

Rapport Final - Yan Rabefaniraka

Yan Rabefaniraka, stagiaire de Meriem Ouederni et David Brunet

31 Juillet 2025

Contents

Sujet de stage	1
Intérêt de la mission en tant qu'étudiant avec une professeure pour tutrice	2
Ressenti - étudiant	2
Utilisation de l'IA par le corps apprenant	2
Apprentissage	2
Facilitation des tâches pénibles	3
Triche	3
Objectif: Co-cognition	3
Pistes d'amélioration vers la co-cognition	3
Tests menés sur des cas d'utilisation en code	4
Jeu des allumettes - Cas grand public	4
Écriture d'un sujet	4
Résultats des tests	4
Conclusion - Jeu des allumettes	4
Ingénierie Dirigée par les Modèles - Filière Réseau de l'N7	4
Zones de test	4
Résultats et conclusion - IDM	4
Préparation d'un notebook pour une possible formation	4
Comparaison des modèles selon les cas d'utilisation	4
Conclusions et pistes d'exploration	4
Remarques :	
• Fichier engendré avec:	
pandoc --toc -o ../pdfs/report1_yan_rabefaniraka.pdf report1_yan_rabefaniraka.md	
• Par facilité d'écriture, on utilisera le masculin 'étudiant' pour qualifier les étudiant.es.	

Sujet de stage

Pour ce stage, il m'est demandé en tant qu'étudiant en sciences du numérique d'établir une **étude de cas approfondie** sur l'utilisation actuelle par les étudiants de l'IA générative dans ce domaine— en particulier des modèles de langage— et d'en déduire une possible utilisation plus efficace pour l'apprentissage. Mon expérience personnelle et mon statut d'étudiant sont un bon apport de départ, accompagnés par ma tutrice qui enseigne directement aux concerné.es.

Cette mission n'a **pas de filière cible** et n'est pas seulement dirigée vers celles spécialisées en informatique, tout d'abord parce que l'aide que l'IA pourrait fournir dans les études s'applique à tous les domaines, mais aussi parce

que l'importance grandissante des petits cas d'usage de l'informatique dans les études supérieures (e.g savoir mettre sur un graphe avec *Python/Matlab* ses données d'expérience, ou encore faire des statistiques sur *RStudio*) permet de se questionner sur les facilitations que l'IA offre à celles et ceux dont le futur métier ne se trouve pas dans les sciences du numérique.

Ainsi, les tests effectués et les conclusions déduites de cette étude de cas— spécifiquement pour les sciences du numérique— doivent être applicables pour un **niveau débutant**, voire en particulier pour des langages haut niveau comme le *Python*.

Cependant, on peut aussi vérifier la puissance / les limites de l'utilisation de l'IA Générative dans l'apprentissage d'une matière complexe. Pour cela, l'**Ingénierie Dirigée par les Modèles**, enseignée (par Mme Ouederni aux 2ème années) à l'ENSEEIH est certainement choisie: C'est une matière suffisamment théorique mais avec des applications directes dans les différents processus de développement/industriels, avec une grande part de notions graphiques. Elle s'enseigne aussi avec 7 cours de *Travaux pratiques* sur machine, avec code et modélisation donc. On pourra ainsi vérifier ce que l'IA peut proposer à un étudiant qui cherche à l'utiliser pour travailler la matière en autonomie.

L'objectif final de cette étude de cas étant la création d'une formation / d'un module qui offrirait aux étudiants de tout horizon un **set d'outils** pour leur enseigner les bonnes pratiques, il n'est pas anodin de donner les bases de l'IA génératives à travers une courte vulgarisation du fonctionnement de la GenAI et des LLMs pour les non-initiés. Le public étant large il est aussi intéressant de mentionner les impacts environnementaux en rappelant les coûts en énergie et en eau de l'apprentissage et de la maintenance des services, de la place des utilisateurs dans la boucle (apprentissage actif, données personnelles) mais aussi des limites de l'IA *i.e* les hallucinations, les erreurs, les biais causés par le *data set*, les biais de confirmation de ChatGPT, etc. Mais, il est aussi essentiel de leur montrer à quel point ces outils sont puissants, en parlant de tout ce qui est inclus avec ChatGPT (compilateur LaTeX, Python), la gratuité de Github Copilot pour tout étudiant avec au choix l'utilisation de modèles de Claude Sonnet, GPT, et Gemini, ainsi que les modèles *offline*. Cela inclut donc un comparatif des différents modèles de langage gratuits actuellement sur le marché, par tâche et *cas d'utilisation* afin de comparer leur pertinence.

