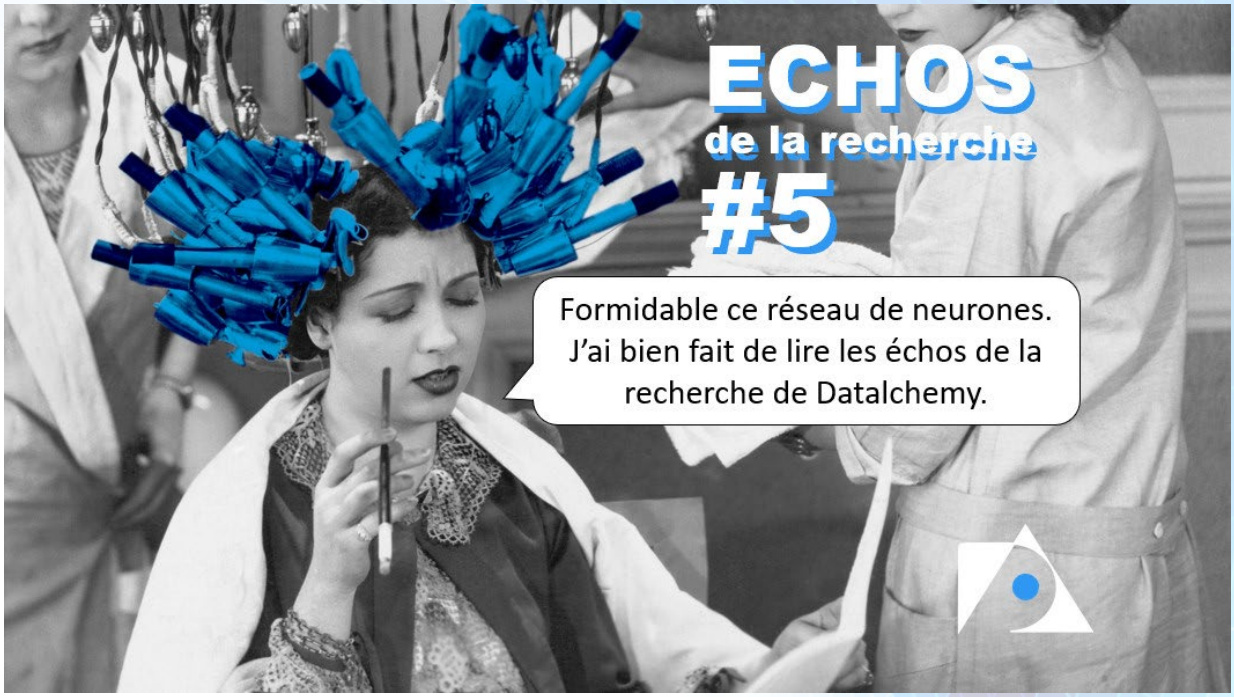


Juin 2023

IA, quel risque à évaluer les risques ?

Les modèles de langage sont toujours l'événement le plus brûlant de l'actualité du *Deep Learning*, et sans surprise, deux des travaux retenus pour aujourd'hui prolongent ces outils. Néanmoins, nous allons un peu nous écarter des multiples travaux visant à légèrement instrumenter ou optimiser ces modèles, pour nous intéresser à deux approches plus originales et plus intéressantes. La première porte sur des travaux de *Google* visant à étendre ces modèles à la compréhension et la manipulation de schémas, quand la seconde elle vise à créer un agent compétent au jeu *Minecraft*, avec (*spoiler*) une petite déception sur l'approche. Néanmoins, avant de partir sur ces travaux, nous allons nous arrêter sur une publication de *Deepmind* importante portant sur la gestion des risques en *IA*, et leur mitigation.

Un système d'alerte pour les risques de l'IA

Une des plus inquiétantes épée de Damoclès au-dessus des outils d'intelligence artificielle porte sur leur robustesse et leur sécurité. En effet, dès qu'on a le malheur de s'éloigner de l'aspect "impressionnant", on re-découvre que ces outils sont aujourd'hui impossible à valider complètement tel qu'on pourrait le faire pour un algorithme plus classique que l'on peut tester exhaustivement. Hors, si entraîner un réseau de neurones est toujours intéressant, nous sommes beaucoup plus attentifs à leur industrialisation et, donc, à la maîtrise de leurs risques.

Le travail récent de *Deepmind* [https://www.deepmind.com/blog/an-early-warning-system-for-novel-ai-risks] est ainsi incontournable pour tout acteur désirant questionner sa stratégie en test et validation de modèles IA. S'il ne donne évidemment pas de solution "parfaite", il a le mérite de rappeler l'ensemble des risques et de proposer une méthodologie itérative pour maîtriser au mieux les outils générés. *Deepmind* s'intéresse ici aux risques dits "extremes", notamment en distinguant les risques liés à une mauvaise généralisation et ceux dus à une mauvaise manipulation.

