mais qui en majorité exigent un recul indispensable quand on veut considérer leurs résultats. En effet (et c'était par ailleurs le sujet de notre dernier webinar), le surplus de recherche crée un effet d'aveuglement, particulièrement dans le domaine du *Deep Learning* où nous évoluons dans un déficit d'approches théoriques assez préjudiciable.

Pourtant, nous ne pouvons pas pour autant ignorer cette recherche. Les *Large Language Models* sont aujourd'hui un sujet brûlant qui touche une majorité de nos clients. Dans la jungle des publications qui sortent, certaines sont plus intéressantes que d'autres, et la reproduction des expériences d'une recherche à l'autre nous permet d'identifier de mieux en mieux les axes de travail pertinents comme les impasses officielles.

Nous vous proposons aujourd'hui de faire un tour d'horizon de certaines publications qui ont été dignes d'intérêt ces derniers mois. Conforme à notre volonté d'appréhender au mieux la robustesse et l'industrialisation de ces outils, nous nous concentrons sur les travaux visant une forme d'interprétabilité des modèles, mais pas uniquement. Au programme, nous avons un travail très intéressant d'optimisation des modèles, deux travaux d'interprétabilité se distinguant du reste, et deux

travaux très réalistes et terre à terre qui recadrent sérieusement les talents incroyables que l'on prête un peu trop vite à ces nouveaux outils.

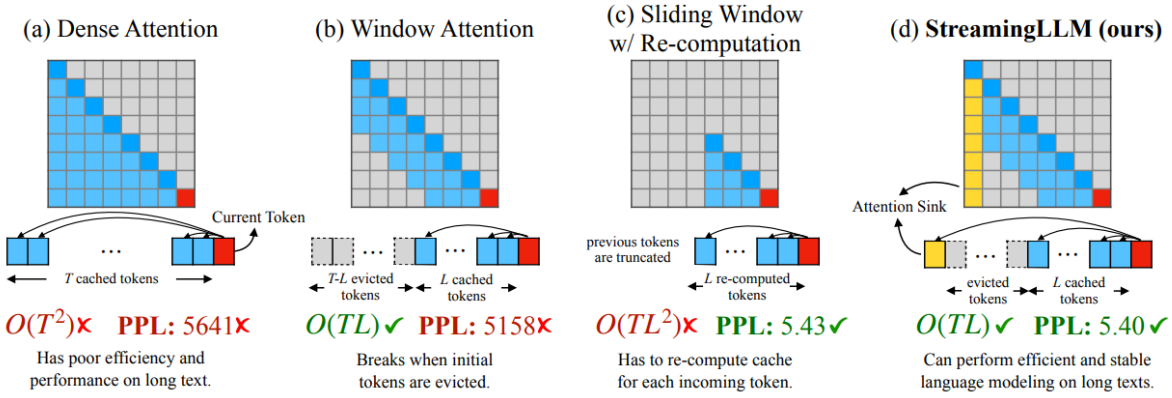
1. Optimiser l'attention - le cauchemar originel de la longueur du contexte

Pourquoi lire ce qui suit : cette complexité croissante avec la longueur est une épée de Damoclès datant de 2017. Tout travail essayant de repousser cette frontière est donc particulièrement intéressant. Ici, la solution a l'élégance d'être assez simple à utiliser, tout en offrant un nouveau regard sur les approches plus anciennes. Ceci dit, gardons toujours du recul, un travail trop récent étant particulièrement risqué à implémenter.

Ce sujet est très connu des chercheurs en NLP depuis 2017 et la sortie du célèbre *Transformer* de *Vaswani et al* : les *Transformers* sont très sensibles à la longueur de la phrase sur laquelle ils travaillent, avec une complexité quadratique rédhibitoire. Et force est de reconnaître que le sujet reste d'actualité, car aujourd'hui encore, ce sujet est considéré comme non résolu par la recherche.

Au-delà d'une complexité technique, cette limitation a des impacts très forts sur l'utilisation d'un *LLM*. Typiquement, exposer au modèle un texte dont la longueur est plus longue que la longueur d'entraînement causera une très forte dégradation des performances. Ainsi, l'application classique du *chatbot* capable de soutenir une conversation dans la durée devient un sujet insoluble, tout du moins via une approche classique.

