[sfeir] WENVISION

TECHWAVES - #GENAI FOR DEV

L'IA générative pour les CIO et les CTO

RELEASE 4 - Octobre 2023 - Fiches mises à jour signalées par :

MAJ - R4

“L'IA GÉNÉRATIVE N'EST PAS UNE MENACE POUR LES DEV, AU CONTRAIRE”

L’ÉDITO

Olivier Rafal
Consulting Director - Strategy

olivier@wenvision.com

Vous avez tous lu comme moi que l’IA générative allait augmenter la productivité des développeurs jusqu’à, in fine, supprimer des emplois d’ici peu. Voire le métier même de développeur. Vous y croyez, vous ? Pas moi.

Oui, **l’IA générative augmente considérablement la productivité des développeurs. C’est un formidable assistant, qui va réaliser les tâches les plus rébarbatives** : écrire le code pour requêter une API, ajouter des champs dans une interface Web ou traduire un code d’un langage vers un autre. Des tâches normées, fastidieuses, sans valeur ajoutée.

L’assistant ira éventuellement plus loin, générant une grande partie du code applicatif souhaité, à partir de prompts, voire de schémas à main levée - une implémentation moderne, dopée à l’IA, de l’esprit du Model Driven Development, né au début des années 2000, où on envisageait de concevoir l’ossature d’un logiciel à partir de sa modélisation en UML.

Le MDD n’a jamais décollé. Trop ambitieux, trop rigide, pas suffisamment performant, etc. De par leur souplesse, leur performance et leur simplicité d’utilisation, les outils de développement intégrant de l’IA générative ont eux toutes les chances de réussir.

POURQUOI L'IA NE SUPPRIMERA PAS DE POSTE DE DÉVELOPPEURS

Cela ne veut pas dire pour autant que les nouveaux outils de dev intégrant de l'IA diminueront le besoin en développeurs dans les années qui viennent. Trois réflexions à ce sujet :

- **Il faut piloter l’IA**. L’IA génère du code demandé par le développeur, c’est lui qui comprend les souhaits du PO, qui a l’architecture du logiciel et son contexte d’utilisation en tête, et doit affiner les prompts en fonction de ça.
- **Il faut vérifier le code produit par l’IA**. IA générative ou pas, un développeur vérifie et teste le code. Toujours.
- **Il faut créer toujours plus de code**. Penser que l’augmentation de productivité va supprimer des emplois à court ou moyen terme, c’est croire que nous devons produire une quantité finie de code. Alors qu’aujourd’hui tech et business sont indissociables, que le logiciel est partout et surtout qu’il doit évoluer constamment.

Non seulement les développeurs ont encore de beaux jours devant eux, mais en plus, l’IA générative leur simplifiera grandement la tâche. Pour la plus grande satisfaction des DSI qui la mettront en œuvre.

LA PAROLE AUX EXPERTS

Consulting

Marie Fontaine
Head of GenAI

marie@wenvision.com

La rapidité d'évolution, à la fois technologique et d'usage, de l'IA Générative est sans précédent. Il est devenu difficile de suivre la production de connaissances dans ce secteur et de comprendre les effets possibles sur nos entreprises. L'histoire commence en 2020, lorsqu'OpenAI ouvre les accès à GPT-3 en Bêta, mais c'est avec ChatGPT que le grand public prend conscience de la puissance de cet outil.

Ce sentiment de dépassement se confronte à celui de l'urgence. Les gains de productivité attendus sont tels qu'aucune entreprise ne peut ignorer cette technologie. En particulier lorsqu'on parle de développement logiciel. Une étude menée par GitHub montre que les développeurs assistés par Copilot mettent 56% moins de temps à accomplir une tâche. McKinsey projette des gains de productivité d'au moins 30% pour le software engineering. Cela fait naître de nouveaux défis pour les entreprises : choisir les bons outils, concilier l'adoption de cette technologie et accompagner les collaborateurs.

Chez SFEIR et WENVISION, nous suivons cette tendance depuis de nombreux mois et nous souhaitons partager notre retour d'expérience au travers des TechWaves - #GenAI 4 Dev. Toute approche stratégique de l'IA Générative nécessite une acculturation, une feuille de route basée sur des cas d'utilisation réels et la mise en place d'une organisation adaptée.

Engineering

Florent Legras
Engineering Director

legras.f@sfeir.com

Les LLM offrent une solution qui permet **d'améliorer largement les applications existantes**. Par exemple, offrir aux utilisateurs métier la possibilité de bénéficier réellement d'un self-service BI, en dialoguant. Pour les développeurs, ces outils permettent **d'augmenter leur productivité** et de résoudre certaines problématiques plus efficacement.

Le déploiement d'un LLM en entreprise nécessite de passer d'un simple POC à une application prête pour la production. Cette transition revêt une importance cruciale. Bien qu'un **POC puisse démontrer les fonctionnalités de base d'une application LLM**, il peut ne pas tenir compte de facteurs essentiels tels que la scalabilité, les performances et la sécurité. **Or, le passage d'un POC à une application déployable est un exercice complexe en raison de la nature versatile des LLM.**

Ces applications doivent être conçues et pensées dans un contexte global, intégrées dans une plateforme data au sein d'un système d'information. Une architecture bien pensée permet la scalabilité, la maintenabilité et l'extensibilité de l'application. Une application prête pour la production comprend des mécanismes pour surveiller et gérer les modèles. Cela implique de suivre les performances, d'identifier les problèmes, ainsi que de garantir la sécurité et la conformité de l'application. Sans ces fonctionnalités, une application LLM risque de générer plus de désagréments que de satisfaction.

“CODER AVEC UNE IA GÉNÉRATIVE ? J’AI RELEVÉ LE DÉFI”

Engineering

Didier GIRARD
Co-CEO SFEIR & WENVISION

didier@wenvision.com

Au cours de l'année écoulée, plusieurs activités ont limité mon temps de programmation, réduisant mes sessions de codage à quelques heures par mois.

Je considère qu'il est crucial de coder régulièrement et, à force de ne pas pratiquer, j'ai commencé à perdre certains réflexes, un peu comme un musicien qui arrête de jouer de son instrument et perd peu à peu sa dextérité.

Cela m'a poussé à relever un défi : **serait-il possible de développer une application basée sur une IA générative, malgré mes compétences un peu rouillées ?**

Pour ce faire, j'ai utilisé Replit et son IA générative, un outil pratique qui ne nécessite aucune installation. Mon objectif était de recréer une application que j'avais développée en Flutter il y a plusieurs années pour compenser ma dyslexie : lorsque je suis fatigué, mon handicap se manifeste et je fais des "fautes de frappe".

L'application Flutter originale traduit mon texte du français à l'anglais, puis de l'anglais au français. Elle est mon assistant personnel. Pour la traduction, elle s'appuie sur l'API de Deepl.

J'aime le langage Go, j'ai donc choisi de travailler avec Replit en Go. L'interface utilisateur a également été conçue par Replit, d'abord en utilisant un formulaire basique, puis en s'appuyant sur Bootstrap.

“J'ai relevé le défi de coder une application avec Replit et son IA générative. Résultat ? Un développement rapide et efficace ! L'IA ne remplacera pas les développeurs, mais nous permettra de coder plus vite, tout comme le rabot électrique a aidé les menuisiers.”

Le développement était très simple et a pris environ 2 à 3 heures.

Enthousiasmé par cette rapidité de développement, j'ai décidé d'ajouter une nouvelle fonctionnalité à mon assistant : la génération de texte. Pour ceci, je me suis appuyé sur GPT-4 d'OpenAI. L'idée étant qu'avec cette évolution, mon assistant personnel m'aide maintenant dans la formulation de mes textes et ne soit plus un simple relecteur. Avec Replit, intégrer l'API d'OpenAI ne m'a pris que quelques dizaines de minutes. Pour l'instant, l'intégration est basique, mais au final, mon assistant correspond à mes besoins.

Cette expérimentation avec Replit a confirmé ce que je soupçonnais. L'arrivée de l'IA générative pour aider les développeurs est une belle avancée. D'ici quelques trimestres, les développeurs les plus efficaces utiliseront ce genre d'outils.

Toutefois, certaines compétences sont nécessaires pour utiliser correctement l'IA générative dans le développement. D'après mon expérience, bien que limitée à quelques heures, je pense que la capacité à déboguer est essentielle. En effet, **l'IA ne génère pas tout parfaitement du premier coup, et être capable d'identifier rapidement les problèmes pour demander à l'IA de générer une nouvelle solution est crucial**.

Je vous encourage à essayer ces outils. Le travail du développeur va changer radicalement avec l'IA, mais elle ne remplacera pas les développeurs. Tout comme le rabot électrique n'a pas remplacé le charpentier, l'IA nous permettra simplement de coder plus rapidement et plus efficacement.

A suivre...

Pour les curieux, voici le repo GitHub : https://github.com/dgirard/wizwizwiz17/tree/main

-> Ce n'est probablement pas le plus beau code, mais disons que c'est une IA qui code depuis moins d'un an qui l'a produit. Dans les années à venir, elle fera mieux !

TECHWAVES #GENAI FOR DEV, MODE D'EMPLOI

[sfeir]

WENVISION

MÉTHODOLOGIE DE NOS TECHWAVES

Les Techwaves sont le résultat d'une méthodologie éprouvée chez SFEIR et WEnvision, pour sélectionner les technologies les plus pertinentes au sein d'un domaine. Celles qui permettront de faire émerger les ambitions des entreprises sans accroître leur dette technique.