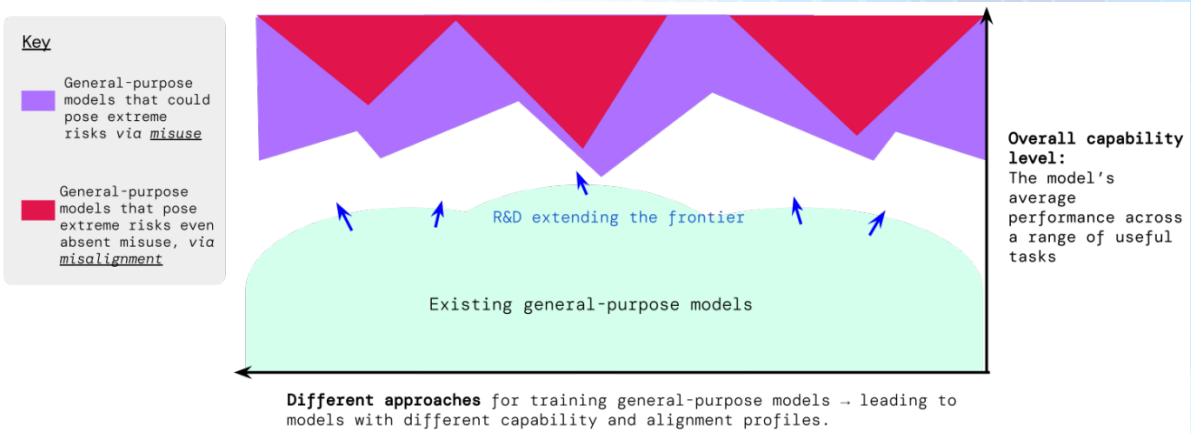
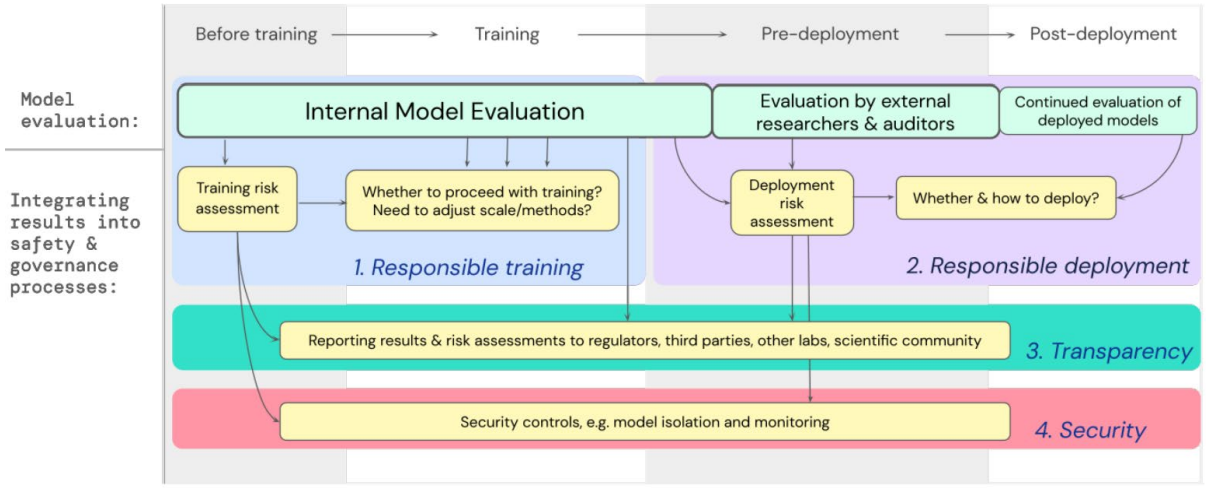
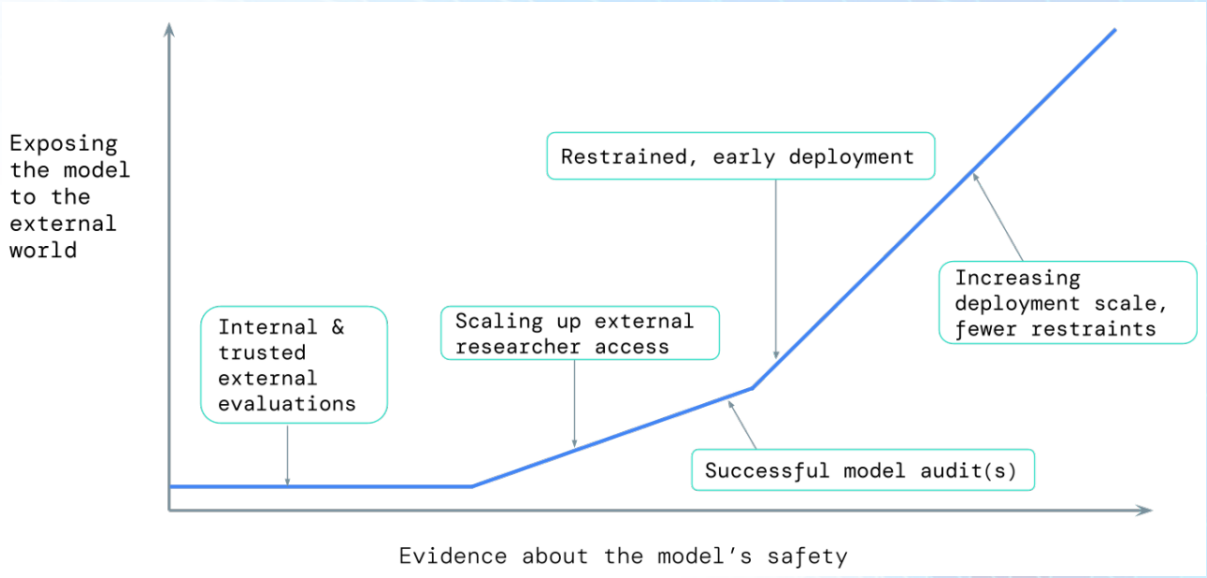


Figure 2 | Leading AI developers push the frontier outward, typically by training models at greater scale and using more efficient architectures and algorithms. This continued expansion takes the field closer to points in model space that could pose extreme risks. The diagram is purely illustrative.

