Aligner les LLMs aux graphes de prérequis pour des explications adaptées

MOKHTARI Dhia Elhak **LAKHMI** Hichem Billal **OUHENIA** Anis

Encadré par **HAMIDI** Massinissa

Les étudiants n'ont pas tous les mêmes connaissances, comment guider les LLMs pour qu'ils s'adaptent à chacun ?

Chaque étudiant développe une **représentation mentale** unique de ses **connaissances**, façonnée par son expérience. Pourtant, les contenus générés par les **LLMs ignorent** souvent ces différences, produisant des explications sans prendre en compte les **prérequis non maîtrisés**, ce qui freine l'apprentissage.

Pour y remédier, chaque étudiant est lié à **un graphe de prérequis personnalisé** où les concepts sont connectés par des **relations de prérequis pondérées** (**weight** ∈ [0,1]) et annotés par une propriété **KNOWN**.

Le graphe est ensuite **encodé** en texte via une fonction d'encodage, et **intégré dans les prompts du LLM** par ingénierie de prompt.
Cette structure permet au système **d'identifier les lacunes critiques** et de générer des **explications adaptées** au niveau réel de l'étudiant.

Méthode envisagée

1. Étudiant
Représente l'apprenant en suivant sa compréhension grâce à une propriété **KNOWN** sur chaque concept.

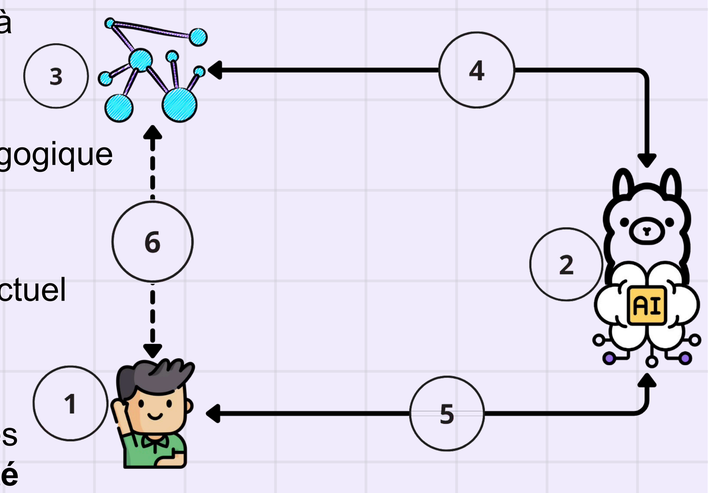
2. LLM
Le LLM est responsable pour la génération de contenu pédagogique adaptatif

3. Graphe de Prérequis
Le graphe de prérequis est destiné à représenter le niveau actuel de connaissances de l'étudiant.

4. Guidage via Graphe
Parcourt le graphe. Si des prérequis manquants sont détectés (**KNOWN** = 0), le LLM est guidé pour les expliquer **en priorité**

5. Interaction d'apprentissage
Cette réponse peut ensuite mettre à jour l'état (**known**) si l'étudiant **valide** l'apprentissage.

6. Feedback Loop
Les connaissances acquises par l'étudiant sont utilisées pour mettre à jour le graphe **dynamiquement**



Scénarios d'apprentissage

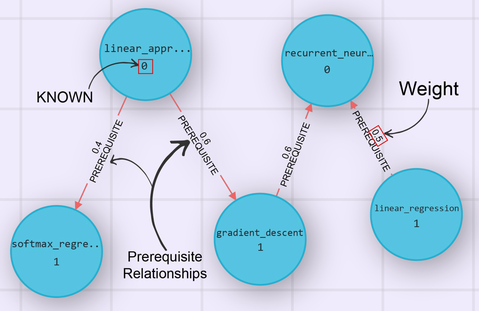
Apprentissage de **la Régression Softmax**

1. Alice sélectionne la régression Softmax comme **concept cible**
2. Le système consulte son graphe de connaissances et détecte que la **régression logistique** et la **descente de gradient** ne sont pas encore maîtrisées.
3. Le LLM commence par **expliquer** ces prérequis manquants, puis aborde la régression Softmax.
4. Les explications et les quiz sont **personnalisés** en fonction de ses connaissances actuelles.
5. Alternativement, Alice peut aussi :
   - Importer un PDF.
   - ❓ Poser une question sur un concept.
   - Utiliser l'approche sur une vidéo et recevoir une explication adaptée selon son niveau.