Dans *Efficient Streaming Language Models with Attention Sinks*, de *Xiao et al* [https://arxiv.org/abs/2309.17453], les auteurs adressent ce sujet d'une manière particulièrement intéressante, en visant un paradigme de *streaming* soit de génération continue dans le temps. Nous n'allons pas trop tomber dans la technique dans cet article (si vous avez des questions, contactez-nous :)), mais un premier résultat très intéressant est qu'une solution anciennement considérée comme pertinente s'avère en réalité être très décevante. Nous parlons ici de la *window attention*, où le modèle ne s'intéresse qu'à un contexte immédiat du *token* sur lequel il est en train de travailler. Si cette approche a souvent été présentée comme une solution, elle est ici invalidée en termes de résultats, et c'est typiquement ce type de retours qui nous intéresse particulièrement pour mieux répondre à nos clients.



(Ci-dessus : représentation en génération de phrases de l'attention donnée aux éléments précédents pour chaque nouvel élément généré, où, dans le mode dense, chaque nouvel élément généré exploite l'intégralité des éléments précédents)

Et en effet, si l'approche de *window attention* est effectivement stable en mémoire, les auteurs observent formellement que les résultats de cette approche s'écroulent dès que la taille de contexte initiale est perdue. Au-delà, ils produisent une observation passionnante sur le concept d'*attention sink*, où les modèles continuent à donner une importance démesurée aux premiers éléments de la séquence même quand on en est assez éloignés. Ce phénomène est identifié comme un problème qui

va nuire fortement à la capacité du modèle à généraliser sur des séquences longues, et les auteurs proposent une solution, dite *StreamingLLM*, qui semble améliorer fortement les résultats. L'approche est relativement simple à mettre en œuvre, avec un code source exploitable et clair.

2. Interprétabilité des LLMs - qu'apprend un LLM ?

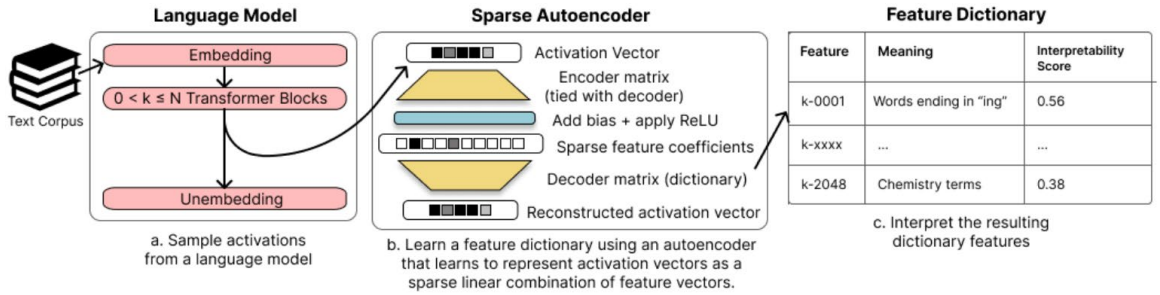
Pourquoi lire ce qui suit : interpréter un modèle de langage est un enjeu fondamental. Les outils issus du *Deep Learning* souffrent terriblement de cet aveuglement sur le fonctionnement interne, avec toujours le risque rampant d'observer une illusion de bon fonctionnement basée sur des biais. Ce type de travail peut s'additionner aux autres approches (par exemple, celles que nous avons données dans notre dernier webinaire sur la robustesse) pour viser un "*best effort*" de contrôle des modèles.

La question hante l'ensemble des chercheurs en *Deep Learning* depuis 2012 : certes, nous pouvons entraîner un modèle sur une tâche et obtenir de bons résultats, mais qu'apprend réellement un réseau de neurones ? Le problème n'a fondamentalement pas changé : il y a trop de paramètres dans un modèle IA pour pouvoir étudier chacun d'une manière correcte et complète. Cela n'empêche pas de travailler (typiquement, les applications de type *neuron coverage* permettent de viser une meilleure robustesse, et l'étude des représentations intermédiaires guide de nombreux travaux), mais cela continue d'être un frein énorme à notre compréhension de ces modèles, leur interprétation et donc, leur utilisation.