Quels sont ces risques ? *Deepmind* tente d'être le plus exhaustif possible, et propose une méthodologie adressant un entraînement responsable, un déploiement impliquant des auditeurs externes, une transparence sur les conditions de l'entraînement, et une politique de sécurité complète, notamment via une isolation du modèle mitigeant attaques adversariales ou autres détournements



L'idée sous-jacente porte sur une exposition croissante d'un modèle auprès du grand public au fur et à mesure que le modèle est validé, tout en conservant une maîtrise des réentraînements afin de conserver la possibilité de contrôler les modèles générés.

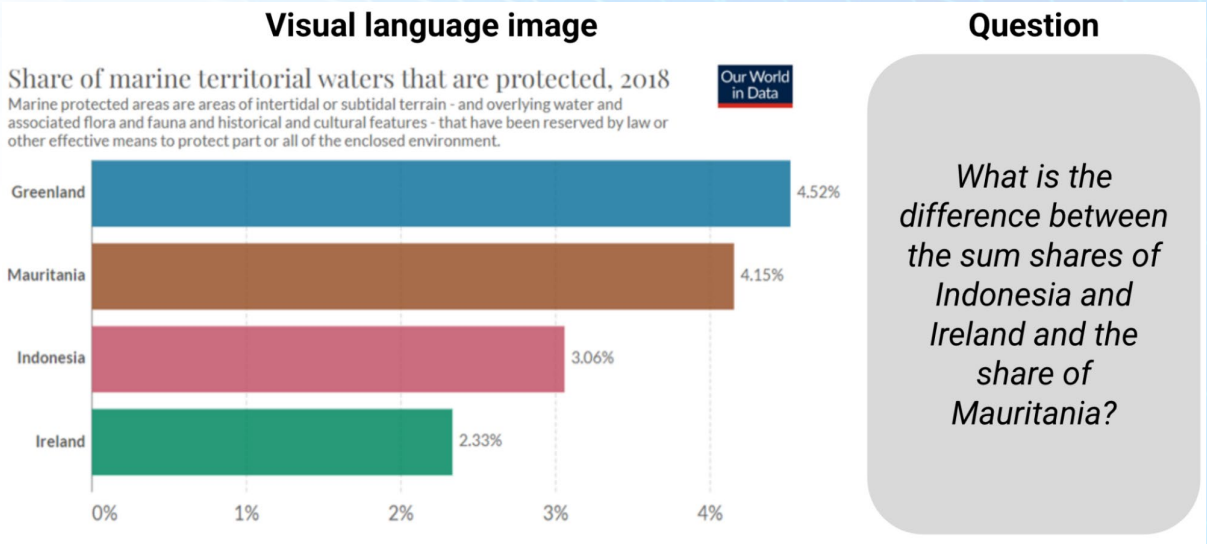


Deepmind identifie enfin six limitations fondamentales à toute approche de ce type, dont la majorité se rapportent à nos déficits de compréhension en *Deep Learning*. :

- La connexion d'un modèle avec des outils externes, notamment d'autres modèles, causant ainsi des comportements non contrôlés. Nous observons déjà de nombreux scénarios de couplage fort entre différents modèles.
- Les menaces encore non identifiées sur les modèles.
- Les propriétés fondamentales encore non maîtrisés et ne pouvant être observées par une évaluation
- L'émergence de comportements liés à la taille des modèles, pour des tailles encore trop nouvelles.
- La maturité insuffisante des outils d'évaluation que nous avons à notre disposition
- Une confiance trop grande dans les outils d'évaluation disponibles aujourd'hui.

Un modèle de langage apprend à interpréter des schémas

Ce sont deux publications différentes de *Google Brain* qui sont entrées dans notre radar ces dernières semaines, qui visent toutes deux à travailler sur l'interprétation d'un schéma numérique. L'enjeu est ici évident : de nombreuses connaissances sont exposées sous la forme de schémas visant à résumer un grand nombre d'informations numériques en un unique visuel. Si des modèles deviennent capable de modéliser ce type d'information, nous gagnons un nouveau champs d'analyse particulièrement puissant pour interpréter des documentations spécifiques. Un exemple d'application, repris depuis le blog de *Google Brain*, est visible ci-dessous :