Pour ce faire, nous évaluons divers critères, comme la facilité de prise en main, la constitution d'une communauté, la disponibilité de modules de formation voire de livres, etc.

Un exercice possible grâce au crowdsourcing interne au groupe, qui nourrit aussi bien notre veille technologique que nos analyses.

Le résultat de ces analyses permet de positionner les technologies sur notre courbe, découpée en 6 secteurs →

1.

EXPÉRIMENTATION

Concerne les développeurs et entreprises innovantes, start-up, qui cherchent des moyens d'obtenir un avantage décisif grâce à la technologie. Ils testent beaucoup, réalisent des "proofs of concept".

2.

DÉCOLLAGE

La technologie est adoptée par les start-up et DSI qui sont prêts à miser beaucoup dessus afin d'obtenir un avantage différenciant. Pour eux, il est vital que le projet aboutisse.

3.

CROISSANCE

Les entreprises un peu plus prudentes, mais qui souhaitent tout de même pouvoir s'appuyer rapidement sur des technologies innovantes, adoptent à leur tour la technologie.

4.

PLATEAU

La technologie a trouvé son marché. Les plus conservateurs, qui souhaitent partir sur des technologies éprouvées, l'ont adoptée. Le marché est capable de répondre à la demande.

5.

DÉCROISSANCE

De moins en moins de nouveaux projets adoptent cette technologie. L'usage est en décroissance. L'utilisation de cette technologie peut être assimilée à de la dette.

6.

MENACE

L'utilisation de cette technologie sur un nouveau projet est à proscrire. Peu d'acteurs sont encore impliqués dessus.

NOS CHOIX POUR CES TECHWAVES : 9 FAMILLES D'OUTILS #GENAI POUR ACCÉLÉRER LES DEV OU BOOSTER UNE APP

MAJ - R4

01 ***LLM généraliste*** *: utile pour le rédactionnel.*

02 ***LLM pour le code*** *: pour composer du code, le traduire, etc.*

03 ***Assistants de programmation*** *: le futur de vos IDE*

04 ***Agents conversationnels*** *: pour faciliter les interactions avec l'IA*

05 ***Orchestrateurs*** *: pour faciliter les interactions*

06 ***GUI*** *: construire des interfaces graphiques augmentées*

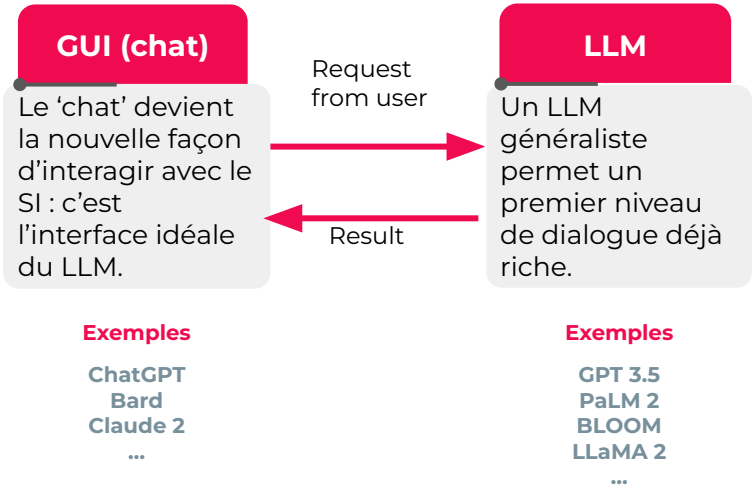
07 ***Prompt tools*** *: pour optimiser l'usage des LLM*

08 ***Vector DBs*** *: pour valoriser l'information interne*

09 ***Embedding tools*** *: pour faciliter l'exploration de vos données*

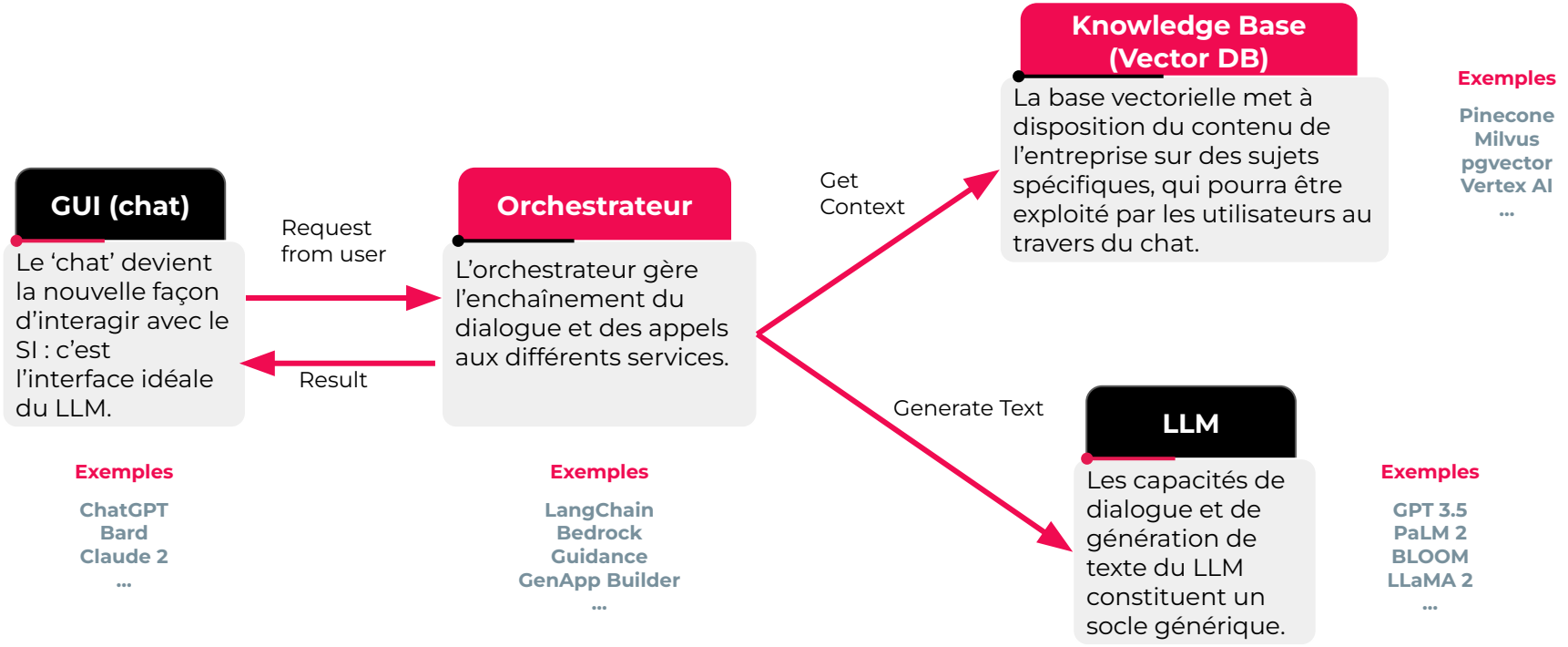
COMMENT LES TECHNOS SE POSITIONNENT ENTRE ELLES : NOTRE VISION D'UNE "POC LLM APP STACK"

MAJ - R4



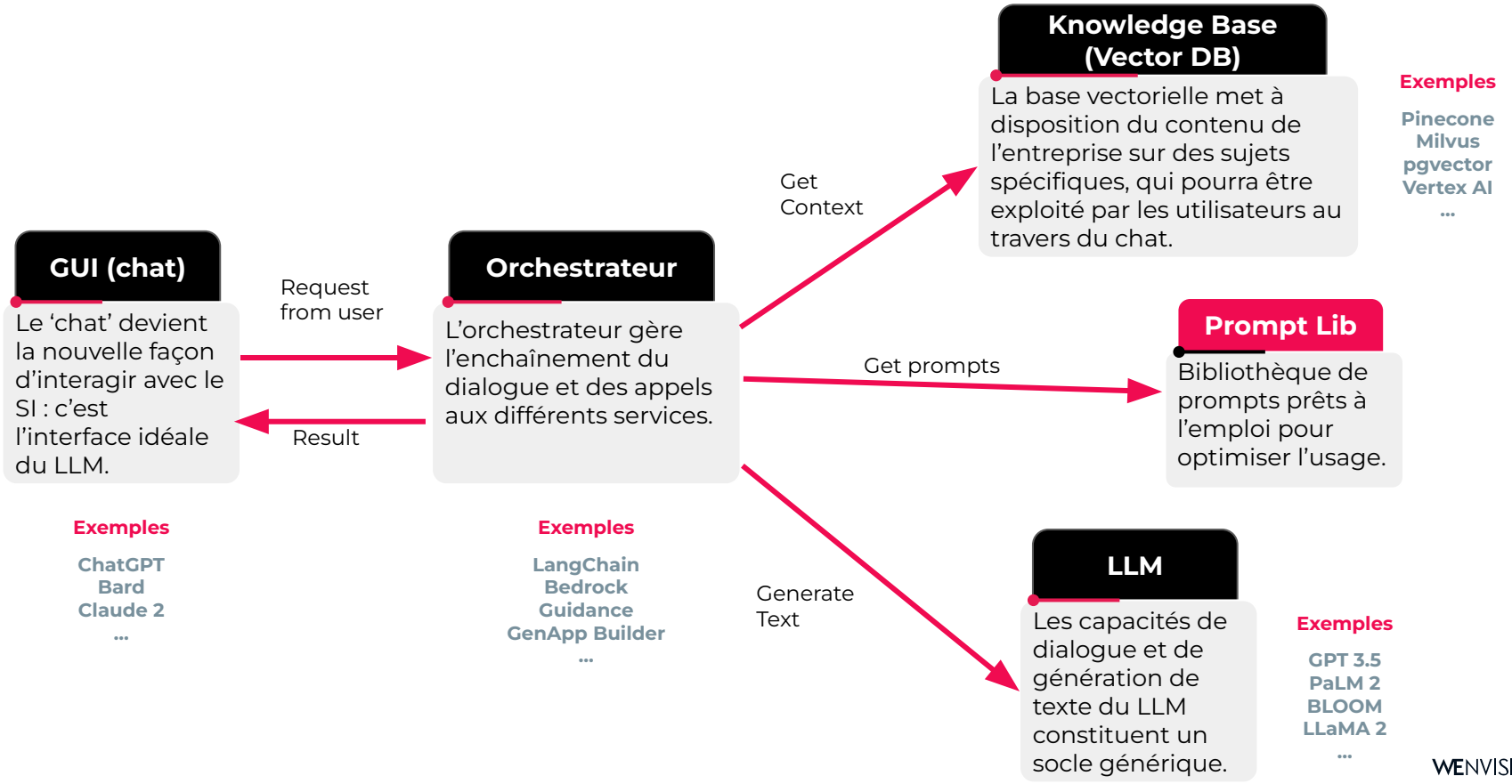
COMMENT LES TECHNOS SE POSITIONNENT ENTRE ELLES : NOTRE VISION D'UNE "MVP LLM APP STACK"