Le premier travail qui nous intéresse est le suivant : *Sparse Autoencoders Find Highly Interpretable Features in Language Models*, de *Cunningham et al* [https://arxiv.org/abs/2309.08600]. Ce travail vise à mieux interpréter la représentation interne d'une phrase, d'un mot, voire d'un concept, à l'intérieur d'un *LLM*, et propose une approche pertinente.

Le problème relevé par les auteurs est celui de la poly-sémanticité, soit : un même "neurone" s'activera pour un grand nombre de concepts différents, d'une manière synchronisée avec d'autres "neurones". Dès lors, l'espoir (un peu naïf) de faire correspondre un concept à un neurone disparaît bien vite. Ce phénomène est pourtant logique, nous observons empiriquement qu'un réseau de neurones pourra appréhender un bien plus grand nombre de concepts que la dimensionnalité exposée par ses représentations internes.

La question qui se pose alors est d'identifier quelle combinaison de quels neurones permettrait de représenter tel ou tel concept. Ici, les auteurs utilisent une approche dite de "*sparse dictionary learning"*, autrement dit, la recherche de combinaisons "économiques" (les plus simples possibles) parmi les activations du réseau. L'approche passe par l'entraînement d'un auto-encodeur spécifique, avec une forte contrainte d'économie sur les représentations de la couche centrale :



Le travail donne des résultats intéressants. Typiquement, ci-dessous, les cinq premières combinaisons détectées sur la première couche du modèle étudié (le score d'auto-interprétabilité est un peu plus polémique, issu d'un travail d'*OpenAI*) :

Feature	Description (Generated by GPT-4)	Interpretability Score
1-0000	parts of individual names, especially last names.	0.33
1-0001	actions performed by a subject or object.	-0.11
1-0002	instances of the letter ‘W’ and words beginning with ‘w’.	0.55
1-0003	the number ‘5’ and also records moderate to low activation for personal names and some nouns.	0.57
1-0004	legal terms and court case references.	0.19

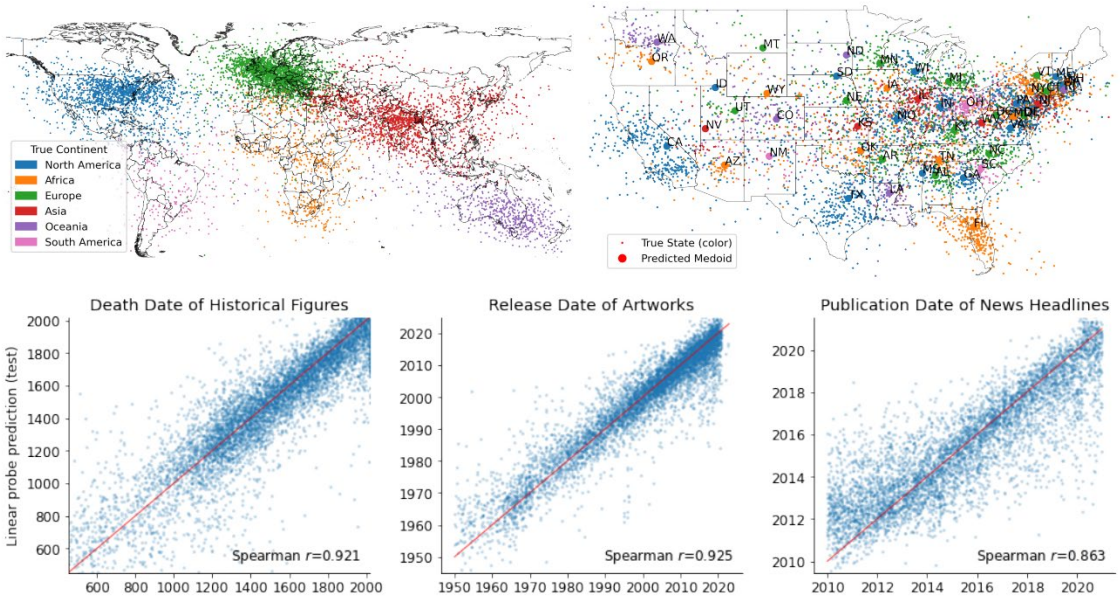
Deux découvertes importantes sont exposées par les auteurs :

- Les représentations apprises par le dictionnaire sont fortement mono-sémantiques, autrement dit, elles isolent correctement un contexte et ne semblent pas s’activer pour d’autres contextes concurrents. Ces représentations semblent donc une clé intéressante pour tenter de décrire les représentations d’un modèle.
- Ces représentations sont directement liées aux prédictions du modèle. Le fait de supprimer arbitrairement l’expression d’une représentation est directement visible dans les prédictions faîtes par le modèle, en accord avec ce qu’est supposé représenter cette expression.