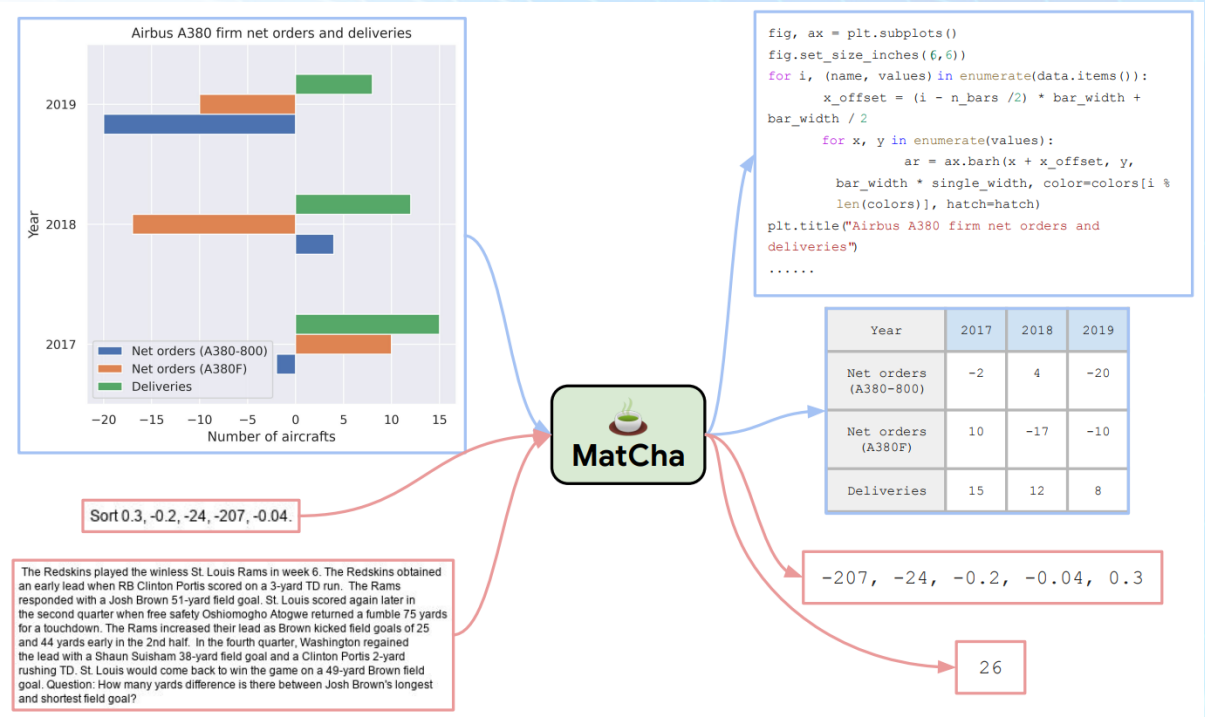


La tâche étant tout sauf triviale, deux travaux différents sont ici mis à l'honneur

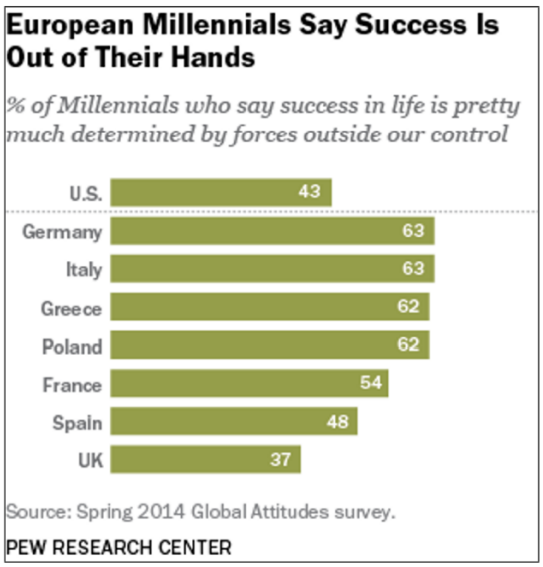
MatCha: Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering, *Liu et al*, [https://arxiv.org/abs/2212.09662], vise à améliorer les travaux précédents en modéle de langage visuel. Le modèle nommé **MatCha** va ainsi viser une modélisation jointe entre les pixels d'un schéma et un langage décrivant ce schéma, via deux formes de pré-entraînements :

- Le premier, *Chart Derendering*, vise à inverser la génération classique d'un schéma, en partant d'un schéma pour générer le code *Python* ayant généré ce schéma, cette tâche forçant une modélisation des valeurs et de la forme du schéma. Le modèle va aussi apprendre à générer une table de valeurs à partir d'un tel schéma, pour retrouver une structuration minimale de la donnée sous-jacente
- Le deuxième pré-entraînement vise sur le "raisonnement mathématique" (guillemets indispensables), pour entraîner le modèle à appliquer des opérations simples comme le tri, la génération de valeurs extremums ou de moyennes, etc. Le dataset **MATH** est ici utilisé.