MAJ - R4



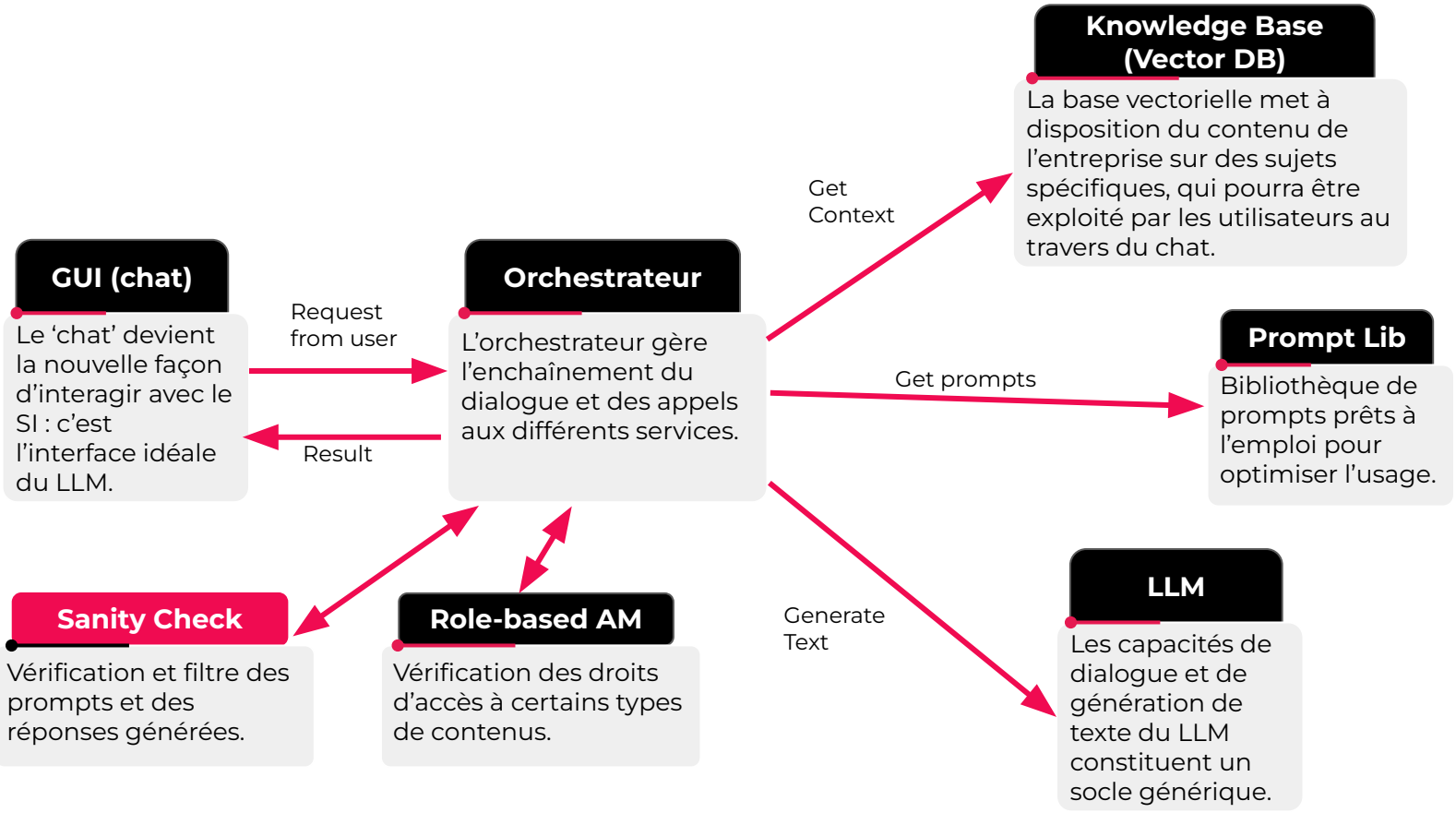
COMMENT LES TECHNOS SE POSITIONNENT ENTRE ELLES : NOTRE VISION D'UNE "CLASSIC LLM APP STACK"

MAJ - R4



MAJ - R4

COMMENT LES TECHNOS SE POSITIONNENT ENTRE ELLES : NOTRE VISION D'UNE "ADVANCED LLM APP STACK"



APPLICATION DES PRINCIPES #GENAI AUX SGBD (MVP)

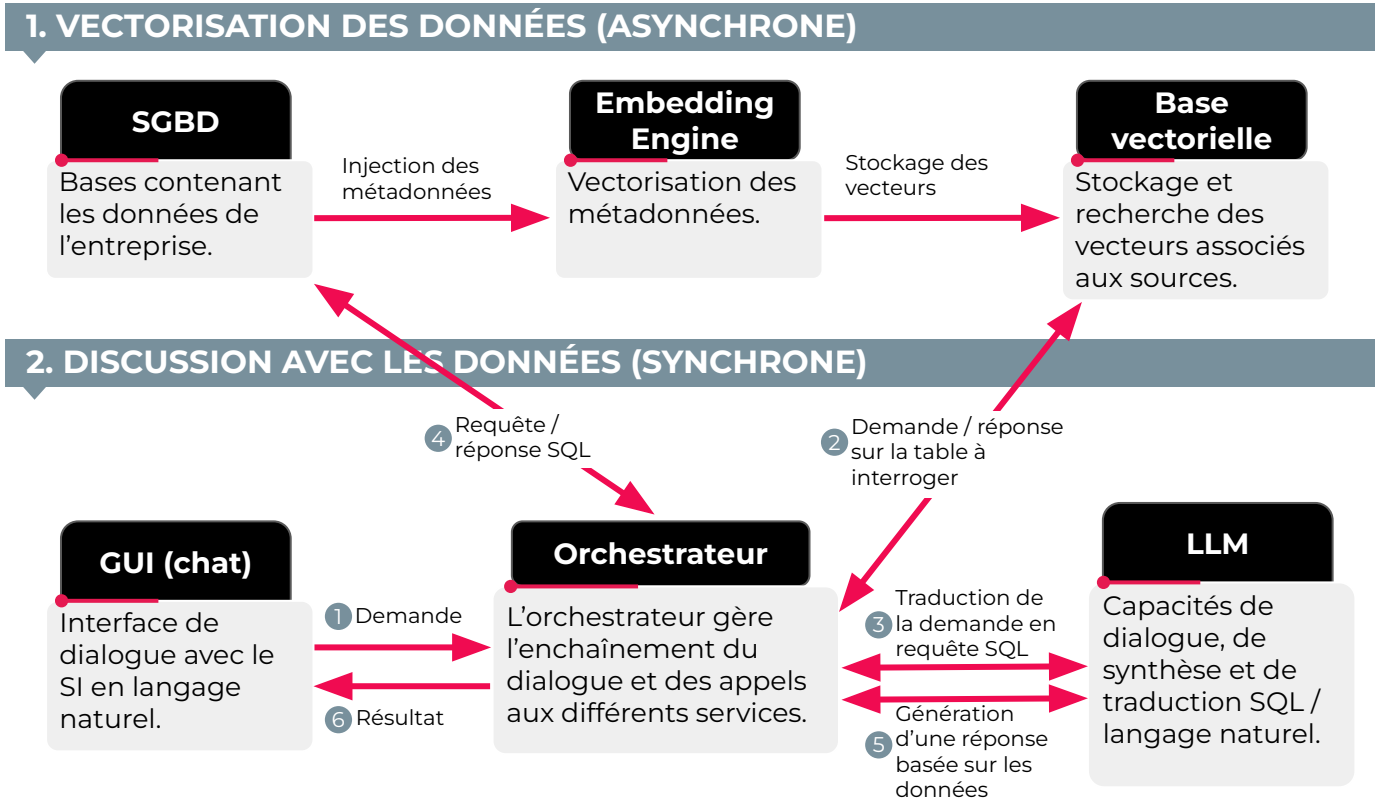
MAJ - R4

Le dialogue est **une forme idéale pour interroger les données de l'entreprise**. Il s'agit non plus de requêter des données, mais de collaborer avec un assistant qui va répondre à des demandes formulées en langage naturel.

