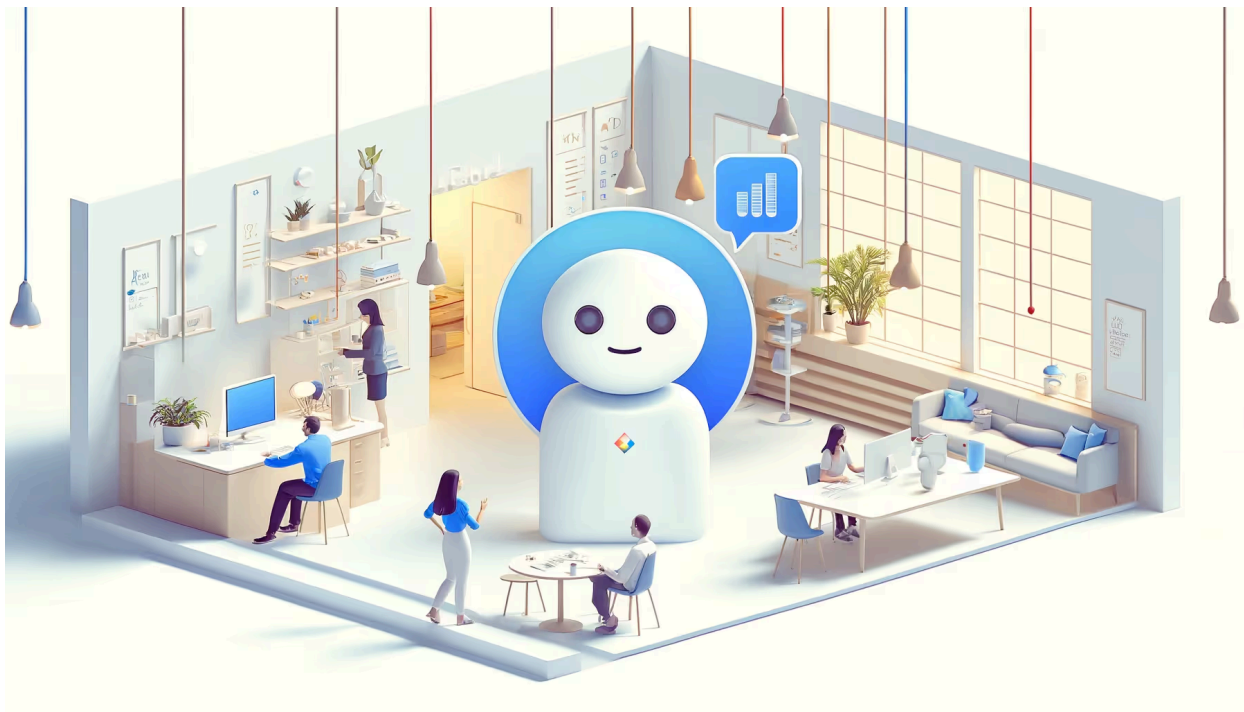


Comment transformer les LLMs en Agents Intelligents ?

Développement des Capacités Agentielles pour les LLMs afin d'automatiser les flux de travail et créer des assistants intelligents



En quelques mois seulement depuis l'émergence des modèles basés sur les transformateurs, les grands modèles de langage (LLMs) ont démontré leur capacité étonnante à engager des conversations semblables à celles des humains, révolutionnant la productivité des entreprises et rendant les moteurs de recherche traditionnels plus ou moins obsolètes..

Cependant, cette capacité est limitée aux informations sur lesquelles ils ont été entraînés ou aux informations disponibles publiquement via les recherches sur Internet, tandis que les données d'entreprise restent inaccessibles..

La génération augmentée par la récupération (RAG) tente de pallier cette limitation en permettant à un modèle de répondre en se basant sur une base de données de documents, fournissant ainsi des réponses justifiées et réduisant considérablement les hallucinations. Cependant, leur portée reste limitée à la réponse aux questions plutôt qu'à l'exécution de tâches de manière autonome, en utilisant différents outils à leur disposition (recherche sur Internet, recherche dans une base de données, reformulation/clarification des questions des utilisateurs...) pour accomplir des tâches complexes..