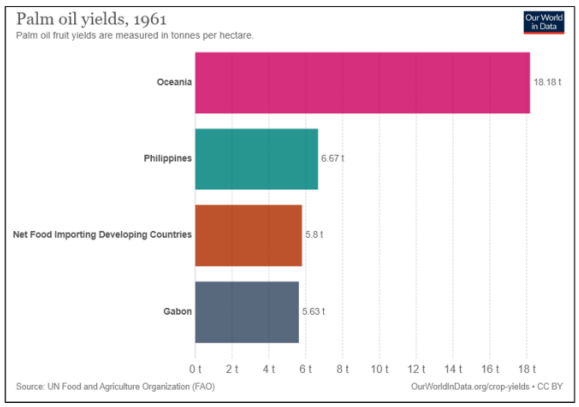
Ces deux tâches de *pretraining* sont visibles ci-dessous, en bleu pour la première tâche, en rouge pour la seconde :



Les résultats sont déjà intéressants, et remercions les auteurs de présenter autant des cas réussis que des cas d'échec :



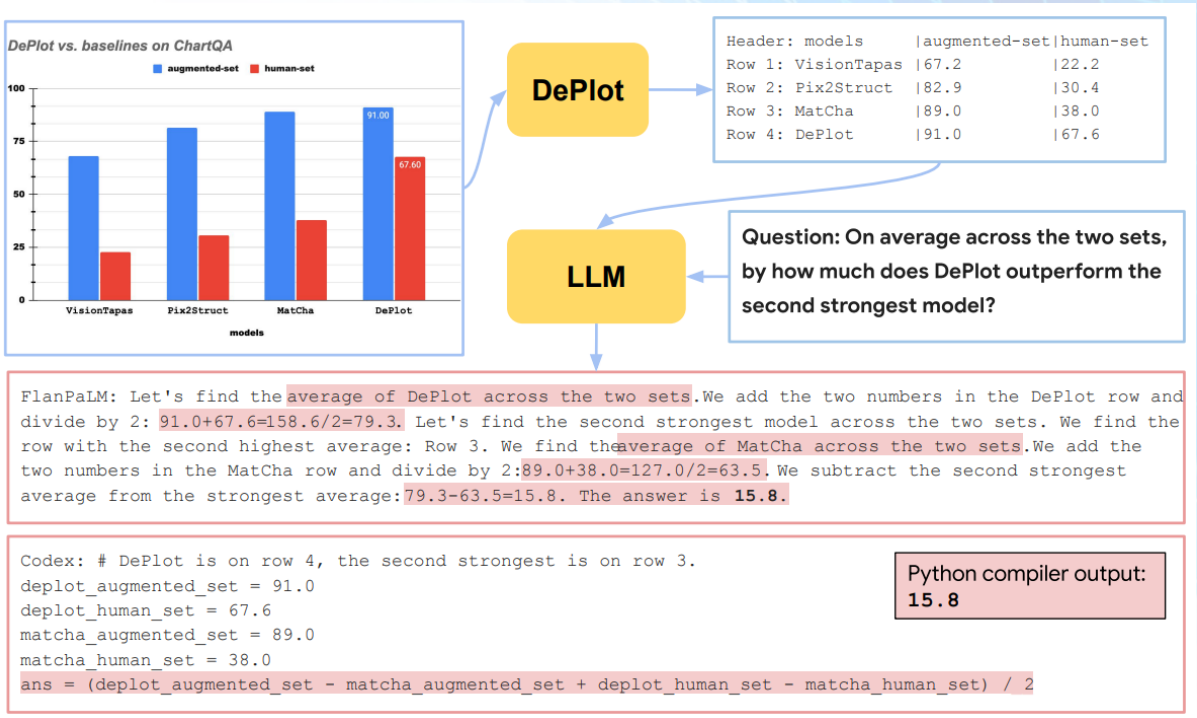
What is the average of last 4 countries' data?
PaLI: **40.94** Pix2Struct: **40.5** MATCHA: **50.5**



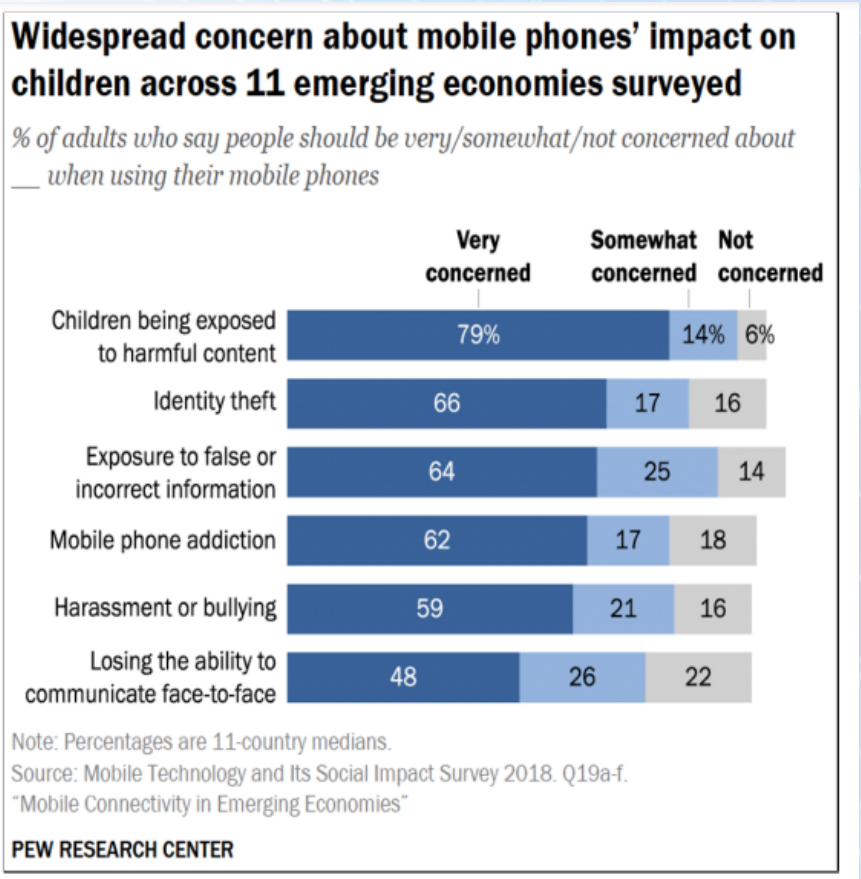
Is the sum of all last three places more than Oceania?
PaLI: **Yes** Pix2Struct: **Yes** MATCHA: **Yes**