Enfin, cela implique donc d'établir des **critères de qualité/validation** de réponses selon le domaine, les prompts, l'IA, etc. Pour ce faire, plusieurs tests seront effectués et mis sur tableau tout au long du stage. Des templates seront également préparés pour d'éventuels *Jupyter Notebooks* (à lancer sur *Google Colab* par exemple). Ces derniers facilitent l'interactivité et augmenteront donc l'accessibilité du projet.

Intérêt de la mission en tant qu'étudiant avec une professeure pour tutrice

En tant qu'étudiant en sciences du numérique, je vais donc me servir de mes propres expériences avec l'IA Générative et vais partager l'usage que j'en fais ou que j'ai pu constater lors de ma 1^{ère} année, que ce soit pour y apporter un regard critique ou m'en servir de base pour développer la suite du projet.

Ressenti - étudiant

Dans cette section, je vais donc étayer les différents usages de l'IA Générative que j'ai jusque là pu observer.

Utilisation de l'IA par le corps apprenant

Apprentissage

On constate plusieurs cas d'utilisation courants que font les étudiants de l'IA générative dans l'apprentissage, en particulier de ChatGPT:

- **Trouver des solutions à des problèmes spontanés** qu'ils rencontrent lors de l'apprentissage d'un cours ou la résolution d'exercices, quand aucun professeur n'est à disposition pour y répondre. Cette utilisation est de plus en plus courante quand les étudiants résolvent des exercices pour lesquels il n'y a pas de correction claire et où ils ne sont donc pas assurés d'avoir les bonnes réponses, voire d'avoir compris le cours. Cela peut aussi s'appliquer à la programmation, quand la documentation pour une librairie est difficile à trouver ou à lire et qui incite régulièrement les étudiants à demander l'explication de lignes de code à un Chatbot. On peut par

exemple penser aux différentes fonctions intégrées de *Matlab* dont la compréhension peut parfois être sinieuse. On peut aussi penser au développement *Web* où les étudiants peuvent demander à ChatGPT de leur expliquer les différentes balises *CSS* ou *HTML*.

- **Expliquer/résumer un cours** qu'ils n'ont pas compris ou dont ils veulent vérifier leur propre compréhension. Cela peut s'apparenter à de l'aide personnalisée, avec les étudiants qui fournissent la section du cours qu'ils n'ont pas comprise. De même, certains s'en servent pour générer des questions typiques ou des quiz, comme des équivalents à la plateforme *Quizlet*.

AJOUTER IMAGES ICI pour illustrer avec Graphes python, code Matlab.

Facilitation des tâches pénibles

Une utilisation courante est aussi celle de la **facilitation des tâches fastidieuses** (répétitives, peu challengeantes) en passant par l'IA. On rencontre le plus ce cas d'utilisation lors de de TP's ou de projets qui nécessitent du code répétitif et peu demandant en réflexion technique pour les étudiants. On peut notamment penser aux méthodes répétitives en programmation orientée objet, comme des *getters* ou des *setters*, ou encore aux matières dont le code n'est pas le sujet principal mais un support d'application (algèbre linéaire, automatique ou télécommunications par exemple).

Triche

On observe parfois des cas où les étudiants font faire tout leur projet / TP par l'IA, sans être capable de reproduire ni *a minima* d'expliquer le résultat. Cela a un apport nul pour l'apprentissage et est donc le cas qu'on cherche à éviter. On peut se baser notamment sur la récente étude menée par des chercheurs du MIT qui ont conclu à une baisse des capacités cognitives liées à l'utilisation exclusive de ChatGPT par des étudiants pour écrire leurs dissertations; En effet, il ne faudrait pas que la facilité d'utilisation de l'IA nuise à l'intégrité réflexive des apprenants, en particulier quand ces derniers n'ont pas encore acquis l'expertise dans le domaine concerné.

Objectif: Co-cognition

L'objectif visé serait pour l'étudiant d'atteindre la co-cognition: Un stade où, ni l'IA ni l'étudiant n'auraient été capables de réaliser le code sans l'autre. C'est-à-dire que le code final serait bien meilleur que celui proposé en un *prompt* par l'IA, et que l'étudiant se voit capable d'expliquer chaque étape puisqu'il aura directement contribué à chacune. Cela implique que ce dernier sait exactement ce qu'il veut et qu'il est capable d'évaluer le code que lui propose l'IA, de revenir dessus, éventuellement de l'améliorer ou de le corriger.

