camp
tocamp

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

Les Grands Modèles de Langage pour le Géospatial

Contexte et État de l'art

Camptocamp

1. Préambule

Dans le cadre du Plan d'action 2024 de la Stratégie suisse pour la géoinformation, **Swisstopo** s'interroge sur le potentiel de l'IA générative et des grands modèles de langage pour faciliter l'usage de son infrastructure de géodonnées.

Swisstopo souhaite notamment préciser les points suivants:

- *Comment les LLM peuvent-ils être utilisés pour améliorer l'expérience d'utilisateurs non experts cherchant des réponses à des questions liées au territoire.*
- *Dans quelle mesure ces LLM peuvent être entraînés à répondre de façon fiable à des questions dont la réponse se trouve dans une infrastructure de géodonnées.*

Swisstopo mandate Camptocamp pour l'accompagner dans cette démarche et demande leur expertise sur 3 éléments:

1. Réalisation d'un état de l'art de l'utilisation des LLM pour le géospatial.
2. Mise en évidence du potentiel de l'IA générative pour l'infrastructure nationale de géodonnées.
3. Développement d'un Proof of Concept qui démontre pratiquement certains concepts du point 2 et se concentre sur la recherche et l'exploitation des jeux de données de la plateforme nationale suisse de géodonnées.

Ce document répond au premier besoin, il définit les concepts importants et dresse l'état de l'art de l'intelligence artificielle et plus spécifiquement des grands modèles de langage ("Large Language Models" ou LLM) appliqués au monde du géospatial.

Le terme géospatial exclut ici le domaine du traitement d'image et des modèles de Deep Learning pour la segmentation et la classification d'objets à partir d'images satellites ou aériennes (computer vision). L'étude se concentre sur l'utilisation des LLM sur des données structurées : données géographiques vectorielles et métadonnées (décrivant des services et des jeux de données vectorielles et raster).

2. Introduction

2.1. La révolution LLM

L'apparition des grands modèles de langage et des techniques d'IA générative bouleverse le panorama de la technologie informatique, et ouvre une voie encore inexplorée dans le champ des possibles. L'intelligence artificielle est omniprésente dans les discussions, chaque jour voit son lot de nouveaux modèles, applications ou produits basés sur l'IA. L'adoption de l'IA générative s'étend à un nombre croissant de secteurs, cherchant à s'intégrer dans tous les aspects de la société.

Le monde du géospatial n'échappe pas à cette règle, l'émergence des LLM ouvre de nouvelles perspectives pour la recherche, l'analyse et le traitement des données géographiques vectorielles, facilitant ainsi une meilleure compréhension des territoires et l'aide à la prise de décision.

2.2. Origines et développement des LLM

1990s

Les premières applications de réseaux de neurones à convolution (CNN) sont mises en production. Le CNN est un sous-type de réseau de neurones artificiels (ANN), qui ont introduit les méthodes d'apprentissage. Les CNN sont optimisés pour la reconnaissance d'image ou de vidéo. La puissance de calcul limitée de l'époque bride le déploiement de ces systèmes à grande échelle.

Pour en savoir plus:

- L'article Wikipedia *History of artificial neural networks* donne un bon historique des ANNs.

2010s

Les calculs sont déportés du CPU (Central Processing Unit) vers le GPU des cartes graphiques (Graphic Processing Unit). Ces chipsets offrent de nouvelles possibilités pour l'entraînement des CNN, ce qui fait décoller cette technique dans le domaine de la reconnaissance d'image (segmentation, classification etc...).

2017

Google publie un papier intitulé "Attention Is All You Need" qui a un impact majeur sur le domaine des ANN en introduisant l'architecture des Transformer, une architecture particulièrement efficace pour traiter le texte. Ce seront les bases des grands modèles de langage.

Pour en savoir plus:

- ➔ L'architecture des Transformer est décrite et illustrée dans le blog The illustrated Transformer.
- ➔ Le papier légendaire Attention Is All You Need qui a défini les Transformer.

2020

OpenAI sort GPT-3, qui devient le plus gros modèle de langage, avec 175 milliards de paramètres et démontre des capacités stupéfiantes sur le traitement du langage naturel.

2022

OpenAI sort ChatGPT qui connait rapidement un succès mondial et fait découvrir les modèles de langage au grand public. La capacité de dialoguer de manière naturelle avec ce service fascine le public, faisant de ChatGPT le produit avec le succès le plus fulgurant de l'histoire d'internet.

Depuis 2023

L'IA se démocratise et voit apparaître de nombreux modèles open source rivalisant avec GPT-3 comme Llama de Meta. L'écosystème foisonnant de Huggin Face alimente cette effervescence, les modèles se multiplient à grande vitesse.

Aujourd'hui, l'actualité IA est en pleine effervescence, les modèles d'IA générative se développent constamment, intégrant de manière de plus en plus naturelle l'audio et la vidéo.

3. Comprendre les Grands Modèles de Langage

3.1. Définition des grands modèles de langage (LLM)

Les grands modèles de langages sont des systèmes d'Intelligence artificielle destinés à

- **Comprendre du texte**: Traiter et analyser des grands volumes de données sous forme de langage naturel
- **Générer du texte:** Générer des réponses aux prompts des utilisateurs, à partir du texte donné en contexte.

Ces systèmes sont entraînés sur de grosses quantités de données via des algorithmes d'apprentissage automatique (Machine Learning) perfectionnés dans la compréhension de la structure du langage humain. Ils sont capables de comprendre le langage, mais également d'en générer. Dès lors, ces modèles deviennent incontournables pour les applications de type traitement du langage naturel, traductions automatisées, génération de code et de texte etc...

3.2. Comment fonctionnent les LLM ?

Un LLM est un programme informatique défini par une structure (une architecture) et par un grand nombre de paramètres. Il fonctionne de la façon suivante:

- **Entraînement:** Dans un premier temps, il ingère et analyse de gros volumes de données pour déterminer les poids du modèle. Ces poids modélisent les données sur lesquelles le modèle est entraîné.
- **Utilisation (inférence):** Une fois l'architecture interne (structure + poids) développée, le modèle prend en entrée du texte, l'analyse, et génère la suite - la réponse la plus probable au problème posé - en se basant sur le texte sur lequel il a été entraîné.

L'architecture originale des Transformers est composée de deux parties: un *encodeur* qui transforme le texte d'entrée en une représentation interne et un *décodeur* qui génère du texte à partir de la représentation interne de l'entrée.