Une telle architecture se décompose en 2 grands domaines : d'une part la préparation des données, qui s'effectue de manière asynchrone, d'autre part la gestion du dialogue, en temps réel.

Attention cependant à cette approche :

- elle demande que les bases de données soient correctement renseignées et les métadonnées intelligibles ;
- elle introduit des risques de sécurité (SQL injection).



FINE-TUNING VS RETRIEVAL AUGMENTED GÉNÉRATION (RAG)

MAJ - R4

FINE-TUNING : DÉFINITION

Le fine-tuning consiste à adapter un LLM à des tâches spécifiques ou à des domaines particuliers. Le processus implique de prendre un modèle pré-entraîné et de le réentraîner sur un ensemble de données plus restreint et ciblé, souvent constitué de textes spécifiques à la tâche envisagée. En ajustant les poids et les paramètres du modèle d'origine, le fine-tuning permet au LLM de mieux comprendre et de générer des réponses plus précises et adaptées à des contextes particuliers.

CAS D'USAGE DU FINE-TUNING

- Adapter un modèle à un contexte particulier, par ex. :
  - aider un développeur,
  - adopter des tournures de phrases juridiques ou littéraires,
  - comprendre du vocabulaire propre à un domaine métier...

RAG: DÉFINITION

Le "RAG" tire parti des capacités de recherche intégrées aux LLM pour interroger une base de données vectorielle (voir fiche associée) ou des documents afin d'en extraire des informations pertinentes en réponse à une requête donnée. Ces informations récupérées sont ensuite utilisées pour enrichir la génération de texte, offrant ainsi une réponse plus contextuelle et informée, s'appuyant sur les connaissances de l'entreprise et plus seulement sur les connaissances générales du modèle.

CAS D'USAGE DU RAG

- Enrichir le modèle avec les données de l'entreprise (tout en préservant le contrôle d'accès et la possibilité de supprimer ou mettre à jour le contenu), par ex. :
  - données financières,
  - descriptifs produits,
  - documentation,
  - framework technique...

LES PLUS

- Capacité à adapter le modèle à son contexte (par ex. le retail, le BTP, le droit, etc.) ou à sa tâche (répondre à un client, donner des instructions, etc.)

LES MOINS

- Processus relativement coûteux et lent
- Quasi-impossibilité de faire oublier à un modèle quelque chose qu'on lui a appris (sauf fine-tuning encore plus poussé et très coûteux)
- Pas de contrôle d'accès : toutes les informations apprises par le modèle deviennent accessibles à l'ensemble des utilisateurs du modèle

LES PLUS

- Processus rapide (possibilité de mises à jour en continu : cf. les LLM qui s'appuient sur Google ou Bing pour ajouter du contenu d'actualité à leurs réponses)
- Accès aux documents et données de l'entreprise
- Possibilité d'un contrôle granulaire sur l'accès à l'information selon les rôles
- Nécessite un LLM générique léger

LES MOINS

- N'a pas d'impact sur la manière qu'aura le LLM de répondre : ce sujet devra être pris en compte dans le prompt (besoin de prompt engineering)

CONCLUSION

LE RAG POUR LE FOND, PUIS ÉVENTUELLEMENT LE FINE-TUNING POUR LA FORME

Le Fine-Tuning paraît séduisant, et il est poussé par les éditeurs, toutefois il s'avère complexe et dangereux à manipuler. S'il s'agit d'améliorer la pertinence des réponses, le RAG sera plus rapide et simple à mettre en œuvre et génèrera tout de suite de la valeur en complément de modèles légers, dont le rôle sera d'interagir en langage naturel. Le Fine-Tuning pourra intervenir plutôt sur la forme des interactions et des réponses (si une bibliothèque de prompts optimisés n'est pas suffisante).

COMPARER LES LLM : LES BENCHMARKS PUBLICS

Définition

Les modèles de langage sont évalués sur diverses tâches pour vérifier leur **efficacité** et **faciliter leur comparaison**. En effet, les LLM peuvent exécuter un large éventail de tâches, il est donc crucial d'évaluer leur performance en fonction des exigences particulières de nos applications.

Pour ce faire, des benchmarks open source permettent de tester et de comparer les performances des modèles. Nous avons classé ces benchmarks en plusieurs catégories, en fonction du **type de problème que le modèle doit résoudre**. Dans nos applications, nous avons généralement besoin d'une utilisation particulière des LLM (génération de code, recherche augmentée, etc.). En fonction de cela, nous pouvons observer les performances des modèles pour choisir celui qui sera le mieux adapté.

Notre œil d'expert : le Few-Shot Evaluation

Dans certains ensembles de données d'évaluation, les modèles ont été plus ou moins assistés dans la production de leurs réponses. C'est ce qu'on appelle le few-shot learning, une technique qui consiste à **fournir au modèle des exemples de questions et de réponses afin de l'orienter vers le type de réponse attendu.** Cela permet de guider le modèle vers la tâche qu'on souhaite qu'il accomplisse.

Par exemple, le benchmark **MMLU** a été évalué en mode 5-shot, ce qui signifie que le modèle a reçu 5 exemples de réponses avant d'être évalué. À privilégier donc pour tester la capacité d'un modèle à fournir le format de réponse attendu. À l'inverse, le benchmark **HumanEval** est en mode 0-shot, ce qui signifie que le modèle est simplement invité à résoudre le problème sans aucun exemple de solution. Ce test est rarement utilisé seul, car beaucoup plus versatile.

Différentes catégories de benchmark permettent d'identifier le meilleur LLM selon son usage

World of Knowledge

Les LLM peuvent être utilisés pour répondre à des questions sur des **connaissances générales ou sur des documents spécifiques.** Ces problèmes sont difficiles car les questions peuvent être complexes, les réponses implicites et les faits difficiles à identifier.
Pour cela, des benchmarks ont été créés, comme **TriviaQA (Joshi et al., 2017).** Ce benchmark regroupe un ensemble de questions permettant d'évaluer la capacité d'un modèle à répondre à des questions en langage naturel.

Génération de code

Les LLM peuvent être utilisés pour générer du code. Pour évaluer la pertinence de cette application, un ensemble de benchmarks a été créé. Le benchmark **HumanEval (Chen et al., 2021)** regroupe un ensemble d'exercices de génération de code. Pour chaque exercice, le but est de tester si le modèle est capable de générer du code qui passe les tests unitaires. Les exercices sont variés et couvrent un large éventail de tâches de développement.

Problèmes mathématiques

Les LLM ont fait des progrès significatifs dans de nombreuses tâches, mais ils ne sont pas encore capables de résoudre des problèmes mathématiques multi-étapes complexes. Des benchmarks comme **GSM8K (Cobbe et al., 2021)** spécifiques sont créés pour évaluer la pertinence des LLM sur des problèmes mathématiques simples. Ces benchmarks sont utiles pour évaluer la compréhension des modèles LLM et leurs capacités de “réflexion”.

Les benchmarks agrégés

Les LLM sont entrainés sur une quantité massive d'informations, y compris des sujets spécialisés. Cependant, il est encore difficile de savoir à quel point ces modèles sont capables d'apprendre et d'appliquer des connaissances de différents domaines. Pour les évaluer, des benchmarks généralisés ont été créés, comme le **MMLU (Hendrycks et al., 2020)**. Ce benchmark couvre 57 sujets, allant des sciences sociales à la philosophie en passant par les problèmes de physique.

TECHWAVES “GEN AI FOR CIO & CTO”

[sfeir]

WENVISION

LLM GÉNÉRALISTES

MAJ - R4

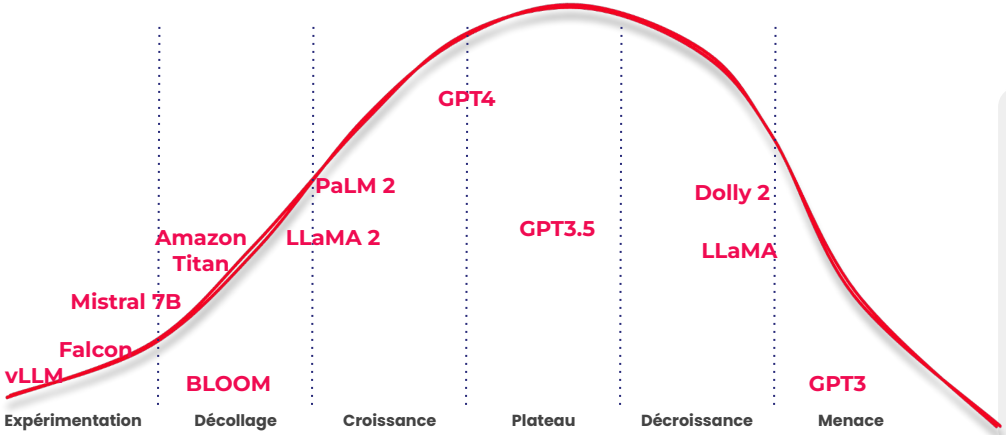
Définition

Un LLM (Large Language Model) est un type de modèle de machine learning qui a été entraîné sur un gros volume de données de type texte. Cela permet aux LLM de "comprendre" des contenus existants et de générer de nouveaux contenus, tels que des textes, du code, des scripts, des paroles de musique, des emails, des lettres, etc.

Les LLM sont encore en cours de développement, mais ils ont le potentiel de révolutionner un large éventail d'industries, en particulier sur les fonctions de développement logiciel, marketing, ventes, service client et développement de nouveaux produits.