Graphe de prérequis des concepts

Nous utilisons l'algorithme **ACE**[1] pour construire le graphe :
- Des **paires de concepts** sont extraites du contenu des cours à l'aide d'une **analyse textuelle.**
- Chaque paire est évaluée selon deux critères :
  CSR (Semantic Relatedness)
  CER (Embedding Relatedness)
- Les paires obtenant des scores élevés sont sélectionnées comme **prérequis** potentiels.
- Le **graphe final** est **orienté, pondéré et acyclique** représentant les **relations de prérequis**.

Pour l'utiliser avec un LLM, le graphe est **encodé.** Des études montrent que la qualité de cet encodage influence la capacité à raisonner.[2]



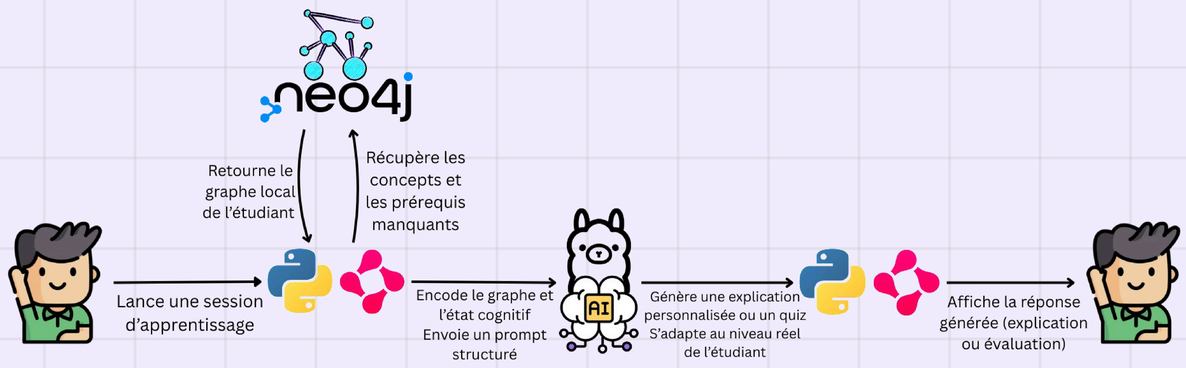
Large Language Model (LLM)

Un LLM **(Large Language Model)** est un modèle de langage de grande taille entraîné sur de vastes corpus textuels.
Il est capable de comprendre et de générer du **langage naturel**, de répondre à des questions, d'expliquer des concepts ou encore de créer des quiz.
Dans ce projet, il est utilisé pour adapter dynamiquement les contenus pédagogiques au **niveau** de chaque **étudiant**.

Prompt Engineering

Consiste à fournir au **LLM** un petit nombre d'exemples structurés (input/output) pour orienter la génération de réponses pertinentes.
- Aident à **encoder** implicitement des schémas de raisonnement sur graphe
- Guident le modèle dans l'interprétation de chaînes de dépendance entre concepts
- Réduisent l'ambiguïté dans les tâches de génération conditionnelle (quiz, explication)

Ce mécanisme améliore la capacité du modèle à raisonner sur des représentations graphes textuelles, en particulier lorsqu'il est combiné à un encodage structuré des relations de prérequis[2]



Résultats de cette Approche

Approche Générique

```
explanation_prompt = f"""
Explique Régression linéaire:
- Précise à quoi il sert et pourquoi il est important.
- Utilise des exemples concrets et des analogies.
- Propose des ressources pour approfondir. """
```

Explique le contenu pédagogique sans tenir compte du niveau actuel de l'étudiant

Linear regression is a statistical method that models the linear relationship between a dependent variable and one or more independent variables. It fits a line (e.g., $y = mx + b$) that minimizes the error between predicted and actual values using a loss function like mean squared error (MSE). To minimize this loss, gradient descent is used, where backpropagation computes gradients of the loss with respect to model parameters. The chain rule from calculus is applied during backpropagation to efficiently update weights and improve predictions.