Pour en savoir plus:

- Understanding Encoder And Decoder LLM
- Mixture of Expert Explained

3.3. Entraîner un LLM

Un LLM fonctionne de par ses paramètres et sa structure. Un grand modèle a en général plusieurs milliards de paramètres, qui sont encodés en nombres décimaux. La valeur de ces paramètres (les poids) détermine la façon dont le LLM va répondre à la question posée.

La phase d'entraînement consiste à déterminer l'ensemble des poids du modèle. C'est un processus itératif qui consiste à

1. Montrer au modèle le début d'un texte;
2. Lui demander de prévoir le mot suivant;
3. Modifier légèrement ses paramètres pour que sa réponse soit plus proche du mot attendu.

La base de connaissances utilisée pour l'entraînement détermine les réponses fournies par le LLM, sa "logique" et ses biais.

Entraîner de bout en bout un LLM est une tâche qui demande une puissance de calcul colossale et a un coût faramineux. C'est donc une tâche réservée à quelques entités qui ont les moyens de l'accomplir.

Pour réduire ces coûts, les opérateurs tentent de réduire le nombre de paramètres, ou la représentation de ces paramètres sans trop dégrader la performance.

3.4. À quoi les LLM sont-ils utilisés ?

Les LLM sont utilisés pour leur capacité à comprendre et générer du texte, tâche qui a toujours été très difficile à résoudre par des programmes informatiques.

Voici quelques exemples de cas d'usage généraux des LLM

Chatbot et assistants

L'un des usages les plus courants est de venir aider l'utilisateur avec un assistant, capable de tenir une conversation en langage naturel pour résoudre votre problème.

Classification de texte

La capacité à catégoriser/étiqueter de gros volumes de données permet d'identifier dans un document une trame commune, une tendance, pour appuyer la prise de décision par exemple. L'analyse de sentiment est une forme de classification, les LLM peuvent aider à percevoir le ton, l'attention, les émotions et les opinions.

Traduction

Les LLM sont souvent entraînés dans plusieurs langues si bien qu'ils sont très performants pour traduire de grandes quantités de textes de façon qualitative.

Génération de code et de texte

Les LLM peuvent être entraînés sur des énormes bases de code et deviennent alors d'excellents assistants pour le développement logiciel. Ils peuvent aussi être entrainés sur de la documentation pour aider les tâches de développement.

Résumé

Résumer, synthétiser un texte ou un document.

Question Answering

Répondre aux questions posées. Les réponses dépendent de la nature du modèle et de la base de connaissances sur laquelle il a été entraîné.

Feature Extraction

Extraire une information spécifique d'un texte, par exemple "extraire un lieu d'une question"

Point important: la base de connaissance d'un LLM est encodée dans les poids du modèle. Ces poids sont rarement ré-évalués, à cause du coût d'entraînement, si bien que les LLM n'ont pas de connaissance à jour et ne peuvent pas résoudre des problèmes liés à l'actualité.

En général, on conseille de ne pas se servir des LLM comme base de connaissance mais pour leur capacité à raisonner, comprendre et générer du texte.

3.5. Diversité dans les modèles de langages

Bien que les Grands Modèles de Langage soient les plus médiatisés en raison de leurs capacités impressionnantes, ils ne représentent qu'une partie de l'écosystème des modèles de langage. D'autres modèles, plus spécialisés, jouent également un rôle crucial dans l'écosystème de l'IA générative textuelle:

- Les **modèles d'embedding**, par exemple, sont utilisés pour représenter le texte sous forme vectorielle dans des espaces sémantiques, facilitant des tâches comme la recherche ou le clustering.
- Les modèles **text-to-SQL** permettent de traduire des requêtes en langage naturel en instructions SQL, répondant ainsi à des besoins spécifiques en bases de données.
- De plus, des **mini-modèles**, conçus pour des tâches ciblées ou pour fonctionner dans des environnements à ressources limitées, offrent des solutions légères et efficaces.

Cette diversité illustre la richesse et la complémentarité des approches en traitement automatique du langage. A cela, s'ajoutent les modèles qui transforment du texte en image, son ou vidéo, qui ne seront pas traités dans ce rapport.

3.6. Faiblesse des LLM: les hallucinations

Dans le contexte d'un LLM, une hallucination désigne la tendance à retourner des informations fausses sur un ton d'expert, impossible à distinguer des informations exactes que le LLM peut fournir. Elles diminuent la confiance dans les réponses fournies et posent de multiples problèmes; personne ne veut fournir des réponses inexactes à ses utilisateurs.

Notons que le problème des hallucinations apparaît principalement lorsque l'on utilise des LLM comme base de connaissances; il n'apparaît pas ou peu lorsque le LLM est utilisé comme moteur d'analyse et de génération de texte.

3.7. Acteurs importants

- **OpenAI** est l'entreprise américaine qui a lancé le service ChatGPT. En partie financée par Microsoft, OpenAI entraîne des modèles de base, principalement fermés, et fournit des services gratuits et payants autour de ses modèles. ChatGPT est le produit qui a eu le succès le plus fulgurant de l'histoire de l'informatique.
- **Anthropic** est une autre entreprise américaine, fondée par des anciens d'OpenAI, qui fournit des services autour de son LLM propriétaire appelé Claude. La qualité de leur modèle généraliste rivalise avec OpenAI.
- **Mistral** est une entreprise française qui fournit des services similaires à OpenAI et Anthropic, avec un modèle généraliste entraîné et opéré en France. Mistra.ai publie une partie de ses modèles sous licence open source et fournit un SaaS pour le modèle généraliste de dernière génération.
- **Hugging Face** est une entreprise américaine, fondée par des français, qui est devenue le hub central de la generative AI Open Source. Leur librairie open source originale facilite énormément l'utilisation de modèles ouverts, en simplifiant le déploiement des pipelines de préparation de données et des modèles. Aujourd'hui, Hugging Face est devenu le dépôt central des modèles ouverts. Microsoft et Meta mettent à disposition leurs modèles sur Hugging Face.
- **Meta**, la maison mère de Facebook, est devenu un champion de l'IA générative Open Source. Meta publie régulièrement des modèles ouverts et utilisables commercialement. Par exemple, le modèle généraliste Llama3 a nécessité des millions d'heures-gpu d'entraînement sur les gigantesques clusters NVIDIA de meta (27'000 nodes!). Il y a déjà des centaines de modèles dérivé (fine-tuné) de Llama 3!
- **Autres**: Microsoft, Apple, NVIDIA et d'autres grands groupes publient également des modèles ouverts, parfois ciblant des segments spécifiques du marché (petits modèles pour mobile, modèles spécifiques à une tâche, etc.).