Cas d'usage

Les LLM peuvent être utilisés dans divers domaines, tels que le traitement du langage naturel (classification de texte et l'analyse de sentiment), la traduction automatique, l'écriture créative et la réponse aux questions. Ils sont également utiles pour générer du code ou du texte, aider à l'analyse de données, détecter des fraudes et segmenter des clients.



Notre œil d'expert

Le domaine des LLM est en constante évolution, avec des leaders tels que GPT-4 d'OpenAI et PaLM 2 de Google, qui dominent actuellement le marché. Cependant, les modèles open source ont leurs propres avantages, notamment en matière de personnalisation et de déploiement *on premises*. Ces technologies évoluent rapidement, et il est crucial de rester à jour avec les dernières tendances et régulations, notamment l'AI Act de l'Union Européenne.

Il est essentiel de développer des solutions flexibles, permettant de remplacer facilement un LLM par un autre. Les LLM ne sont pas universels, il peut donc être judicieux d'utiliser différents modèles pour différentes tâches.

Pour tirer le meilleur parti des LLM, les entreprises doivent se tenir informées des meilleures pratiques et des avancées technologiques tout en respectant les régulations en vigueur. La clé du succès réside dans l'adaptabilité, la compatibilité, et le choix du modèle approprié pour chaque tâche spécifique.

Focus sur...

OpenAI

GPT-4 est le LLM d'OpenAI. Il s'agit du 4e modèle de la série GPT (Generative Pre-trained Transformer), publié en mars 2023. GPT-3 possédait 175 milliards de paramètres ; GPT-4 bien plus, certainement en additionnant plusieurs modèles. Il est multimodal, ce qui signifie qu'il peut traiter à la fois du texte et des images.

Google

Google PaLM 2 excelle dans la compréhension, la génération et la traduction de textes. Succédant au modèle PaLM 2022, il a démontré des résultats de pointe dans diverses tâches. Google prévoit d'utiliser PaLM 2 pour améliorer la recherche sur Google, créer des produits alimentés par l'IA et aider les utilisateurs dans leur travail.

TII Technology Innovation Institute

Falcon est un LLM avec 40 milliards de paramètres entraînés sur un trillion de tokens. Il s'agit d'un des LLM open source les plus puissants. Disponible gratuitement sous la licence Apache 2.0, Falcon s'est avéré performant sur des tâches, comme : génération de texte, traduction, réponse aux questions, résumé, génération de code.

LLM POUR LE CODE

Définition

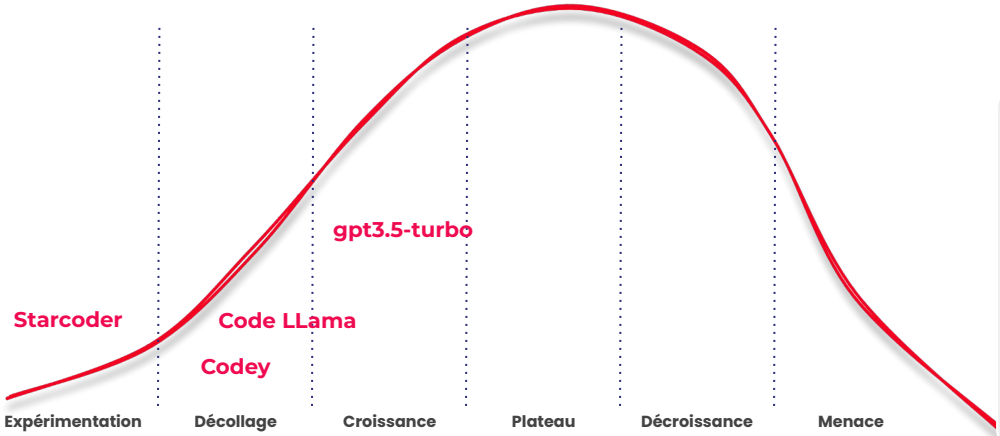
Les LLM spécialisés développeurs sont des LLM capables de générer du code. Ils sont entraînés sur d'immenses ensembles de données de texte et de code, et ils sont capables de comprendre et de générer du code dans différents langages de programmation.

Ces LLM sont en général des LLM généralistes qui sont ensuite customisés (Fine Tuning) pour les activités de développeurs.

Cas d'usage

Les LLM pour les développeurs sont des modèles qui peuvent accomplir diverses tâches telles que la génération de code, l'explication du code, la traduction de langages, le débogage, la restructuration du code, la génération de tests et la révision du code.

Dans la grande majorité des cas ,ces modèles sont utilisés à travers des API et viennent enrichir des outils ou IDE existants.



Notre œil d'expert

Ces modèles améliorent la productivité en générant des extraits de code, en proposant des suggestions et en facilitant les tâches de développement. Les LLM spécialisés sont en cours d'intégration avec succès aux outils et plateformes de développement les plus courants, ce qui les rend accessibles et largement adoptables par les développeurs.

Ces LLM spécialisés vont transformer la manière dont les développeurs abordent le développement de logiciels, ceci se traduira par un gain de temps considérable et une amélioration de la productivité.

Toutes les entreprises doivent se préparer à intégrer des coûts de licence pour l'usage de ces outils.

A noter que les licences d'utilisation de ces outils protègent les entreprises utilisatrices contre la récupération de leurs données par les fournisseurs de services. Ceci étant dit, une solution comme Starcoder pourra satisfaire les entreprises les plus prudentes.

Focus sur...

OpenAI

gpt3.5-turbo est le LLM d'OpenAI qui traduit du langage naturel en code. gpt3.5-turbo est au cœur de GitHub Copilot et supporte une douzaine de langages de programmation. Python est son langage le plus performant.
OpenAI propose aussi une API qui permet d'utiliser ce LLM en se passant de son intégration dans un IDE.

Google

Google Codey est basé sur son LLM PaLM 2. Codey peut générer du code à partir de descriptions en langage naturel, compléter des extraits de code et fournir des suggestions. Il peut être utilisé pour déboguer le code et identifier les erreurs potentielles. Accessible via une API, il fait aussi partie de la base de Replit Ghostwriter.

starcoder

Hugging Face et ServiceNow ont collaboré pour créer StarCoder, LLM open source pour le code, dans le cadre de l'initiative BigCode. Entraîné sur plus de 1 000 milliards de tokens et prenant en charge plus de 80 langages, StarCoder compte 15,5 milliards de paramètres et affiche des performances égales ou supérieures à celles de modèles fermés.

ASSISTANTS DE PROGRAMMATION

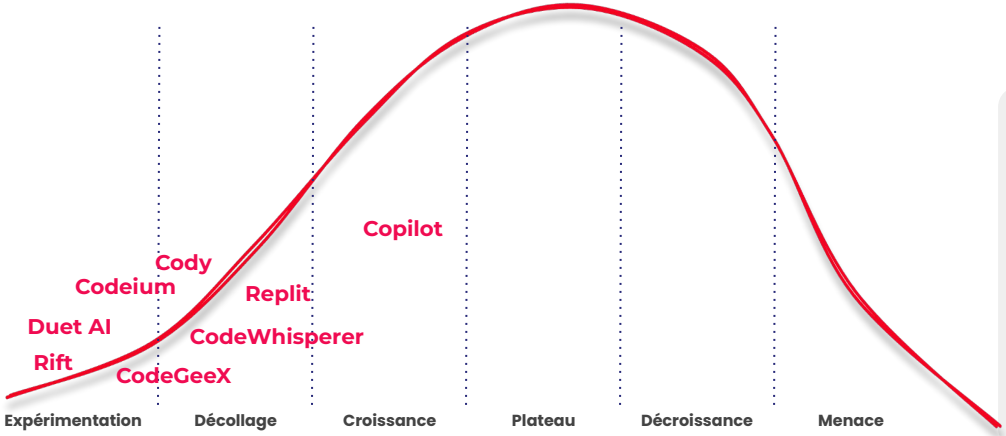
MAJ - R4

Définition

Les assistants de programmation alimentés par IA peuvent aider à coder plus rapidement et de manière plus efficace. Ils fonctionnent en utilisant un modèle linguistique volumineux entraîné sur des millions de lignes de code disponibles en Open Source. Cela leur permet de faire des suggestions en fonction de ce que vous avez déjà tapé, et de générer du code à la fois correct et idiomatique.

Cas d'usage

Ces outils sont d'une grande aide pour les développeurs. Ils offrent de nombreux cas d'utilisation : écriture de nouvelles fonctionnalités, refactorisation du code, tests et apprentissage de nouveaux langages. Ils peuvent également aider les non-anglophones à écrire du code en anglais, tester du code et expliquer du code, générer des expressions régulières. Enfin, ces outils peuvent être utilisés comme des tuteurs pour progresser.



Notre œil d'expert

Le domaine des outils de développement assisté par une IA générative est actuellement le plus avancé en termes de maturité. Ces outils ont été introduits très tôt et ont été très bien accueillis par la communauté des développeurs, qui les trouve extrêmement utiles pour améliorer leur productivité.

GitHub Copilot a été le premier outil à arriver sur le marché et a rapidement pris une avance considérable. Cependant, le marché évolue rapidement et des solutions telles que Replit, qui propose un environnement entièrement en ligne, se positionnent comme de bons concurrents.

Des outils plus spécialisés visent des tâches spécifiques pour améliorer la productivité dans des domaines précis. Code Interpreter, une fonctionnalité (en bêta) de ChatGPT, renforce par exemple ses capacités, notamment en matière d'analyse et de préparation de données. Chez AWS, CodeWhisperer vise en premier lieu à faciliter l'utilisation des produits AWS.