Le second travail est ***DePlot****: One-shot visual language reasoning by plot-to-table translation*, de *Lu et al* [https://arxiv.org/abs/2212.10505]. Basé directement sur le premier, il vise lui à adresser le sujet de la compréhension visuelle de diagrammes, dans un premier temps sur des transcriptions de schéma vers texte, pour ensuite tester des questions ouvertes sur le texte généré.
Le modèle va toujours tenter de générer, à partir du schéma, une table linéarisée de données. Cette approche est intéressante car elle vise une uniformisation de la forme de la

donnée générée qui permet notamment de contrôler le fonctionnement et de se projeter facilement vers un outil d'application. Elle pose aussi des limites évidentes sur le type de schéma qui peut être adressé par ce modèle.
Le modèle peut ensuite être couplé à un modèle de langage classique (quel qu'il soit) pour effectuer des tâches d'analyse :



Les auteurs reproduisent les tests faits sur le premier modèle, pour ensuite observer de bien meilleurs résultats en utilisant *DePlot* avec un modèle de langage.



Question: Is the average of all the bars in "identity theft" greater than the highest value of the gray bar?

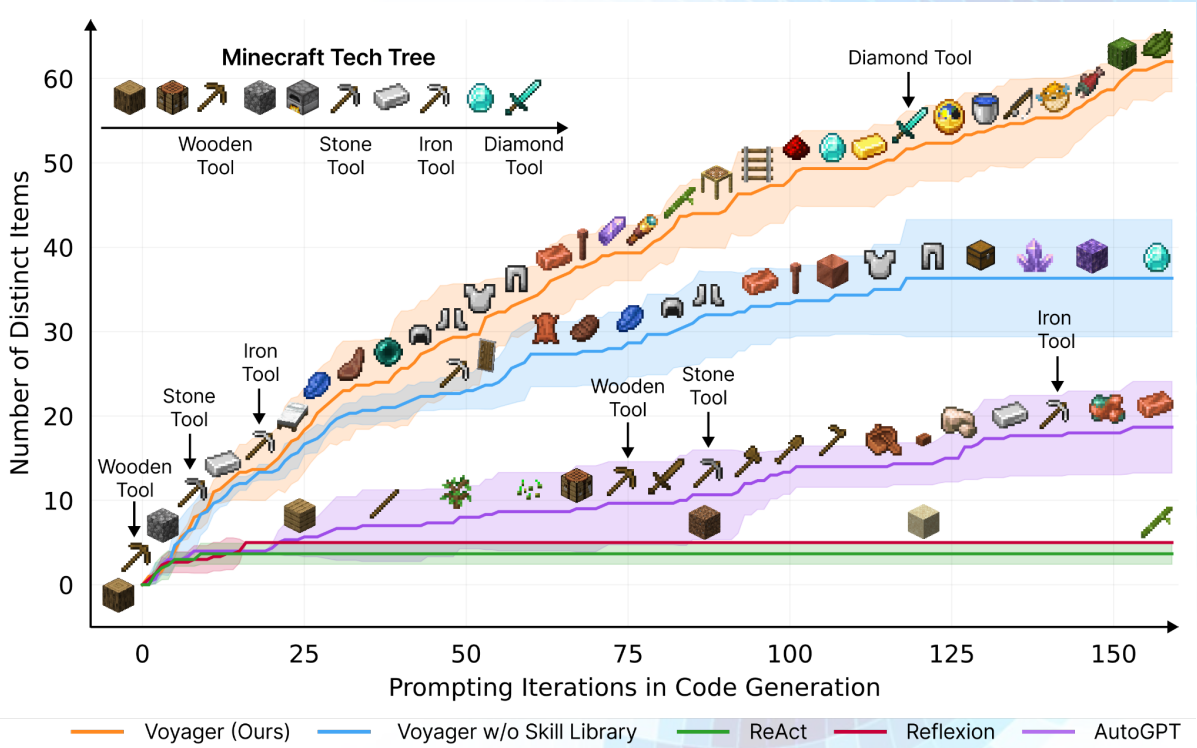
MATCHA: **No**

DEPLOT+FlanPaLM CoT: Let's find the average of all the bars in "Identity theft". The sum of all the bars is 66+17+16 = 99. The average is 99/3 = 33. The highest value of the gray bar is 26. 33 is greater than 26. The answer is **Yes**.