Ce processus d'apprentissage par la pratique est une révolution en devenir pour les LLMs : on parle de développement de "capacités agentielles"

1. Qu'est-ce qu'un Agent ?

Pour construire un agent efficace, nous pouvons exploiter les capacités intrinsèques des LLMs, qui vont au-delà des simples tâches de traitement du langage naturel (NLP) comme la compréhension et la génération de texte, pour inclure de véritables capacités de raisonnement. Les avancées récentes en IA générative se concentrent sur l'amélioration des LLMs avec des capacités agentielles, leur permettant de planifier et d'exécuter des tâches.

Les capacités agentielles impliquent l'utilisation d'une gamme d'outils pour atteindre des objectifs spécifiques. Pour accomplir des tâches complexes, un agent doit sélectionner les bons outils pour le travail et les utiliser efficacement. Cela nécessite que l'agent comprenne la tâche à accomplir, identifie les outils nécessaires et les applique dans le bon ordre.

Par exemple, si un utilisateur pose cette question : « Répondez à cette plainte client après avoir examiné notre contrat et recherché les articles pertinents du code civil. »

La liste des tâches de l'agent pourrait ressembler à ceci ::

1. Clarifier la question de l'utilisateur pour s'assurer de la compréhension
2. Analyser le document de plainte client joint
3. Récupérer le contrat du client dans la base de données de l'entreprise
4. Rechercher les articles pertinents du code civil
5. Générer une réponse à la plainte client

En décomposant des tâches complexes en étapes gérables et en utilisant les bons outils pour chaque étape, l'agent peut répondre efficacement à la plainte du client et fournir une solution de haute qualité.

2. Qu'est-ce qu'un tool ?

Un tool est un flux de travail prédéfini conçu pour accomplir une tâche spécifique, qui peut être accédé et utilisé par un agent pour atteindre un objectif particulier ou répondre à une demande utilisateur..

En ce sens, un outil est un module ou un composant autonome qui exécute une fonction spécifique ou un ensemble de fonctions, et peut être combiné avec d'autres outils pour accomplir des tâches plus complexes. Les outils peuvent être considérés comme des blocs de construction qu'un agent peut utiliser pour construire une solution à la demande d'un utilisateur..

Exemples de tool:

- Un module NLP qui peut analyser un texte et extraire des informations pertinentes
- Un outil de requête de base de données qui peut récupérer des données spécifiques d'une base de données
- Un modèle de machine learning qui peut générer du texte ou faire des prédictions en fonction des données d'entrée
- Un outil d'automatisation de flux de travail qui peut exécuter une série de tâches dans un ordre spécifique

En accédant et en utilisant ces outils, un agent peut accomplir des tâches complexes et répondre aux demandes des utilisateurs de manière plus efficace..

3. Les avantages du développement des capacités agentielles : un meilleur ancrage du LLM dans la réalité

Cette augmentation agentielle du LLM présente un triple avantage ::

Étendre l'utilisation du LLM à la planification d'actions tout en exploitant mieux ses capacités de raisonnement. À cet égard, ce premier avantage est pour l'utilisateur du LLM.

Avoir un accès augmenté au monde extérieur grâce à l'utilisation d'outils dédiés, étendant ainsi (et potentiellement mettant à jour, comme dans RAG) les informations auxquelles le LLM peut accéder au-delà des données vues pendant sa phase d'entraînement..

Exploiter des modèles prédéfinis pour exécuter des tâches spécifiques et atteindre des objectifs avec facilité..

De plus, grâce à sa performance agentielle, le LLM reçoit de nouveaux retours de ses actions et augmente ainsi ses capacités de raisonnement tout en apprenant à se confronter au monde extérieur. .

Les capacités agentielles peuvent être considérées comme une première étape pour ancrer le LLM dans le monde réel..