Le déploiement de ce type d'outil à grande échelle constitue un bon test pour évaluer l'adoption de l'IA générative au sein d'une entreprise et confirmer qu'il n'y a pas d'obstacles majeurs à son utilisation.

Focus sur...

GitHub

Github Copilot est le leader du marché. Il a été le premier outil à utiliser l'intelligence artificielle pour aider les développeurs. Son intégration transparente avec les IDE populaires permet une expérience fluide. De plus, sa bonne intégration avec le leader des gestionnaires de code, Github, facilite la collaboration et la gestion du code source.

replit

Replit Ghostwriter est une bonne solution pour le développement en ligne, il ne nécessite aucune installation, il est toujours à jour et propose des tarifs intéressants. Il est possible de l'utiliser n'importe où, sans avoir à se soucier de l'installation de logiciels ou de la mise à jour de Ghostwriter.

amazon web services

Amazon CodeWhisperer se concentre principalement sur les cas d'utilisation liés au développement sur AWS, mais il peut également être utilisé pour écrire des applications indépendantes. Dans la plupart des cas, le code écrit avec CodeWhisperer peut fonctionner partout.

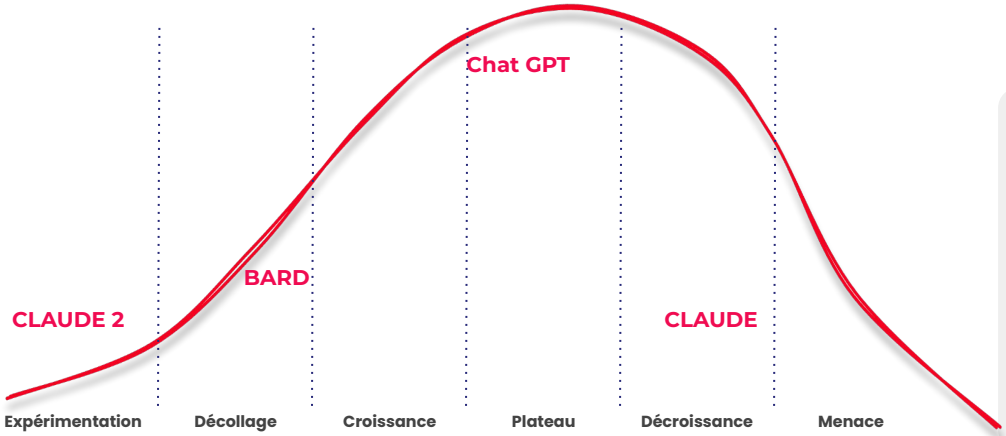
LES AGENTS CONVERSATIONNELS

Définition

Les systèmes de Chat basés sur les LLM sont les plus connus du grand public. Ils illustrent la capacité des modèles de langage à s'adapter à des domaines spécifiques et à fournir des réponses pertinentes et spécialisées. Ils sont le fruit d'un entraînement sur de vastes ensembles de données et d'une architecture avancée de *Transformer*, qui leur permet d'apprendre et de générer du texte de manière contextuelle et cohérente.

Cas d'usage

Ces IA conversationnelles sont extrêmement utiles pour les développeurs, que ce soit pour résoudre des problèmes spécifiques ou trouver des idées liées à la programmation. Ils offrent une grande aide en permettant de rester à jour avec les nouvelles technologies et frameworks, ainsi qu'en améliorant les compétences grâce à des exercices d'algorithmique ou des katas. Certains les considèrent comme une évolution plus avancée de Stack Overflow.



Notre œil d'expert

Ces outils sont largement connus et utilisés par de nombreuses personnes de nos jours. Cependant, dans le milieu professionnel, il n'est pas recommandé de les utiliser. Les licences d'utilisation et la récupération des données pour l'entraînement des modèles posent des problèmes lorsqu'il s'agit de les utiliser à des fins professionnelles. Nous pouvons les recommander à titre individuel pour la veille technologique, l'idéation ou encore pour résoudre des problèmes sur des projets personnels.

Parmi les principaux outils, nous avons ChatGPT, BARD et CLAUDE. **ChatGPT** est le plus populaire et permet de dialoguer facilement avec un agent gratuitement, fournissant des réponses pertinentes, notamment pour l'idéation. Cependant, en ce qui concerne la veille technologique, il est en retard car il ne dispose pas de connaissances postérieures à septembre 2021. **BARD**, quant à lui, est la version grand public de PaLM 2, très performant dans de nombreuses tâches, mais il n'est pas encore disponible en France.

Enfin, **Claude** a été entraîné en partenariat avec des sites comme Quora ou Notion. Il est encore en phase exploratoire mais peut apporter un contrepoids intéressant par rapport aux autres acteurs.

Focus sur...

OpenAI

ChatGPT est un agent conversationnel développé par OpenAI. D'abord basé sur le LLM GPT-3, il s'appuie aujourd'hui sur GPT-4 dans sa version payante, ChatGPTPlus, Le modèle a été entraîné sur une vaste quantité de données provenant d'Internet, lui permettant d'acquérir une compréhension approfondie du langage naturel.

Google

Bard est un chatbot d'IA conversationnelle générative développé par Google. Il s'appuie sur la famille de modèles de langage PaLM. Bard a été créé en réponse directe à l'émergence du ChatGPT d'OpenAI, avec la capacité d'analyser des URL. Il a été lancé en mars 2023 dans certaines parties du monde ; il est en train d'être déployé globalement.

ANTHROP\C

Claude 2, version du 11 juillet 2023, est un agent qui excelle dans la génération de texte, l'écriture de code et l'assistance aux utilisateurs.
Développée par Anthropic, start-up créée par des anciens d'OpenAI, cette technologie est guidée par un ensemble de principes éthiques qui différencient Claude des autres chatbots IA sur le marché.

ORCHESTRATEURS

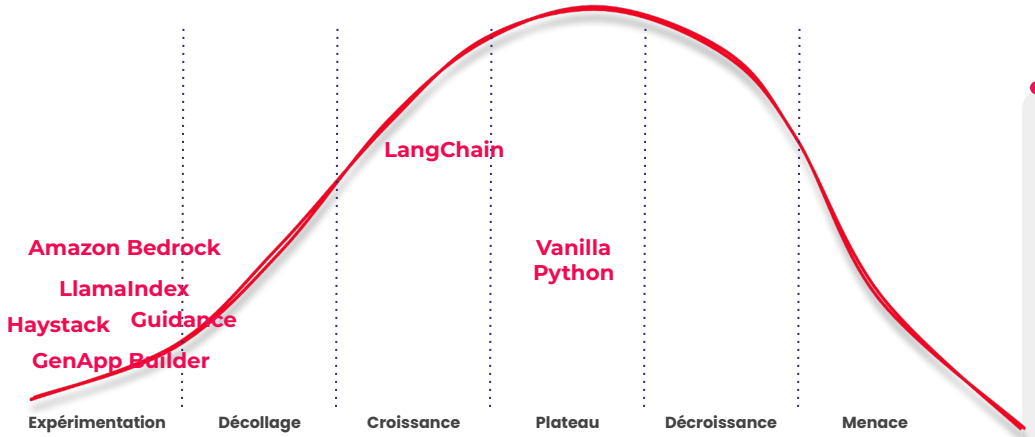
MAJ - R4

Définition

Les orchestrateurs LLM sont conçus pour faciliter l'interaction entre les utilisateurs et les LLM en gérant des tâches complexes qui nécessitent généralement plusieurs appels d'API. Ils aident à structurer les conversations, à gérer l'état du dialogue, à gérer les entrées et sorties et à contrôler les échanges avec le modèle.
Ces orchestrateurs sont des outils précieux pour exploiter pleinement les capacités des LLM, en facilitant l'interaction avec les modèles et en offrant des fonctionnalités supplémentaires pour une utilisation plus efficace et adaptée aux besoins spécifiques des utilisateurs.

Cas d'usage

Ces outils peuvent être utilisés pour construire des chatbots capables de générer du texte, de traduire des langues, de répondre à des questions... Ils permettent également d'automatiser des tâches actuellement effectuées manuellement et d'intégrer des LLM avec des bases de données vectorielles pour créer des applications plus puissantes. Les orchestrateurs permettent in fine de créer des agents autonomes, capables de planifier des actions et de les effectuer.



Notre œil d'expert

Ces outils ne sont disponibles que depuis peu de temps. Ils sont très vite devenus indispensables pour construire des applications basées sur les LLM. Ils permettent non seulement d'accélérer le développement, mais également d'ajouter des fonctionnalités supplémentaires aux LLM, ce qui les rend incontournables lors de la création d'un outil. Que ce soit pour la gestion des prompts, l'optimisation du contexte, le chaînage des appels API ou encore la vectorisation des documents en vue de leur intégration dans le modèle, les orchestrateurs simplifient et accélèrent l'intégration de ces fonctionnalités.

L'un des orchestrateurs les plus populaires est Langchain, une initiative open source qui a connu une progression remarquable en passant de 0 à plus de 49 000 étoiles sur GitHub en seulement quelques mois. Ce framework permet de se connecter aux modèles les plus couramment utilisés et de réaliser des chaînages d'appels avec des API externes. Grâce à une communauté active et engagée, de nouveaux composants sont régulièrement ajoutés, enrichissant ainsi les fonctionnalités de l'outil de manière quasi quotidienne.

Concernant les hyperscalers, Microsoft a sorti Guidance en open source. Bien que moins utilisé que Langchain, il n'en reste pas moins un solide concurrent dans l'avenir.