1. Generalization: Generalization refers to the ability of a machine learning model to make accurate predictions on unseen data, by learning patterns from training data and avoiding overfitting.
2. Softmax Regression: It is a generalization of linear regression used for multi-class classification problems. Softmax function maps outputs of a neural network into probabilities that sum up to 1, allowing us to predict class memberships in multi-class tasks. For example, the output layer of an image classification model using softmax regression would give probabilities for each class (e.g., dog, cat, car) rather than a single value (like in linear regression).
3. Binary Linear Classifier: This is a special case of linear classifiers designed to distinguish between two classes. In a binary classification problem, input features are passed through a linear function and the sign of the output determines the predicted class. For instance, considering whether an email is spam or not based on features such as the number of exclamation marks or capital letters can be seen as a binary linear classifier.

Notre Approche

Obtention des prérequis à partir du graphe des prérequis

```
get_prerequisites(tx, concept):
    query = """
    MATCH (c:Concept)-[:PREREQUISITE]->(p:Concept)
    WHERE c.name = $concept
    RETURN p.name AS prerequisite
```

Injecter le concept dans le prompt

```
explanation_prompt = f"""
                Explique le concept '{concept}' de manière claire et accessible :
                - Précise à quoi il sert et pourquoi il est important.
                - Utilise des exemples concrets et des analogies.
                - Propose des ressources pour approfondir.
                """
```

Vérifier la valeur de la propriété {KNOWN} dans le nœud du concept

```
if known == 0:
with driver.session() as session:
prerequisites = session.read_transaction(get_prerequisites, concept)
if prerequisites:
prereq_text = f"""
Pour bien comprendre le concept '{concept}', il faut connaître les notions suivantes : {', '.join(prerequisites)}.
Explique ces prérequis simplement, avec des exemples concrets, et propose des ressources pour les étudier.
"""
else:
prereq_text = "Ce concept ne nécessite aucun prérequis particulier.\n"

explanation_text = prereq_text + "\n\n" + explanation_prompt
elif known == 1 :
explanation_text = explanation_prompt
```

Explication des prérequis s'ils ne sont pas maîtrisés

	Knowledge Graph + LLM	ChatGPT (Modèle Générique)
Approche pédagogique	Guidée par des concepts structurés dans l'EKG (prérequis → compréhension progressive)	Réactive aux questions, sans parcours structuré ou suivi des prérequis
Génération de quiz/exemples	Générés dynamiquement à partir du concept + PDF + quiz	Peut en générer si demandé, mais sans lien avec un graphe/pdf
Suivi de progression	Oui, via known: 0/1 dans Neo4j, avec quiz + validation par score	Non : pas de structure de progression suivie nativement
Interaction	concept → quiz → explication → PDF → synthèse	on peut poser n'importe quelle question à tout moment
Objectif principal	Aider l'étudiant à maîtriser progressivement les concepts	Répondre à tout type de question, pas forcément dans un but d'apprentissage structuré
Confidentialité	Haute : tout peut tourner localement	Dépend des serveurs OpenAI (données peuvent transiter)

Conclusion

Cette recherche montre qu'en alignant des modèles de langage avec des graphes de connaissances structurés, on peut créer un système éducatif personnalisé. En combinant l'algorithme ACE et le prompting, notre approche permet aux LLMs de fournir des explications adaptées au niveau de l'étudiant tout en enrichissant son graphe de connaissances.

Références
(1) Aytekin, M. C., & Saygın, Y. (2024). ACE: AI-Assisted Construction of Educational Knowledge Graphs with Prerequisite Relations. Journal of Educational Data Mining, 16(2), 85–114. https://doi.org/10.5281/zenodo.14250896
(2) Bahare Fatemi, Jonathan Halcrow, Bryan Perozzi (2023). Talk like a Graph: Encoding Graphs for Large Language Models. https://arxiv.org/abs/2310.04560