4. Consommer un Grand Modèle de Langage

Il n'existe pas beaucoup d'alternatives lorsque l'on souhaite intégrer un LLM dans une application. On peut soit se reposer sur des modèles propriétaires prêts à l'emploi, soit utiliser un modèle open source qu'il faudra probablement déployer et maintenir.

4.1. Modèles proposés en Software as a Service (SaaS)

A titre d'exemple, **ChatGPT** d'OpenAI est le produit phare de l'IA, c'est un service propriétaire. Basé sur un modèle closed source, OpenAI fournit des services de Chat, de génération d'images, de vidéos, d'analyse de données, faciles à l'emploi, mais payants.

Les modèles propriétaires sont parmis les plus performants et les plus innovants, ils ont été entraînés avec des volumes considérables de données et fonctionnent avec des centaines de milliards de paramètres. Ces modèles ont bénéficié d'investissement faramineux (R&D, conception, entraînement, affinage, inférence) justifiés par un potentiel commercial basé sur une offre de service.

La contrepartie, hormis le prix, réside souvent dans l'utilisation des données que vous lui envoyez dans vos requêtes. Rien ne gage que ces données restent confidentielles ou que leur traitement soit compatible avec les lois sur la protection des données personnelles suisses et européens.

En résumé, l'utilisation d'un modèle propriétaire demeure pertinente pour des tâches complexes, à condition que les données ne soient pas sensibles, que le budget soit adapté, et que l'outil soit largement exploité pour justifier l'investissement.

Notons qu'une entreprise française, **Mistral.ai**, fournit ses propres LLM sous forme de service. Dans ce cas, l'entreprise est française et opère ses modèles en Europe, ce qui simplifie les aspects en lien avec les lois sur la protection des données personnelles.

4.2. Modèles Open Source

Les communautés open source sont pleinement dans la course de l'IA générative et ont littéralement explosé cette année avec des communautés comme Hugging Face qui rassemble des milliers de modèles open source, des données d'entraînement, des poids pour des modèles existants, et plein d'autres services. Lorsqu'il s'agit de modèles spécifiques, destinés à une tâche particulière, la communauté open source propose des solutions très performantes et facilement paramétrables. Par contre, dans le domaine des grands modèles généralistes, les performances ne rivalisent pas encore avec les derniers modèles d'OpenAI comme GPT-4o.

L'avantage d'utiliser de tels modèles est que l'on peut contrôler leur aspect opérationnel, la protection des données ou les coûts de mise en service. On peut aussi choisir son modèle,

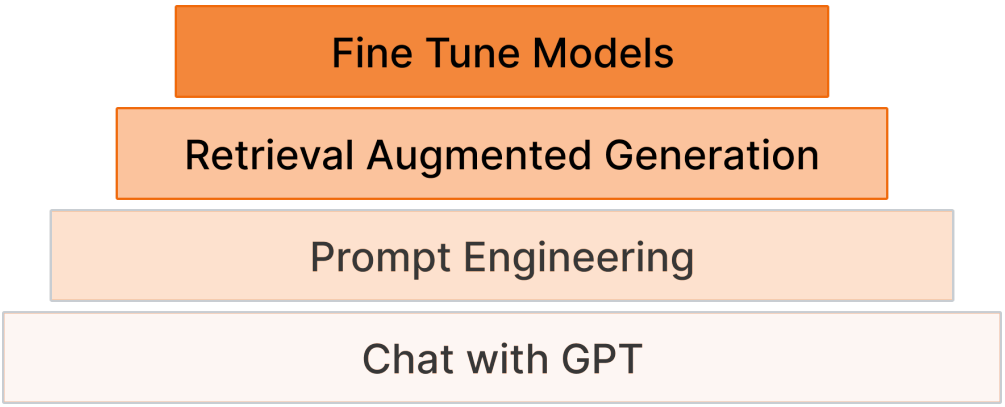
l’entrainer ou le fine-tuner avec les données de nos cas d’usages, donc maîtriser son utilisation.

5. Contexte du géospatial

À l'instar de nombreux autres secteurs, le domaine du géospatial présage de l'énorme potentiel que les LLM peuvent lui apporter. En plus d'amplifier les capacités d'analyse et d'automatisation des traitements de données géographiques, l'IA générative apporte de nouvelles opportunités pour venir enrichir les données et les métadonnées, améliorer la recherche et la découvrabilité des informations, faciliter leur valorisation, et piloter des applications géospatiales au profit d'une meilleure expérience utilisateur et d'une valorisation accrue des données.

5.1. La pyramide de complexité des LLM pour le géospatial

Il existe plusieurs techniques d'intégration des modèles de langages dans les applications géospatiales. Selon le niveau de personnalisation ou d'intégration, la complexité de mise en œuvre varie. Nous représentons ces techniques sous la forme d'une pyramide, chaque étage est une étape de prise en main de la complexité d'intégration des LLM. Plus on monte dans cette pyramide, plus la complexité de mise en œuvre est élevée, mais l'on peut davantage exploiter le potentiel des LLM pour mieux servir nos données et nos usages.



1. **Pyramide de complexité d'intégration des LLM**

5.1.1. Interaction de base

Le premier étage de cette pyramide est simplement l'usage d'un outil propriétaire de chat comme ChatGPT. Il est déjà bien connu et utilisé et est parfaitement capable d'accomplir des première taches liées à de l'information géographique comme

- répondre à des informations de lieu (Où est Paris, quelle est l'emprise de la Suisse etc..)
- Interroger des données géographiques sur lequel il a été entraîné
- Servir d'outil d'analyse de données géospatiales en déposant des fichiers CSV ou GeoJSON. Le LLM lit vos données et répond à vos questions sur leur contenu. Il peut

également générer du texte ou des graphiques en effectuant des calculs à partir de vos différentes entrées.

Cas d'usage: En savoir plus sur les instances de geOrchestra

Une fois le fichier GeoJSON déposé dans le chat, ChatGPT peut analyser son contenu

Combien y a-t-il d'instances de geOrchestra ?

```
- Il y a un total de 31 instances de GeoOrchestra répertoriées dans
  le fichier JSON fourni.
```

Où sont-elles situées ?

```
- Voici un résumé des localisations des 31 instances de GeoOrchestra
  mentionnées dans le fichier JSON :
  - Université de Berne - Berne, Suisse (Coordonnées : 7.4376,
  46.9508)
  - CRAIG - Auvergne, France (Coordonnées : 3.1766, 45.7764) ...
```

Le modèle lit le fichier et ajoute même des informations supplémentaires qui ne sont pas contenues dans le fichier, comme la localisation précise (ex. : Auvergne, France).