Pistes d'amélioration vers la co-cognition

Les principes que j'ai utilisés pour essayer d'approcher la co-cognition (/ co-construction) sont: - **Role prompting** → Définir un rôle et des compétences spécifiques à l'IA (e.g: "Tu es un programmeur fonctionnel"). Cela permet d'ajouter du contexte qui dirige la génération de l'IA vers un résultat plus spécialisé; C'est en particulier efficace avec des IAs généralistes comme **ChatGPT**. Cela s'est révélé moins impactant avec **Github Copilot** qui est *fine-tuned* avec les répertoires publiques **Github**, et qui peut donc très bien faire le travail sans. - **Contexte détaillé** → Fournir les spécifications complètes du projet c'est-à-dire les consignes du projet données par le professeur concernant le code et le raisonnement, mais aussi ce qu'on attend exactement de l'IA, par exemple "expliquer le code/la syntaxe" ou encore "écrire un squelette modulaire du code". - **Collaboration par itération**: Être capable d'interagir avec les réponses de l'IA. Cela implique de travailler de son côté sur ce qu'elle renvoie, comme corriger son code, repérer les erreurs de compréhension (ou hallucinations), les manquements, ou simplement améliorer les prémices qu'elle propose. La co-cognition/co-construction ressemblera inévitablement à une discussion entre deux collègues qui améliorent ou questionnent tour à tour leur proposition. - **Connaissances établies dans le domaine**: Pour que tous les points précédents soient efficaces, cela nécessite de l'utilisateur qu'il soit au moins légèrement éduqué voire expert dans le domaine sur lequel il utilise l'IA. En clair, il a besoin de connaître les bases de la syntaxe du langage de code/modélisation utilisé, les détails exacts du sujet avec les attentes et contraintes, et est capable de questionner les générations de l'IA. Son but est de rebondir sur ces dernières pour produire encore mieux; elles peuvent servir de base de réflexion ou de base de code pour quelque chose qu'il saurait faire correctement/décemment tout seul.

Tests menés sur des cas d'utilisation en code

Pour l'étude de cas approfondie, nous avons pris deux sujets distincts dans leur exigence. Cela nous permettra de nous faire une idée sur la viabilité d'utiliser une IA pour un cas accessible comme pour un cas plus complexe.

Jeu des allumettes - Cas grand public

Le jeu des allumettes est un sujet de projet très court que j'ai eu à réaliser en **Ada** puis en **Java**. La légère complexité vient des contraintes imposées qui sont claires et rigides, et qui permettent d'évaluer la capacité d'un étudiant à implémenter un cahier des charges très explicite dans ses attentes.

Écriture d'un sujet

La première étape a été de réécrire simplement le sujet du jeu des allumettes. Il inclut les règles du jeu et conditions de victoire, ainsi que les contraintes de programmation et exigences. J'y aussi mis des extraits d'affichage suffisamment complet pour qu'il soit reproductible. Cela pourra servir de base pour évaluer les réponses de l'IA et ainsi savoir comment mieux poser les prompts à partir du sujet directement. La qualité des prompts sera aussi auto-évaluée selon la grille du PPAi6, au même titre que la qualité des réponses sera évaluée selon le code obtenu.

Résultats des tests

- **Consommation passive** →
- **Création de contenu** →
- **Tentative de co-crédation** →

Conclusion - Jeu des allumettes

Ingénierie Dirigée par les Modèles - Filière Réseau de l'N7

L'*Ingénierie Dirigée par les Modèles* est une matière relativement complexe enseignée à l'ENSEEIH. L'itération choisie est celle pour la filière **Réseau** de l'école, qui n'est donc pas forcément composée d'étudiants experts en informatique. Ainsi, on peut observer l'utilité de l'IA dans l'apprentissage et l'application de langages de modélisation, pour des étudiants dont le domaine d'expertise visé n'est pas la programmation ou le logiciel. Cette section part directement du postulat qu'on veut la co-crédation et utilise donc les points mentionnés dans la section correspondante.

Zones de test

écrire les langages/points étudiés et comment l'ia a été utilisée.

Résultats et conclusion - IDM

Préparation d'un notebook pour une possible formation

Détailler les avancées sur le notebook pour David

Comparaison des modèles selon les cas d'utilisation

Insérer le tableau réalisé (fais le en mieux qd même)

Conclusions et pistes d'exploration