

LLM for Coding JAVA

« JavaDevLLM »

Augmented education in the age of generative artificial intelligence.

AI-assisted coding for JAVA programming

Enseignement augmentée dans l'ère de l'intelligence artificielle générative.

Assistance par l'IA en codage pour la programmation en JAVA

Grace à l'évolution des intelligences artificielle générative à base des LLM (Large Language Model), la disponibilité de l'information fiable et correcte semble être aujourd'hui une réalité. Les (LLM) sont une catégorie de modèles de base formés sur d'immenses quantités de données, ce qui les rend capables de comprendre et de générer du langage naturel et d'autres types de contenu pour effectuer un large éventail de tâches. Les LLM fonctionnent en s'appuyant sur des techniques d'apprentissage profond et de grandes quantités de données textuelles. Ces modèles sont généralement basés sur une architecture de transformateur, comme le transformateur génératif pré-entraîné, qui excelle dans le traitement des données séquentielles telles que les entrées textuelles. Les LLM sont constitués de plusieurs couches de réseaux neuronaux, chacune ayant des paramètres qui peuvent être ajustés avec précision pendant l'entraînement, et qui sont encore améliorés par de nombreuses couches connues sous le nom de mécanisme d'attention, qui se concentrent sur des parties spécifiques des ensembles de données.

Le paysage de l'enseignement surtout de la programmation est entrain de changer, car les LLM pour l'assistance au codage ont change complètement la vision des enseignants d'une part, et d'autre part la façon dont les développeurs rédigent, déboguent et optimisent le code. La mission actuellement des LLM est très large à pour objectif d'aider les développeurs dans une multitude de tâches de programmation. Ces missions peuvent allant de l'accélération des tâches de programmation jusqu'au l'écriture de fonctions. Il faut noter que les LLM sont entraînés sur une très grande quantité de données (sous forme de code), et ils sont aussi capable d'écrire et d'expliquer un nouveau code, et de déboguer des erreurs. LLM est un outil de travail que l'enseignant et le développeur peuvent les intégrés dans leurs quotidiens du travail.

Actuellement, les LLM sont de grande importance pour les développeurs. Les LLM permettent aux développeurs d'être plus efficace dans le travail, de se concentrer sur des tâches stratégiques, éliminer les opérations répétitives et minimiser le temps allouer pour chaque projet réaliser. Les LLM réduisent éventuellement les erreurs humaines dans la programmation, mettant à disposition des développeurs des suggestions et des corrections syntaxiques en temps réel. Les LLM améliorent l'apprentissage, ce qui peut être un outil pour l'enseignant à former des ingénieurs augmentés. Nous pouvons le considérer comme un accompagnateur et agir comme un tuteur pour expliquer et aider les étudiants / développeurs à comprendre le fonctionnement. Les LLM améliorent la collaboration entre les équipes des développeurs, en gardant des normes de codage entre les différents membres.

Dans ce projet, nous souhaitons développer un LLM open source pour assister les développeurs ainsi que les étudiants en codage de langage de programmation JAVA via l'intelligence artificielle. Le LLM développé « JavaDevLLM » est principalement un outil de support pour que les étudiants / développeur se concentre sur la logique en automatisant l'assistance dans le codage. « JavaDevLLM » sera en mesure d'assister les développeurs dans la génération de code, l'explication d'algorithmes complexes et notation de base, proposition des suggestions de code en temps réel, le support pour débogage, et la suggestion à des codes ou programmation en répondant à des requêtes en langage naturel. Les fonctionnalités principales de « JavaDevLLM » sont :

- Génération améliorée de code et d'explications
- Raisonnement et débogage
- Compréhension du langage naturel
- Prise en charge multilingue

Il est recommandé pour « JavaDevLLM » d'utiliser LLAMA 3, car cette version est connue par ses capacités avancées de raisonnement et de génération de code : <https://ai.meta.com/blog/meta-llama-3/>

Des notions doivent être maîtriser pour un meilleur développement :

- Fine-tuning
- Vector DB
- RAG
- Embedding
- LLM
- NLP

Quelques liens utiles :

<https://github.com/mukel/llama3.java/blob/main/Llama3.java>

<https://github.com/mukel/llama3.java>

<https://github.com/tjake/llama>

<https://ai.meta.com/blog/meta-llama-3/>

<https://ai.meta.com/blog/code-llama-large-language-model-coding/>

<https://github.com/meta-llama/codellama>