Ces interactions sont assez basiques et deviennent limitées si vous souhaitez poser des questions plus spécifiques ou intégrer ces capacités dans votre propre application.

5.1.2. Prompt Engineering

Un **prompt** est l'intégralité du texte qui va être fourni comme input au modèle de langage.

Le prompt engineering, c'est l'art de construire des prompts très exhaustifs pour obtenir des résultats plus spécifiques. Cela permet de contextualiser la requête faite au LLM, en définissant précisément ce qui est attendu comme réponse et comment celle-ci doit être formulée. On peut y préciser l'objectif, le ton, le style, l'audience, le format de sortie, ou fournir des exemples ou toute autre information pertinente.

En général, les concepteurs d'applications définissent des modèles de prompt basés sur les exigences de l'application. Les requêtes utilisateur sont ensuite incorporées dans ce template pour obtenir les résultats désirés grâce au prompt engineering.

Cas d'usage: Afficher les données OpenStreetMap sur une carte, à partir d'une requête en langage naturel.

Requête humaine : *Afficher tous les restaurants végétariens à Bâle*

Voici les différents aspects que le modèle de prompt doit prendre en charge :

Extraction de l'information: Le LLM doit comprendre ce que l'utilisateur veut afficher (les restaurants végétariens) et dans quelle ville (Bâle).

Génération d'une requête API: Le LLM doit générer une requête Overpass (API en ligne gratuite pour récupérer les données OSM) en fonction de la question. Il doit générer une emprise (bbox) pour la localisation et les filtres Overpass corrects pour les types de lieux attendus.

Retour de la requête API: Le LLM doit renvoyer uniquement la requête d'API sans texte supplémentaire.

Fournir des exemples: L'ajout d'exemple de question et de format de réponse aide le LLM à générer des URLs correctes.

La sortie attendue est :

```
https://overpass-api.de/api/interpreter?data=[out:json][timeout:25];nwr[%22amenity%22=%22restaurant%22]%22diet%22=%22yes%22;out;%3E;out%20skel%20qt;
```

Cette approche fonctionne uniquement si le LLM connaît la syntaxe de l'API Overpass, ce qui est généralement le cas avec les grands modèles. Si l'on souhaite obtenir des résultats similaires avec ses propres données, sur lesquelles le modèle n'a pas été entraîné, le prompt engineering ne suffit pas et il faut mettre en place un système RAG (Retrieval-Augmented Generation).

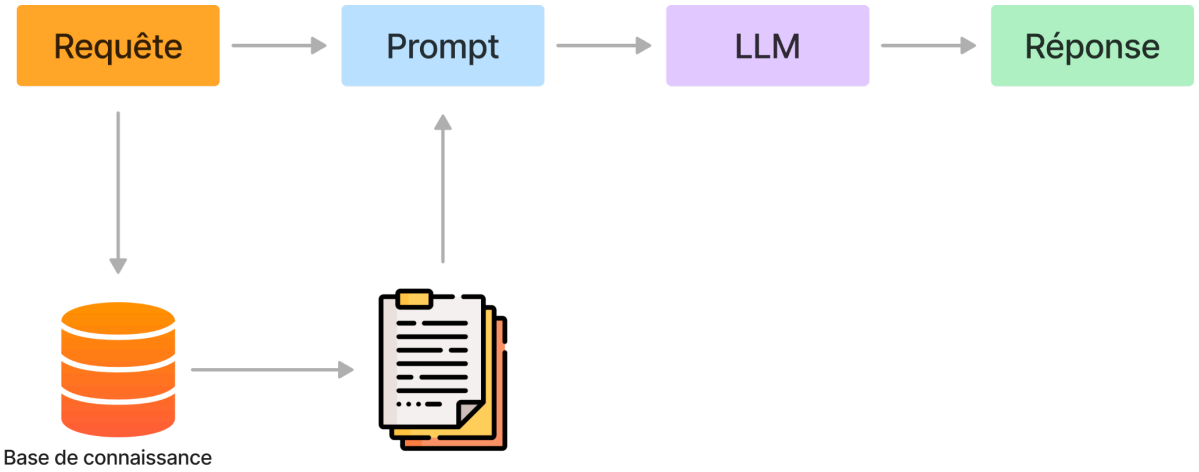
5.1.3. Retrieval-Augmented Generation

Le Retrieval Augmented Generation (RAG) est une technique décrite dans un papier de Meta en 2021 qui explique comment utiliser un LLM avec ses propres données sans les coûts importants générés par l'entraînement ou l'affinage (fine-tuning) d'un modèle. Étant donné que les LLM n'ont pas accès à vos données géospatiales spécifiques, il est nécessaire d'ajouter une phase de récupération (retrieval) pour extraire vos données et les insérer dans le contexte passé au LLM.

Conceptuellement, la technique est relativement simple. Elle est faite de deux étapes qui sont:

- **Le Retrieval (récupération)**
  - Analyse la question de l'utilisateur et cherche (souvent à l'aide de la recherche sémantique) les documents qui contiennent la réponse.
  - Construit un prompt qui contient la question de l'utilisateur et les documents qui permettent d'y répondre (que l'on appelle souvent "le contexte").
- **La Génération**
  - Demande au LLM de générer une réponse à la question en se basant uniquement sur les informations qui se trouvent dans les documents fournis dans le prompt. Insister pour que le modèle réponde "je ne sais pas" si la

réponse ne se trouve pas dans les documents fournis dans le prompt.



2. Architecture classique d'un RAG

Le RAG a de nombreux avantages, il

- Permet l'utilisation d'informations qui n'ont pas été présentées au modèle lors de sa phase d'entraînement.
- Permet l'utilisation d'informations fréquemment mises à jour, sans devoir entraîner ou affiner à nouveau le modèle.
- Grâce aux points ci-dessus, améliore la pertinence des réponses fournies.
- Diminue voire supprime les risques d'hallucination. Le prompt d'un RAG contient généralement une phrase du style "répond uniquement à l'aide des informations fournies en contexte, si la réponse à la question ne se trouve pas dans le contexte, alors répond simplement "je ne sais pas", sans chercher à inventer une réponse". Et cela fonctionne bien.

Cas d'usage: Mise en œuvre d'un système de géo-intelligence sur des données.