Focus sur...

LangChain

LangChain est un framework open source de création d'applications utilisant des LLM. Il offre une API de haut niveau, prend en charge plusieurs LLM pour faciliter le développement d'applications telles que des chatbots, des outils de synthèse de documents, des outils d'analyse de code, des assistants personnels....

Microsoft

Microsoft Guidance est un cadre ouvert pour contrôler les LLM, avec une syntaxe simple. Il prend en charge diverses structures de sortie, le prototypage et la mise en cache. Les applications potentielles incluent la génération de texte, la traduction, et l'interaction utilisateur dans les chatbots.

Google

Gen App Builder de Google Cloud permet aux développeurs de créer rapidement des applications d'IA générative d'entreprise en utilisant des modèles, des outils de recherche et des technologies d'IA conversationnelle. Elle permet de créer des chatbots, des assistants numériques, etc. en quelques minutes ou heures.

GUI POUR LES LLM

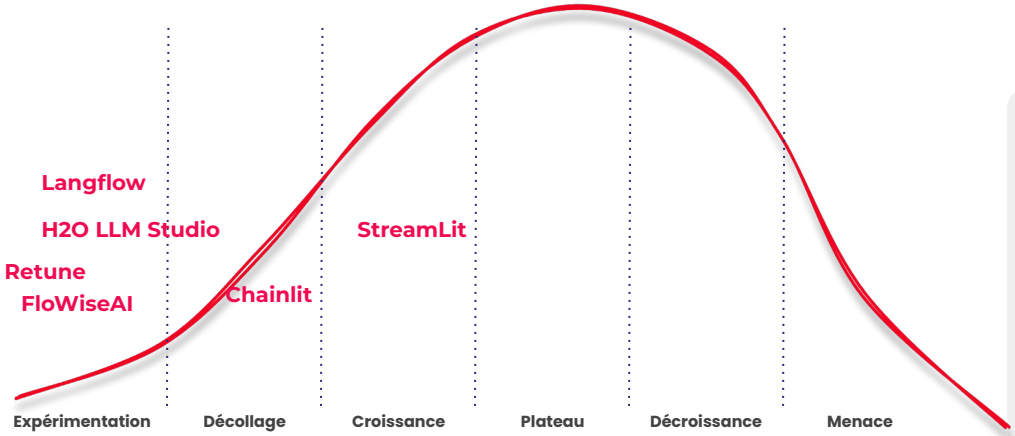
Définition

Les outils d'interface utilisateur graphique (UI) sont bien connus dans l'écosystème de l'informatique. Que ce soit pour les applications mobiles, les sites web ou les clients lourds, nous avons toujours accès à des interfaces utilisateur. Ce qui change aujourd'hui, c'est que nous pouvons nous appuyer sur des frameworks existants pour offrir des fonctionnalités améliorées grâce aux LLM.

Le poids de l'UI dans l'adoption d'un outil a été démontré par ChatGPT, qui a réussi le tour de force de la démocratisation de l'IA Générative.

Cas d'usage

Une part essentielle dans la création d'applications basées sur les LLM réside dans l'interface utilisateur (UI) qui permet aux utilisateurs d'interagir facilement avec ces modèles. Afin de simplifier cette intégration, il existe des frameworks spécifiquement conçus pour aider à formaliser celles- ci et accélérer le déploiement.



Notre œil d'expert

Il existe aujourd'hui de nombreux frameworks disponibles dans le domaine de l'interface utilisateur (UI). Pour notre part, nous avons choisi de nous concentrer sur des outils nouveaux qui s'intègrent facilement avec les LLM.

Notre premier choix se porte sur Streamlit, un framework créé en 2018 qui a connu une adoption importante auprès des data scientists et des analystes. Son avantage réside dans le fait qu'il est écrit en Python, ce qui en fait un outil idéal pour l'intégration avec les systèmes de données.

Chainlit, également en Python, permet de construire des expériences utilisateur orientées LLM. Encore jeune, ce framework doit faire sa place dans un écosystème bien rempli.

Le dernier choix est Flowise, une initiative différente des deux précédentes, car il s'agit d'une surcouche directe de LangChain. Ce framework permet de créer des applications en mode No Code / Low Code, facilitant l'interconnexion de différentes boîtes à outils.

Focus sur...

Chainlit

Chainlit est une librairie Python open source conçue pour construire et partager rapidement des applications LLM, principalement sous la forme de Chat.
Cette librairie est facile d'utilisation, elle propose nativement une compatibilité avec de nombreux LLM.

Streamlit

Streamlit a été créé en 2018. À l'origine, il a été conçu pour être un outil permettant aux data scientists de créer des tableaux de bord interactifs pour leur travail. Cependant, Streamlit a rapidement gagné en popularité auprès d'autres développeurs, et il est maintenant utilisé par un large éventail de personnes.

FlowiseAI

Flowise est une initiative open source permettant de créer des applications LLM via une interface graphique.
Basée sur le framework Langchain, Flowise apporte une surcouche UX/UI permettant la création d'applications sans avoir à coder.

PROMPT ENGINEERING

Définition

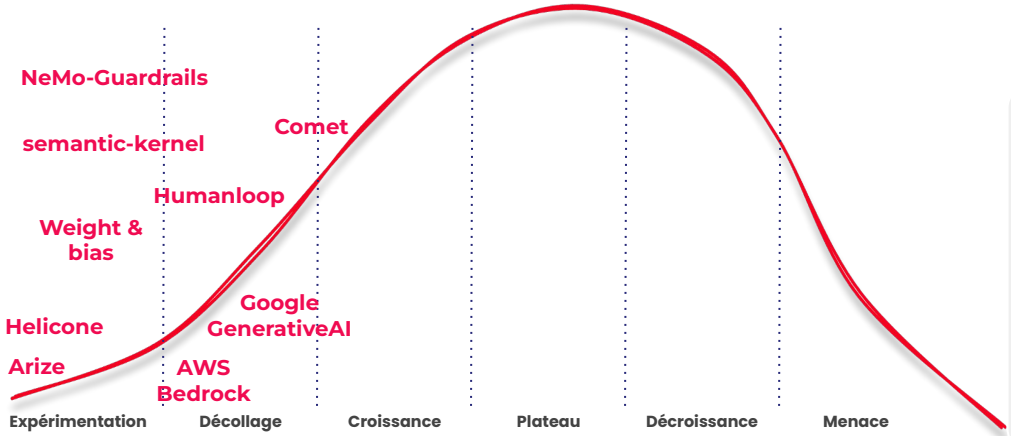
Le "prompt engineering" est le processus consistant à formuler et à structurer de manière stratégique les instructions et les requêtes données à un modèle tel qu'un LLM. L'objectif du prompt engineering est d'optimiser les performances et les résultats du modèle en ajustant les formulations des prompts pour obtenir les réponses, les comportements ou les sorties souhaitées.

Il existe plusieurs cas d'utilisation du *prompt engineering*, mais nous nous concentrerons sur deux domaines spécifiques : la conception des prompts et le suivi des utilisations.

Cas d'usage

Dans la phase de conception, les outils offrent la possibilité de créer des couches supplémentaires de prompts et de les versionner. Cette approche simplifie le processus de conception et de gestion des prompts utilisés avec les modèles de langage (LLM).

Pour l'analyse, il existe des outils dédiés à la visualisation et à l'analyse des usages effectués par les utilisateurs sur les LLM. Ces outils fournissent des moyens de visualiser les schémas d'utilisation des prompts, ce qui offre des informations précieuses pour l'optimisation et l'amélioration des performances des systèmes.



Notre œil d'expert

L'utilisation de stratégies de *prompt engineering* devient de plus en plus complexe et essentielle en tant que moyen de différenciation des produits. De nombreux développeurs commencent de nouveaux projets en expérimentant avec des prompts simples, tels que des instructions directes (*zero-shot prompting*) ou quelques exemples de résultats (*few-shot prompting*). Bien que ces prompts donnent souvent des résultats satisfaisants, ils ne permettent généralement pas d'atteindre les niveaux de précision nécessaires pour les déploiements en production. Il existe plusieurs outils dans ce domaine, la plupart payants, mais certains proposent également des niveaux gratuits pour tester leurs solutions. Cependant, nous ne savons pas encore si ces outils seront durables à long terme ou s'ils seront intégrés à des solutions plus complètes, comme les orchestrateurs.

Des outils de visualisation des prompts tels que **Humanloop** ou **Helicone** fournissent une interface supplémentaire entre les interactions utilisateur et le modèle, permettant de suivre les prompts et d'observer les schémas de comportement. Certains hyperscalers, tels que **Google** ou **Amazon**, offrent la possibilité de *fine-tuner* automatiquement les prompts en entraînant une surcouche sur le modèle.

Focus sur...

aws

Amazon Bedrock est un service serverless qui permet d'accéder à une variété de modèles de fondation provenant d'Amazon et des meilleures start-ups. Bedrock permet de démarrer rapidement et de personnaliser les modèles de fondateurs en toute confidentialité. Les modèles sont ensuite déployés à travers des APIs gérées par AWS.

GOOGLE GENERATIVE AI

Google Generative AI offre une interface prête à l'emploi qui permet de *fine-tuner* les modèles de langage en utilisant des exemples de prompts et en les personnalisant. La plateforme permet également de ré-entraîner une surcouche du modèle, offrant ainsi une approche guidée par le ML pour le *prompt engineering*.

Humanloop

Humanloop est un outil puissant qui assiste les développeurs dans l'utilisation efficace des LLM. Il facilite l'expérimentation de nouveaux prompts, la collecte de données et de commentaires des utilisateurs, et l'optimisation des modèles pour des performances améliorées tout en tenant compte des contraintes de coûts.