Si ce travail n’est pas le démarrage de l’IA interprétable qui reste encore un fantasme, il s’agit d’une étape importante vers une meilleure compréhension du modèle.
Le second travail est un peu plus polémique, mais a réussi à “faire le buzz”, ce qui d’un point de vue scientifique est rarement une très bonne nouvelle. Mais force est de reconnaître que le titre de la publication fait rêver : *Language models represent space and time*, de *Gurnee et al* [https://arxiv.org/abs/2310.02207]...

Les modèles de langage représentent donc l’espace et le temps ? Doit-on crier aux IAs magiques créant une représentation interne du monde lors d’un apprentissage dédié à la complétion de texte ? Rassurez-vous, comme très souvent, derrière un titre assez racoleur se cache un travail certes intéressant, mais pas non plus révolutionnaire.

Ici, les chercheurs vont tenter de générer une information spatiale (longitude/latitude) ou temporelle (année) à partir de mots spécifiques (noms de ville, œuvres culturelles ou événements historiques, etc.). Pour trouver cette information, ils vont donc entraîner un petit modèle linéaire (de 4098 à 8192 paramètres) pour extraire, à partir de la représentation interne du modèle auquel on soumet ces mots spécifiques, l’information spatiale ou temporelle voulue. Et oui, cela semble fonctionner correctement :



Néanmoins, on peut déjà observer qu'entre le titre et le travail réalisé, se cache un petit ravin. Ce que l'on peut affirmer face à ces travaux, c'est que l'on peut extraire des représentations intermédiaires cette information d'une certaine manière. De là à affirmer que le modèle aurait, d'une manière interne, une "représentation de l'espace et du monde", il ne faut pas non plus s'emballer. Il est fort possible que le modèle ait appris ces informations par coeur d'une manière implicite (comme cela arrive régulièrement), et que le sous-modèle arrive à extraire ces informations via une approche supervisée classique.

Au-delà, les auteurs observent aussi que le fait de rajouter dans la phrase en entrée, outre le lieu ou l'événement digne d'intérêt, des éléments totalement aléatoires, fait drastiquement chuter les résultats. Là encore, nous observons notre limite actuelle de compréhension de ces modèles.

3. Quand Deepmind (et d'autres) observent formellement des limites aux LLMs

Pourquoi lire ce qui suit : Il est indispensable de connaître les limites des modèles de langage avant de démarrer un nouveau projet. Ici, nous observons déjà qu'une pratique d'amélioration des modèles est une impasse, ce qui nous permet de mieux choisir les approches techniques pour résoudre un problème. Ensuite, nous (re)-découvrons que certaines tâches sont encore trop complexes pour ces outils. Mieux connaître les limites des modèles de langage nous permet de nous engager en confiance sur un projet impliquant ces états de l'art de la recherche.

Dans la frénésie de recherche, on a souvent tendance à accepter une théorie comme un fait établi. *Deepmind* vient (avec talent) recadrer les choses dans un de ses derniers travaux : *Large Language Models Cannot Self-Correct Reasoning Yet*, de *Huang et al* [https://arxiv.org/abs/2310.01798]

En effet, face aux nombreux problèmes des LLMs, une approche qui a été particulièrement mise en valeur est celle de l'auto-correction d'un modèle, avec de nombreux travaux académiques observant des améliorations notables des résultats. L'intuition qui se cache derrière est qu'un modèle pourrait améliorer ses propres réponses si on lui demande de chercher par lui-même des erreurs dans sa génération pour les améliorer. Si un optimisme de bon ton flottait autour de cette approche, considérée comme une des meilleures pistes pour améliorer ces modèles, *Deepmind* nous offre ici une leçon magistrale d'épistémologie particulièrement importante.

Dans un premier temps, les chercheurs de *Deepmind* reproduisent les résultats classiques des autres chercheurs. Face à un *dataset* spécifique, on pose des questions au LLM en l'invitant à se corriger, et on observe des conclusions "rassurantes" :

Table 1: Results of GPT-3.5 and GPT-4 on reasoning benchmarks with the setting in Section 3.1.1.