Requête humaine: *Quand les ordures sont-elles collectées dans mon quartier, Paris 8ème ?*

Imaginons disposer d'un jeu de données contenant les horaires de collecte des ordures pour Paris, mais que le LLM n'a pas été entraîné sur ces données. Voici comment un système RAG permet de répondre au problème:

- **Indexation des données**: Idéalement, les données sont indexées dans un espace vectoriel sémantique pour permettre un retrieval rapide et précis.
- **Extraction de la requête**: Le système RAG identifie les éléments clés de la requête (horaires de collecte des ordures), le jeu de données qui contient ces informations et la localisation (Paris 8ème).
- **Géocodage**: Le système récupère la géométrie de Paris 8ème à l'aide d'un service de géocodage.

- **Recherche sémantique**: Le système effectue une recherche sémantique pour trouver le jeu de données adéquat, comme le “planning collecte des ordures”.
- **Récupération du jeu de données** : Le RAG extrait les objets géographiques (horaires et positions des lieux de collecte) des jeux de données à partir des liens disponibles dans les métadonnées (les liens d'accès à la donnée).
- **Intersection spatiale** : Le système réalise une jointure spatiale pour trouver les horaires spécifiques de la localisation demandée.
- **Réponse finale**: Les objets géographiques pertinents sont passés au LLM, qui analyse la structure et le contenu des données et fournit l'heure précise de collecte des ordures.

Le RAG est une technique très utilisée qui a de nombreux avantages. Une grande partie des applications basées sur les LLM développées aujourd’hui utilise le principe du RAG. **Avec le RAG, on n’utilise pas le LLM comme base de connaissances, mais uniquement pour sa capacité à comprendre et générer du texte**.

Pour en savoir plus:

➔ Qu'est-ce que la génération augmentée de récupération [oracle.com]
➔ What Is Retrieval-Augmented Generation, aka RAG? [nvidia.com]

5.1.4. Fine-tuning

Le fine-tuning, ou “affinage” en français, est une technique qui consiste à prendre un modèle de base, entraîné sur des millions d’éléments et de continuer à l’entraîner avec un jeu de données spécifique à un problème donné.

Bien que plus complexe à mettre en œuvre, le fine-tuning permet d’atteindre plus de justesse dans les retours du modèle, car il connaît précisément votre patrimoine de données géographiques et/ou leur structure.

Exemples appliqués au géospatial:

- **Fine-tuner un LLM sur des données géospatiales spécifiques**: Il est possible d’affiner un modèle sur des données géospatiales pour qu'il réponde directement aux questions sans avoir besoin de récupérer des informations. Cependant, le RAG est généralement préféré pour cet usage car l'affinage et l'hébergement d'un LLM peuvent être très complexes et coûteux.
- **Fine-tuner un modèle d’embedding de métadonnées**: Dans le processus RAG, la phase de récupération basée sur la recherche sémantique est cruciale. Trouver le bon jeu de données correspondant à la demande de l'utilisateur est essentiel pour que la réponse soit pertinente. Les modèles d’embedding sont utilisés pour représenter le texte sous forme vectorielle dans des espaces sémantiques. De ce fait, affiner ces modèles leur permet s’adapter au mieux à la structure d’une métadonnée

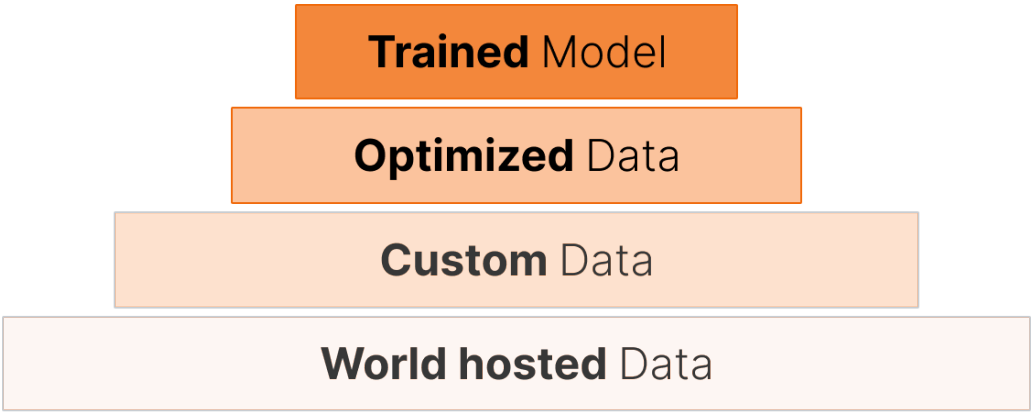
(DCAT, ISO) ainsi qu'à vos propres données pour spécifiquement porter plus de poids à certains éléments de la métadonnée.

- **Fine-tuner un modèle text-to-SQL pour générer des requêtes spatiales**: De nombreux modèles peuvent transformer du texte en requêtes SQL s'ils connaissent la structure des jeux de données (afin d'utiliser les bonnes colonnes pour la requête). Il est possible d'entraîner un modèle à forger des requêtes spatiales, telles que des requêtes SQL PostGIS, pour gérer l'intelligence géographique en joignant, agrégeant et calculant des indicateurs géospatiaux. On pourrait également envisager d'autres modèles comme text-to-OSM, text-to-OGC-API, qui permettraient au RAG d'interagir directement avec les standards géospatiaux open.

5.2. La pyramide de complexité d'un point de vue de la donnée

Comme un miroir à la pyramide de complexité des techniques d'intégration des LLM, la pyramide des données montre que la nature des applications intégrant l'IA générative dépend aussi beaucoup de l'exposition des données géographiques sur lesquelles elles s'appuient.

La donnée peut exister sous différentes formes, chaque forme dépend de son usage et des ambitions des résultats escomptés.



3. **Pyramide d'intégration des données dans les applications LLM**

5.2.1. Données de référence

On parle ici de données de référence dont la structure est largement publiée sur Internet comme les données OpenStreetMap ou Overture Maps. Les grands modèles de langages propriétaires ont été entraînés sur ces structures et connaissent donc la nature de ces données. Cela signifie que ces modèles sont capables de répondre à des questions y faisant référence, souvent pour les afficher selon certains critères. Voici quelques exemples:

OpenStreetMap: l'API OverPass est un service public permettant de récupérer les données d'OpenStreetMap. Les spécifications de cette API, la nomenclature, la syntaxe et les

différents paramètres sont connus des LLM si bien qu'ils peuvent générer des requêtes OverPass à partir des prompts utilisateurs.