Pourquoi c'est intéressant : La gestion de documents est un sujet central chez *DatAlchemy*. Nous avons travaillé de nombreuses fois dans des configurations où nous devons adresser une masse de documents pour extraire de l'information. Le fait de pouvoir interpréter des schémas et générer une information qualifiée est un énorme avantage pour régulariser les informations extraites et compléter ce qui est extrait du texte pur. Un nouveau jouet pertinent dans notre boite à outils 🙂

Une IA qui sait jouer à Minecraft (ou pas)

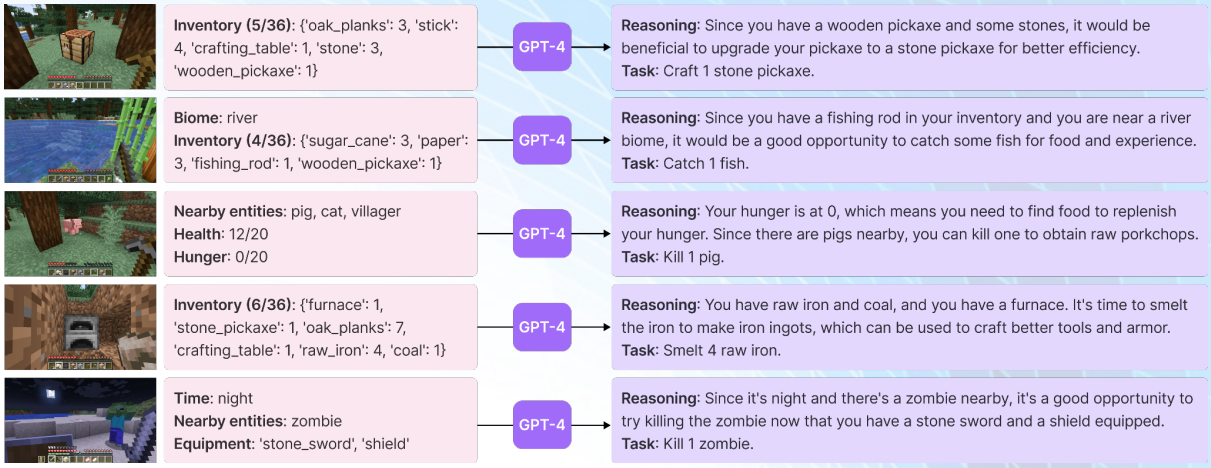
Beaucoup de bruit récemment sur le travail *Voyager: An Open-Ended Embodied Agent with Large Language Models*, *Wang et al* [https://voyager.minedojo.org/], dans lequel on retrouve des acteurs prestigieux comme *NVIDIA* ou *Standford*. Cette approche vise à créer incrémentalement un agent jouant au jeu *Minecraft* en s'appuyant sur les modèles de langage, et obtient des résultats impressionnants, où un agent va progressivement réussir à construire des éléments de plus en plus complexes :



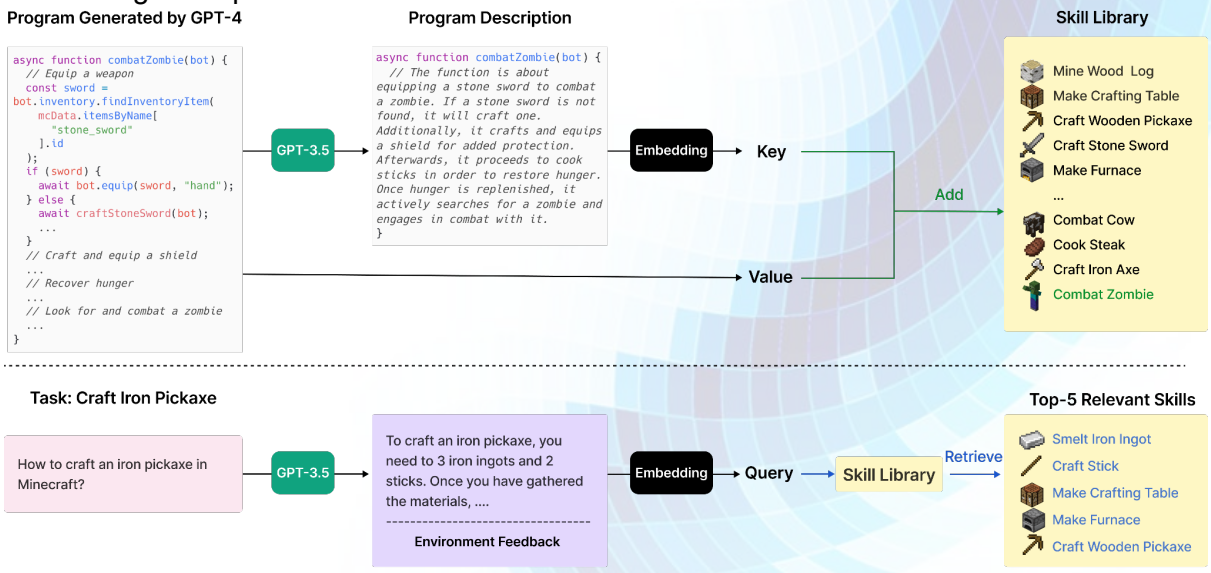
Du point de vue du renforcement, un domaine que nous suivons avec beaucoup d'attention, le résultat est impressionnant. Néanmoins, en allant analyser l'approche, nous allons observer des choses très intéressantes, mais d'autres plus décevantes.