		GSM8K	CommonSenseQA	HotpotQA
GPT-3.5	Standard Prompting	75.9	75.8	26.0
	Self-Correct (Oracle)	84.3	89.7	29.0
GPT-4	Standard Prompting	95.5	82.0	49.0
	Self-Correct (Oracle)	97.5	85.5	59.0

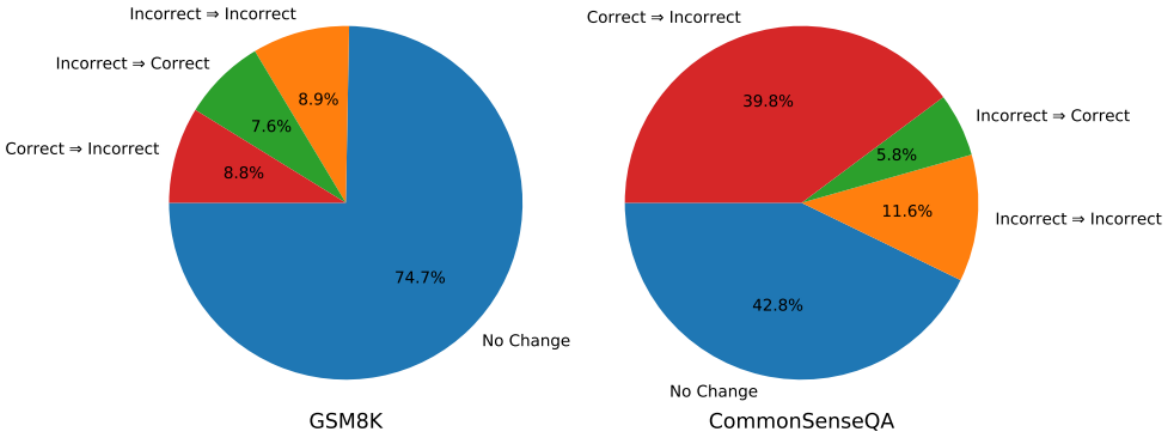
Mais, comme souvent, le diable se cache dans les détails. Et les auteurs mettent le doigt sur un défaut intrinsèque de ces tests académiques : on ne demande au modèle de se corriger que quand la première réponse donnée est fausse. Hors, cette méthodologie ne retranscrit absolument pas le problème qui nous intéresse! En effet, en utilisation normale, si nous parlons d'auto-correction du modèle, nous ne voulons pas d'information externe permettant de diriger le modèle. Dès lors que l'on

supprime cette interférence, les résultats d'auto-correction des modèles montrent, en réalité, une réelle dégradation des performances :

Table 3: Results of GPT-3.5 and GPT-4 on reasoning benchmarks with *intrinsic self-correction*.

		# calls	GSM8K	CommonSenseQA	HotpotQA
GPT-3.5	Standard Prompting	1	**75.9**	**75.8**	**26.0**
	Self-Correct (round 1)	3	75.1	38.1	25.0
	Self-Correct (round 2)	5	74.7	41.8	25.0
GPT-4	Standard Prompting	1	**95.5**	**82.0**	**49.0**
	Self-Correct (round 1)	3	91.5	79.5	**49.0**
	Self-Correct (round 2)	5	89.0	80.0	43.0

May refer to Table 6 of Appendix B for results with different feedback prompts for GSM8K.
The results are consistent, and the variance is low across different feedback prompts.