Un LLM type ChatGPT est donc capable de générer les URLs Overpass pour ce type de question:

- *Les sommets de plus de 4000m en Suisse*
- *Les ponts au dessus de rivières autour de Lausanne*
- *Les restaurants végétariens à Paris*

Eg.

```
["natural"="peak"](if:number(t["ele"])>500)(45.8179,5.9576,47.8085,
10.4921)
```

Les URL générées permettent de récupérer les objets géographiques correspondant au format GeoJSON par exemple.

Overture Maps: La fondation Overture Maps propose un jeu de données plus complet que OpenStreetMap, incluant d'autres sources de données (Meta, Google, Microsoft) souvent acquises par IA (traitement d'image). La fondation met à disposition ces données au format Cloud Native GeoParquet, sur lequel on peut exécuter des requêtes SQL avec des outils comme DuckDB. Des partenaires (type CARTO ou Snowflakes) distribuent ces données sous d'autres formats comme BigQuery, la base de données Cloud de Google.

La capacité à interroger ces données via des requêtes SQL décuple les possibilités de retrieval et ouvre les portes de la géo-intelligence. Il s'agit de questionner le territoire pour en extraire des indicateurs d'aide à la prise de décision. On peut faire des jointures, des agrégations, des calculs et tout ce que le SQL géospatial permet.

Eg. *Répartition des bornes de recharge de voiture électriques, sur un maillage H3 de niveau 6 à Londres*

Le LLM est capable de générer une requête BigQuery complexe permettant de récupérer les objets géographiques demandés, afin de créer une carte de répartition.

4. Représentation cartographique de la requête générée par LLM

Toutes ces opérations peuvent être réalisées uniquement par Prompt Engineering. Pour réaliser le même genre de requête sur des données géographiques propriétaires, sur lesquelles les LLM n'ont pas été entraînés, il faut construire un RAG.

5.2.2. Données spécifiques d'une plateforme

Il s'agit généralement des jeux de données publiés sur des plateformes de données géographiques (SDI) comme l'Infrastructure Nationale de GéoDonnées ou des plateformes geOrchestra. Les LLM ne connaissent ni leurs données, ni leurs structures. Ces données sont généralement stockées dans des bases de données géographiques (eg. PostGIS) et diffusées via des APIs.

Si l'on souhaite poser une question sur le contenu d'un jeu de données de notre plateforme (eg. *Combien d'accroches vélo dans ma ville?*), une phase de retrieval est nécessaire pour récupérer le jeu de données non connu et le passer au LLM.

A ce stade, la phase de retrieval se basera sur les APIs publiques standards disponibles (OGC APIs). Durant la phase de génération, le LLM se chargera d'analyser le contenu du jeu de données et répondra à la question posée.

Globalement, les architectures de type RAG sont souvent préconisées pour appliquer le potentiel des LLM à ses propres données.

5.2.3. Données préparées et optimisées pour la recherche sémantique

L'IA générative peut-être utilisée en amont de la phase de génération pour améliorer la recherche dans les données ou les métadonnées. Pour cela, les données doivent être transformées pour devenir plus intelligibles pour les LLM.

5.2.3.1. Sans optimisation

Un RAG peut utiliser une recherche lexicale pour sa phase de retrieval. Pour la recherche de métadonnées par exemple, une plateforme géographique diffuse des services standards de découverte, le CSW ou OGC API Records, dont les spécifications sont connues des LLM. On peut donc imaginer le workflow suivant dans notre RAG

1. Transformation de la requête en langage naturel vers une requête OGC API Records par un LLM
2. Appel de la requête OGC Records
3. Retour des résultats

Ce workflow permet d'intégrer le langage naturel dans une application de recherche, mais n'améliore pas la recherche, basée sur les mêmes technologies.

5.2.3.2. Avec optimisation

Les modèles d'embedding permettent de stocker les métadonnées dans un espace vectoriel pour faire de la recherche sémantique. Cette recherche présente plus de potentiel qu'une recherche lexicale, car elle se base sur le sens de la question et non sur des mots-clés (nous décrirons cette technique dans la section suivante).

L'optimisation (indexation) vient de la façon dont on représente les métadonnées dans un espace vectoriel adapté à la recherche sémantique.

Ce qu'il est important de comprendre à ce stade, c'est que l'on peut ingérer ses données et ses métadonnées dans une base de données à espace vectoriel pour profiter de l'IA dans la recherche. Ceci signifie que les APIs standards OGC (comme celles de GeoNetwork) ne seraient plus utilisées dans le RAG.

5.2.4. Données directement intégrées dans le modèle

Enfin, le dernier niveau d'intégration des données est le fine-tuning. On apprend à un modèle existant à répondre aux questions de notre territoire. Le modèle ingère toutes les données et les inclut dans ses réponses. Concrètement, les données seront donc présentes dans les poids du modèle.

Il est recommandé de n'utiliser le fine-tuning de modèle qu'en dernier recours, et de privilégier le RAG. D'après 2 publications :

- RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture
- Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLM

Il apparaît que le rapport coût-bénéfice du RAG est généralement nettement meilleur que celui de l'affinage. Ce dernier est généralement bien complexe à mettre en œuvre (compétences et coûts) et à maintenir dans le temps.

6. Cas d'usages géospatiaux

6.1. Améliorer la recherche des données géographiques

Le point d'entrée des plateformes de géodonnées et des portails open data est souvent une barre de recherche, c'est le point de départ de toute analyse: trouver les jeux données qui contiennent potentiellement l'information que je recherche. C'est un enjeu crucial pour l'analyse de tendance, la géo-intelligence ou la prise de décision face à des enjeux territoriaux.

Généralement, les outils mis à disposition pour rechercher une donnée ou un jeu de données sont basés sur des outils de recherche lexicale ("full text search"). Pour que cette recherche fonctionne, il faut que l'utilisateur cherche les mêmes mots-clés que ceux utilisés dans la description de la donnée. L'utilisateur doit donc posséder une bonne connaissance du domaine, connaitre la structuration des jeux de données et savoir comment formuler correctement sa requête pour que le système fasse une réponse pertinente.

L'IA générative propose des outils pour cibler les utilisateurs non experts des plateformes de géodonnées en leur fournissant un nouveau moyen d'interagir avec leur plateforme via le langage naturel.