4. L'exemple de TRICE, une approche en deux étapes pour le développement des capacités agentielles

Plusieurs articles de recherche sont actuellement publiés, reflétant l'intérêt croissant pour les agents. L'article "Toolformer" explique comment les LLMs peuvent "s'auto-apprendre à utiliser des outils externes" via des API simples.

La méthodologie CRAFT (Customizing LLMs by Creating and Retrieving from Specialized Toolsets) vise à créer des ensembles d'outils spécialement conçus pour certaines tâches données et à équiper les LLMs d'un composant qui récupère des outils de ces ensembles pour améliorer leur capacité à résoudre des tâches complexes..

Une expérience de recherche récente introduit une nouvelle méthodologie : TRICE (Tool learning with execution feedback) qui s'appuie sur la capacité d'ancrage du LLM dans le monde réel grâce à l'utilisation d'outils. Le LLM est formé à utiliser des outils en deux étapes ::

Une étape de formation comportementale, où le LLM apprend à utiliser un outil par imitation comportementale

Une étape de renforcement par retour d'exécution (RLEF, Reinforcement Learning through Execution Feedback) qui aide le LLM à sélectionner l'outil approprié parmi un ensemble d'API disponibles afin d'accomplir une tâche donnée..

L'expérience montre que cette méthodologie surpasse d'autres méthodologies comme "Toolformer" et donne un aperçu de la manière dont un LLM agentiel peut potentiellement évoluer vers des capacités de résolution de tâches complexes générales lorsqu'il est correctement exploité à travers la dimension d'ancrage de son rôle d'agent. En particulier, pour le raisonnement mathématique, qui est l'un des cas d'utilisation les plus difficiles pour l'utilisation classique du LLM, l'approche TRICE utilise l'outil Calculatrice, tout comme l'expérience "Toolformer", mais surpasse "Toolformer" dans cet exercice.

5. La capacité agentielle : surmonter les limitations traditionnelles des LLMs pour construire des flux de travail automatiques

La différence de performance entre "Toolformer" et "TRICE" est en quelque sorte reconnue dans les deux articles, soulignant les limitations intrinsèques du LLM et comment les deux méthodologies ont réussi à les surmonter. Dans la méthodologie "Toolformer", les principales limitations proviennent de deux facteurs ::

Le LLM est très sensible à l'ingénierie des invites : à cet égard, la phase de formation de la méthodologie TRICE vise à améliorer la précision avec laquelle le LLM parvient à identifier l'outil adéquat à utiliser pour résoudre un cas d'utilisation donné, surmontant l'ambiguïté liée à la phase d'invite..

Le LLM est limité à ses connaissances pré-entraînement et n'est pas ancré dans la réalité : la méthodologie TRICE permet un meilleur retour d'utilisateur lors de la sélection des outils adéquats pour résoudre un problème..

Pour une utilisation commerciale, les capacités agentielles du LLM permettent la création de flux de travail en entreprise en exploitant mieux les capacités de raisonnement du LLM tout en surmontant ses limitations traditionnelles (longueur des fenêtres de contexte, manque de mémoire appropriée, ingénierie des invites complexe, capacités limitées de zéro-coup, hallucinations...).

LightOn a mis en œuvre sa propre version des capacités agentielles dans Paradigm comme une version augmentée de son "constructeur de tâches" actuel et de chat avec documents (RAG) déjà disponible avec la version précédente de Paradigm (voir le prochain blog à venir).

[1] Retrieval Augmented Generation: LLMs instructions are Augmented with information available in a private database

[2] "BOLAA: BENCHMARKING AND ORCHESTRATING LLM-AUGMENTED AUTONOMOUS AGENTS", arXiv:2308.05960v1 [cs.AI] 11 Aug 2023

[3] "CRAFT: CUSTOMIZING LLMS BY CREATING AND RETRIEVING FROM SPECIALIZED TOOLSETS", arXiv:2309.17428v1 [cs.CL] 29 Sep 2023

[4] "Making Language Models Better Tool Learners with Execution Feedback", arXiv:2305.13068v1 [cs.CL] 22 May 2023