Le point très intéressant porte sur l'utilisation d'un modèle de langage selon une approche incrémentale :

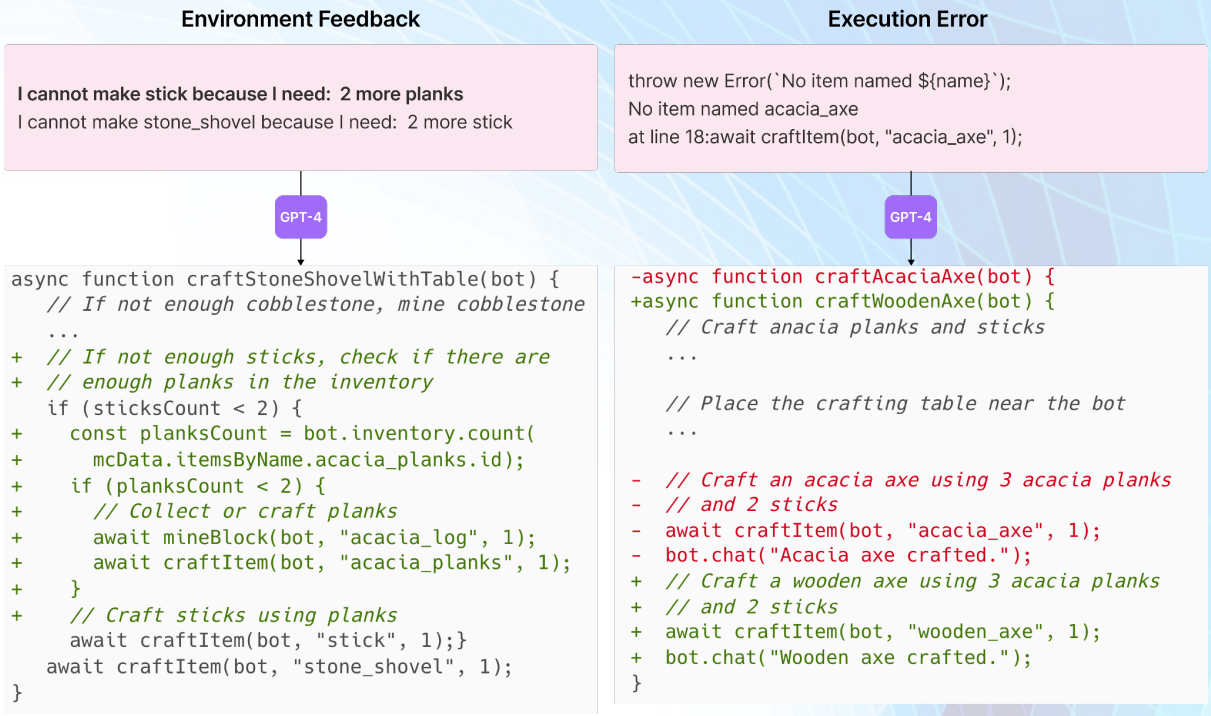
L'approche dite d'*automatic curriculum* cherche à développer une exploration incrémentale basée sur *GPT4*. Le modèle de langage, ici, va convertir une description formatée de la situation observée pour extraire une nouvelle tâche à partir d'une situation observée :



La gestion des compétences vise à encapsuler la description d'une action sous la forme d'un *embedding* (issu de GPT 3.5) plus simple, prolongeant les approches classiques de projection dans un espace latent de l'information. Une compétence est modélisée sous la forme d'un code source d'interaction avec *Minecraft* (nous en reparlons très vite) qui sera lui-même généré par *GPT4*.



Enfin, le mécanisme permettant de valider un code source de compétence issu de *GPT4* en adressant d'une manière pertinente la possibilité d'observer des erreurs dans une prédiction de ce modèle. Les retours de *GPT4* vont être progressivement testés pour observer des erreurs d'exécution, et chaque erreur va conduire à une nouvelle interrogation de *GPT4* jusqu'à obtenir un code qui s'exécute correctement.



Pourquoi c'est intéressant mais un peu décevant

Le point fondamental de ce travail est une instrumentation de *GPT4* à différents niveaux pour générer de nouveaux objectifs pour l'agent, ainsi que de nouvelles interactions. Considérant les défauts de *GPT4*, et notamment sa capacité à halluciner des résultats faux, le mécanisme d'interaction est particulièrement intéressant en ceci qu'il nous donne un moyen d'améliorer le retour d'un modèle de langage d'une manière itérative. Nous commençons à observer de nouvelles méthodologies d'utilisation de ces modèles de langage vers plus de robustesse, une bonne nouvelle. Néanmoins, le test effectué ici (production d'un code source dont on contrôle la bonne exécution) n'est pas nécessairement extensible à d'autres problèmes.

La déception est ici que nous ne sommes pas réellement dans une approche de renforcement, où un agent apprend à modéliser le problème et à apprendre une *policy* de réaction à l'environnement. Si c'était le cas, la prouesse serait remarquable, vu la complexité de *Minecraft*. Ici, le point le plus intéressant, celui de la connaissance sur le jeu, est totalement dédiée à *GPT4*. Hors (le lecteur intéressé se reportera à nos derniers articles), il existe encore un énorme doute sur le *dataset* d'entraînement de *GPT4* et donc la qualification de ses résultats, entre vraie généralisation ou une "simple" application de la connaissance adressée. Présentée autrement, la question est de savoir si cette approche pourrait s'appliquer à un nouveau problème moins présent dans le *dataset* d'*OpenAI*.