6.1.1. Définition de la recherche sémantique

La recherche sémantique est un type de recherche qui se base sur la signification, le sens des phrases et des mots. Par opposition, la recherche lexicale se concentre sur la similarité lexicale, c'est-à-dire les mots en commun entre la recherche et les documents du jeu de données.

Une propriété très intéressante de la recherche sémantique dans le contexte suisse est que cette dernière est par essence multi-lingue.

6.1.2. Fonctionnement

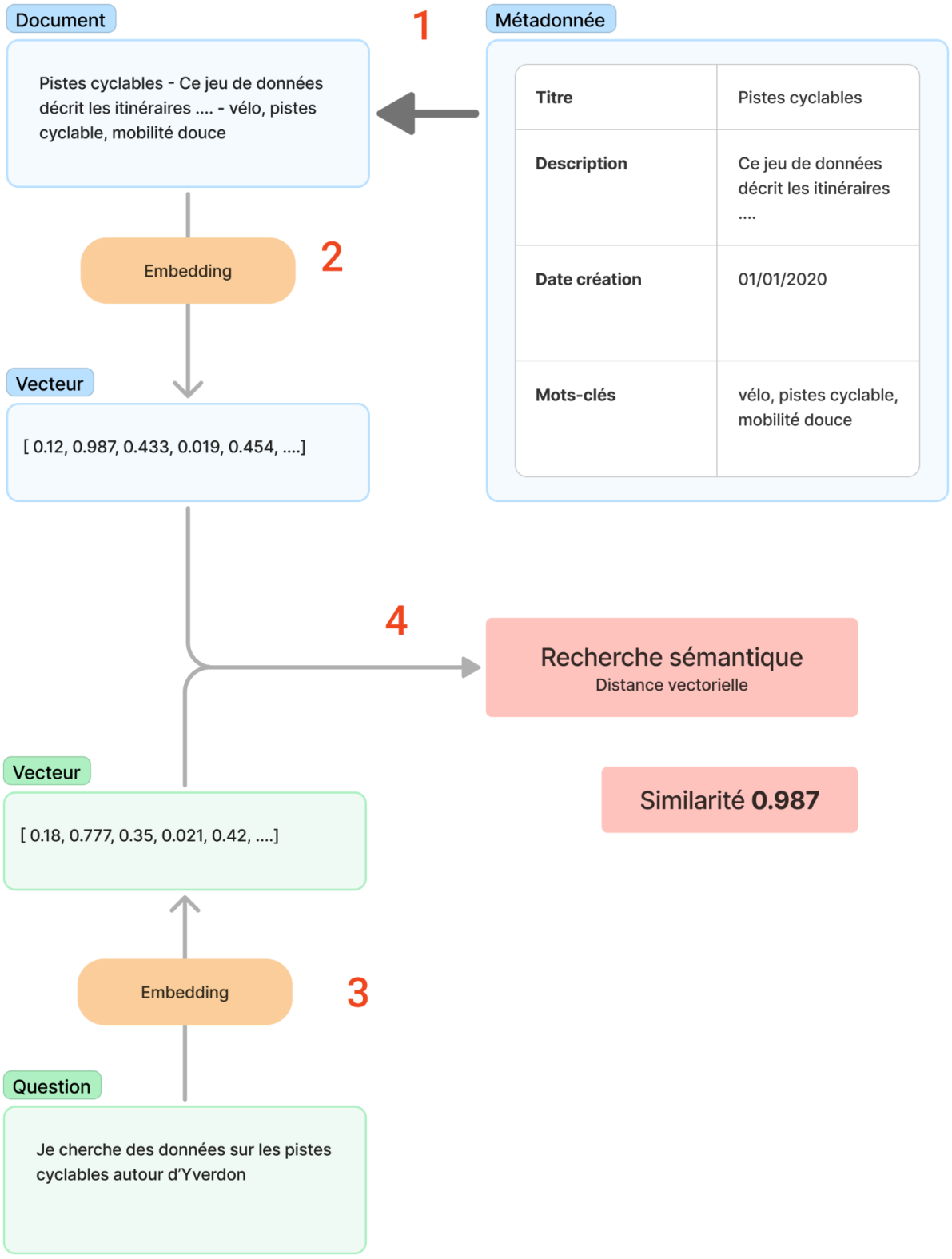
L'idée est d'indexer dans une base de données vectorielles, via un modèle d'embedding, les informations sur lesquelles s'appuie la recherche.

Un embedding permet de transformer du texte en représentation vectorielle, dans des vecteurs à plusieurs centaines, voire milliers de dimensions. On peut voir une dimension comme un poids indiquant à quel point le texte représente une idée :

- *Est-ce que ça parle de vélo*
- *Est-ce que le texte est positif*
- *etc...*

Ainsi, la valeur de ce vecteur détermine son sens.

Dans le contexte du géospatial, ce sont les métadonnées qui permettent de retrouver facilement un jeu de données. Les métadonnées décrivent, en partie, la nature du jeu de données, son contenu et la façon dont il a été créé. Le titre, les mot-clés et la description sont par exemple d’excellentes informations à indexer pour la recherche.



5. **Fonctionnement de la recherche sémantique pour les métadonnées géographiques**

1. Sachant que les modèles d'embedding ne fonctionnent que sur du texte, il faut agréger les métadonnées qui nous intéressent sous forme de texte, en concaténant par exemple le titre, le résumé et les mots-clés.
2. Ce texte est ensuite indexé dans la base vectorielle. Ainsi, chaque métadonnée du catalogue aura une représentation sous forme de vecteur (représentant la signification du contenu de la métadonnée), et pourra être soumise à une recherche sémantique.
3. Lors de la recherche, la requête de l'utilisateur est transformée via le même modèle d'embedding en représentation vectorielle.
4. Ainsi, calculer la similarité entre la requête et une métadonnée revient à calculer la distance entre le vecteur de la question et le vecteur de la métadonnée. Les données géographiques qui nous intéressent sont donc celles pour lesquelles la distance entre la question et la métadonnée est la plus petite (et la similarité la plus élevée). Pour extraire uniquement les données pertinentes de cette liste de résultats, on peut appliquer un seuil sur le score de similarité ou bien utiliser un nouvel appel à un LLM pour décider du ou des résultats à renvoyer à l'utilisateur.

Pour en savoir plus:

- Semantic Search with Sentence Transformers [sbert.net]
- What is embedding? [ibm.com]
- Qu'est-ce que la recherche sémantique? [elastic.co]
- Qu'est-ce que la recherche vectorielle? [elastic.co]
- Semantic Search for Geospatial Data Discovery [linkedin.com]

6.1.3. Recherche hybride

La recherche sémantique apporte des fonctionnalités intéressantes et permet de contourner des problèmes de la recherche lexicale.