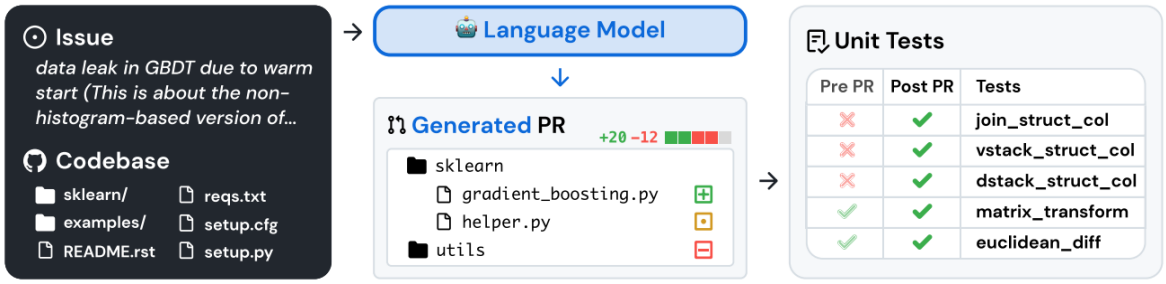


Ce travail est très important, car il montre une fois de plus la tendance qu'a la recherche académique en *Deep Learning* à se jeter sur des solutions en oubliant de critiquer correctement le problème étudié. A la seconde où on se projette vers une utilisation des modèles, ces oublis deviennent des problèmes incontournables. Ici, *Deepmind* invalide cette approche. Les auteurs néanmoins reconnaissent que cette auto-correction peut améliorer les choses dans des cas très particuliers. Ils identifient surtout l'intérêt de valider les sorties d'un modèle en interrogeant d'autres modèles pour confirmer la stabilité de la réponse. Définissant cette approche comme la "*self-consistency*", ils voient là un bien meilleur moyen de robustifier un modèle (certes beaucoup plus coûteux), peu éloigné des *expert mixtures* connues du *Machine Learning* depuis des décennies.

Dernier travail digne d'intérêt, *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?* de *Jimenez et al* [https://arxiv.org/abs/2310.06770] a l'intérêt de nous dégriser en mettant à terre un fantasme classique des *LLMs*. Nous parlons ici du "mythe" des intelligences artificielles capables de jouer un rôle de développeur informatique, mythe sur lequel certains acteurs économiques s'appuient pour réduire leurs équipes en développement.

(Avec un peu de mémoire, vous vous souviendrez d'une revue de la recherche où nous argumentions sur ce que nous pensions, plutôt en mal, concernant ces affirmations)

Ici, les auteurs observent que les *benchmarks* souvent utilisés sont très simples, voire trop simplistes (comme *HumanEval*), avec des problèmes simples pouvant être résolus en quelques lignes de code. Et ils proposent donc un nouveau *benchmark* bien plus réaliste, basé sur de vrais *repositories Github* :



Les résultats sont peu surprenants pour ceux qui gardent du recul face à ce phénomène, et évidemment très décevants :

	BM25 Retrieval		"Oracle" Retrieval	
Model	% Resolved	% Apply	% Resolved	% Apply
ChatGPT-3.5	0.20	10.50	0.52	12.38
Claude 2	**1.96**	29.86	**4.80**	47.00
GPT-4*	0.00	4.50	1.74	13.20
SWE-Llama 7b	0.70	37.84	3.00	**54.80**
SWE-Llama 13b	0.70	**39.41**	4.00	52.10

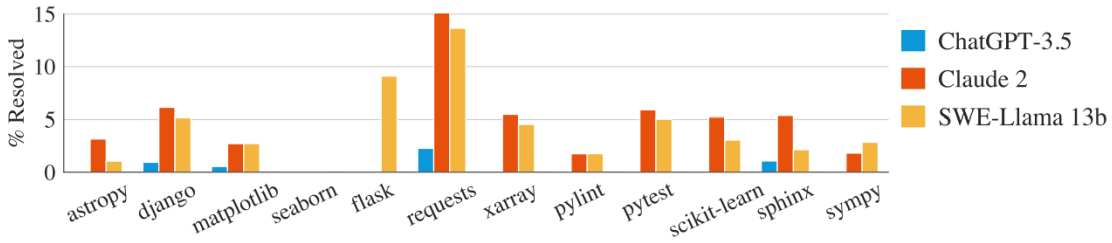


Figure 4: Resolution rate for three models across the 12 repositories represented in SWE-bench.

On rappelle qu'une question fondamentale dans les affirmations sur *GPT 4* est de savoir à quel point le *training set* (inconnu) contient déjà les problèmes sur lesquels nous testons les modèles. Ici, nous observons que sur des problématiques plus complexes, même GPT 4 est très décevant (moins de 2% de résolution). Codeurs de tous les pays, votre emploi n'est aujourd'hui pas en danger 🙄

(Nous ne promettons rien pour demain, car malgré l'esprit critique, force est de reconnaître que les choses vont très vite).