BASES DE DONNÉES VECTORIELLES

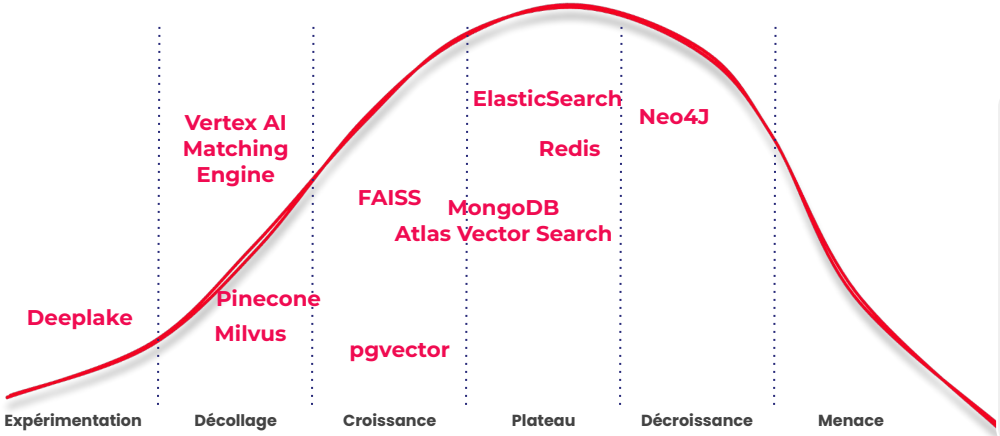
Définition

Les bases de données vectorielles sont principalement de type NoSQL qui stockent et gèrent des “embeddings” (vecteurs). Ces vecteurs sont une méthode de représentation numérique de texte ou d'autres données qui peut être utilisée pour la recherche par similarité dans un espace multidimensionnel (deux concepts similaires auront des vecteurs proches).

Les bases de données vectorielles sont conçues pour stocker et récupérer efficacement des vecteurs de grande dimension, en utilisant des techniques d'optimisation telles que le hachage, l'approximation et la recherche basée sur les graphes.

Cas d’usage

Nous avons choisi de mentionner les bases vectorielles ici car leur utilisation peut stimuler les performances des LLM. En combinant une base vectorielle avec un LLM, il est possible de créer un système de recherche textuelle personnalisé, performant et efficace, notamment pour des documents PDF.
Cette approche permet d'améliorer la précision et la pertinence des résultats de recherche, en exploitant les caractéristiques vectorielles des documents et en utilisant les capacités de génération et de compréhension de texte des LLM.



Notre œil d’expert

Vertex AI Matching Engine, Pinecone, Weaviate, Milvus et Faiss sont quelques-unes des principales bases de données vectorielles qui redessinent le paysage de l'indexation des données et de la recherche par similarité.

Vertex AI Matching Engine permet d’avoir une base serverless et gérée par l’infrastructure Google Cloud, ce qui simplifie le déploiement et la maintenance.

Faiss s'est imposé comme une excellente option pour la recherche de similarités à haute performance. Milvus gagne rapidement en popularité grâce à ses capacités d'indexation et d'interrogation évolutives.

Des bases plus anciennes comme Redis, Postgre (extension pgvector) ou Neo4J peuvent également servir à stocker et rechercher des vecteurs.

Focus sur...

Meta

FAISS (Facebook AI Similarity Search) est une bibliothèque open source développée par Facebook AI Research qui offre des fonctionnalités de recherche de similarité rapide et évolutive. Elle est principalement utilisée pour la recherche de vecteurs similaires dans de grandes bases de données.

Google

Vertex AI Matching Engine simplifie la comparaison de similitudes basée sur des vecteurs. Il permet le stockage et l'interrogation de représentations vectorielles de données. Cet outil serverless permet aux développeurs d'effectuer des recherches de similarité de manière efficace dans divers types de données.

milvus

Milvus est une base de données vectorielles open source performante, capable de gérer des volumes massifs de données vectorielles. Son support pour différents index de recherche vectorielle et son filtrage intégré en font un choix populaire pour les applications nécessitant des opérations de recherche efficaces sur des données vectorielles.

LES EMBEDDINGS

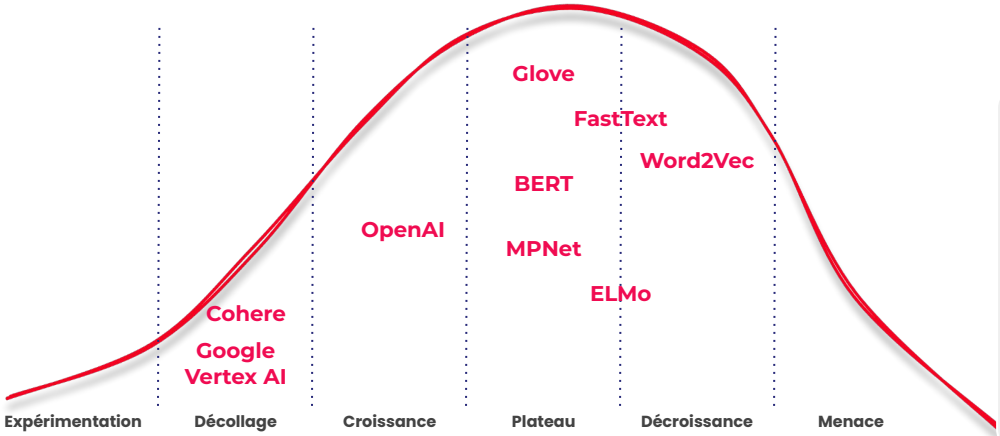
MAJ - R4

Définition

L'embedding consiste à exprimer des mots ou des phrases en représentation numérique (un vecteur). Cette représentation permet de capturer les liens sémantiques et le sens sous-jacent entre les mots d'un corpus de texte. L'idée est de regrouper dans une représentation numérique les mots apparentés. Les embeddings sont générés par des modèles de langage spécialisés conçus pour cette tâche.

Cas d'usage

C'est un élément critique d'une stack pour une app basée sur le LLM. D'une part, il permet la conversion des documents bruts pour qu'il puissent être stockés dans une base vectorielle. Et d'autre part, il permet de contextualiser la requête d'un utilisateur. En effet, l'embedding va convertir cette requête en espace vectoriel et permettre de chercher dans la base de données vectorielle les documents ayant le plus de liens sémantiques avec la question.



Notre œil d'expert

L'embedding n'est pas propre au LLM, Google utilise cette technique pour optimiser les résultats de recherche depuis déjà une décennie (Word2Vec). Le choix du modèle d'embedding dépendra principalement de critères techniques tels que la précision, la vitesse, l'utilisation du stockage et de la mémoire, parmi d'autres.

Hugging Face donne accès très facilement à une grande quantité de modèles d'embedding open source. Ces derniers peuvent être utilisés directement dans la stack sans faire appel à un service externe. D'importants acteurs offrent des embeddings via des API, notamment Google, OpenAI, Cohere, ou Hugging Face qui facilite le déploiement des embeddings open source.

Le choix entre un gros acteur et une solution open source pour l'embedding dépend de l'expertise disponible, des coûts, de la sécurité et du contrôle des données. Les solutions open source nécessitent une gestion interne du modèle, engendrant des coûts humains et des besoins d'infrastructure, tandis que les gros acteurs offrent une gestion externalisée avec un modèle économique basé sur l'usage. Toutefois, cela implique le partage de données sensibles avec ces acteurs, ce qui requiert un cadre contractuel robuste.

Focus sur...

OpenAI

Lorsqu'en décembre 2022, OpenAI sort son nouveau modèle, text-embedding-ada-002, il remplace cinq modèles distincts. Ce nouveau modèle surpasse leur précédent modèle le plus performant, Davinci, sur la plupart des tâches, tout en étant proposé à un prix inférieur de 99,8%.

Google

Le service d'embedding de Vertex AI est encore très récent mais donne accès à des modèles d'embedding pour du texte et pour des images. Ce dernier génère des vecteurs pouvant être une combinaison de données d'image et/ou de texte.

Hugging Face

Hugging Face propose plus de 500 modèles d'embedding sur son Hub. Leur plateforme offre une grande variété de choix pour des cas d'usage spécialisé, notamment en fonction des langues. De plus, leur service de déploiement de modèle s'étoffe de nouvelles fonctionnalités régulièrement. Cela rend la mise en production de modèle de plus en plus accessible.

UN GRAND MERCI !

Les TechWaves sont un exercice collectif. Nous adressons tous nos remerciements aux équipes de SFEIR et WENVISION qui ont contribué à les mettre en musique.

Un merci tout particulier à Adrien Lasselle, Arnaud Domard, Bertrand Mondolot, Julia Wabant, Michaël Sherding, Pierre-Alexandre Picard, Romain Viau, Salim Elakoui, Vincent Matthys.

WENVISION

TECHWAVES - #GENAI FOR DEV

www.sfeir.com

www.wenvision.com

Techwaves GenAI

L'analyse des experts de SFEIR et WEnvision sur les solutions de GenAI

Vous êtes perdus et ne savez par où commencer sur le sujet GenAI ?
Les équipes de SFEIR et WEnvision ont compilé les solutions du marché pour vous offrir un condensé et un classement sur une courbe de tendances technologiques des solutions de GenAI.

Les Techwaves sont mises à jour toutes les 6 semaines.

A retrouver sur :
consulting.wenvision.com et sfeir.com