Néanmoins, l'expérience montre que dans beaucoup de cas, les meilleurs résultats de recherche sont obtenus en combinant plusieurs techniques de recherche. Les résultats de la recherche sémantique sont combinés avec des résultats en provenance d'une recherche lexicale ou géographique. Comment combiner ("merger") les résultats de différentes recherches est un sujet qui doit être traité avec attention et pour lequel il y a plusieurs solutions possibles.

La combinaison de la recherche sémantique et de la recherche lexicale (et géographique) est appelée recherche hybride ("hybrid search").

Pour en savoir plus :

- Hybride Retriever avec DuckDB [Architecture et performance].

La recherche sémantique est très importante pour une bonne phase de retrieval. Si l'on pose la question *"Quelle est l'évolution des bornes de recharges électriques à Lyon ces 5 dernières années"*, il est primordial que notre système trouve quelle donnée du catalogue

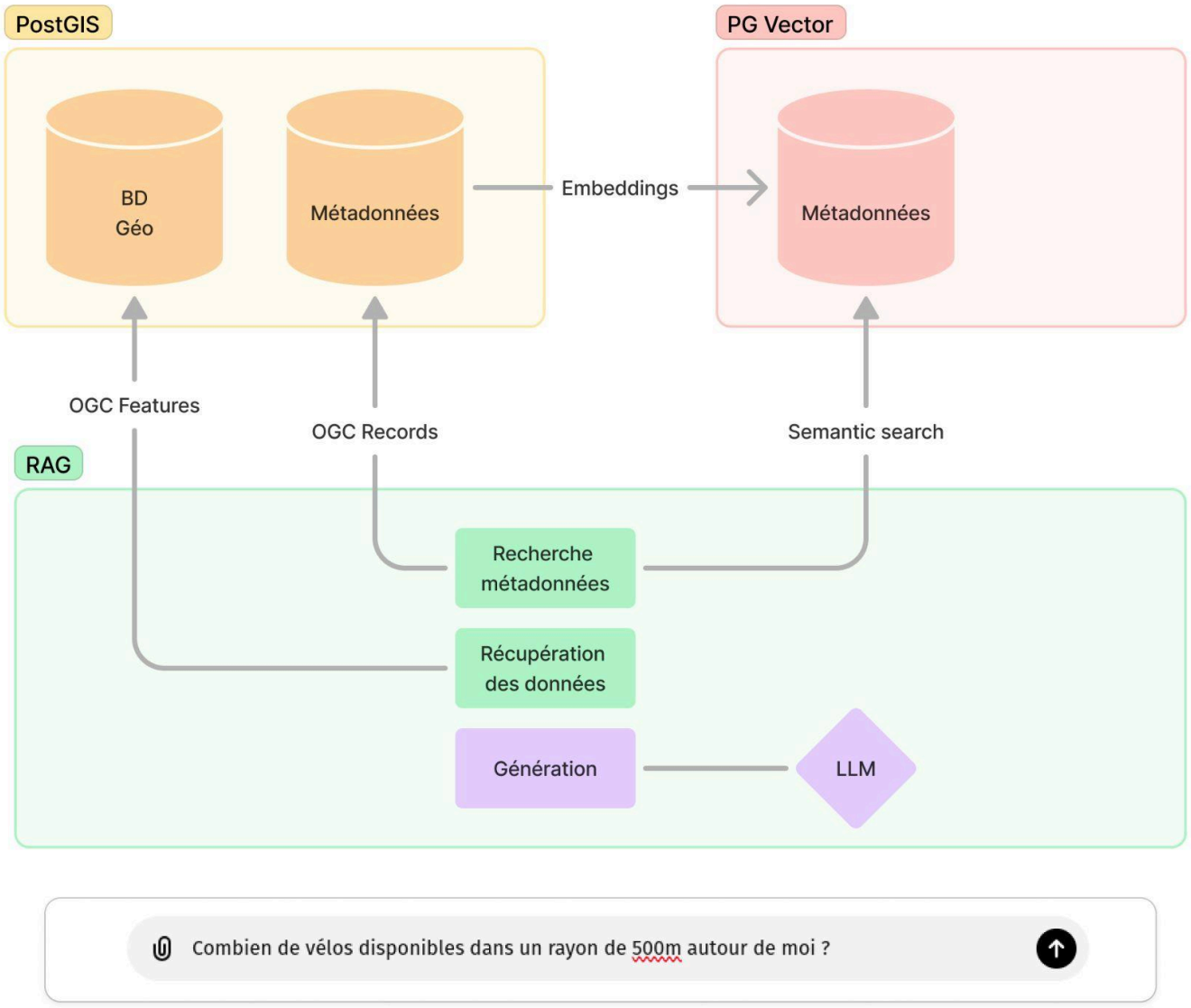
contient cette information. Voyons maintenant ce que l'IA apporte pour répondre concrètement à cette question, en allant chercher à l'intérieur des données.

6.2. Opérer son territoire via la géo-intelligence

La géo-intelligence, c'est la science qui tente de répondre à des questions diverses (territoriales, sociales, économiques, sociétales) par l'information géographique. Croiser des données, faire des cartes de répartition, extraire des tendances; tout ce qui peut aider à la prise de décision. C'est vraiment là que réside le potentiel des LLM: laisser aux publics ou aux décideurs l'opportunité de poser des questions complexes en langage naturel, et attendre que le système basé sur l'IA croise les bonnes informations et génère les bons indicateurs.

Comme vu précédemment, le RAG est l'architecture préconisée pour ce type de fonctionnement. Bien sûr, il existe de nombreuses façons de construire ce système.

Voici globalement comment monter un RAG pour gérer la géo-intelligence en langage naturel.



6. Architecture RAG pour la récupération de données géographiques

La phase de génération peut appeler un LLM pour extraire l'information demandée depuis l'une ou les couches de données géographiques récupérées. Néanmoins, on lui déporte une grosse responsabilité, sans forcément garantir la pertinence des résultats.

C'est ici que l'on peut intégrer des modèles plus spécifiques comme des modèles text2SQL qui vont générer la requête SQL adéquate, sur laquelle le système pourra itérer.

Les entreprises qui fournissent des services géographiques propriétaires ont globalement toutes la même ambition de développer les services d'IA qui répondront aux problèmes de géo-intelligence (CARTO, Esri, Google, Apple, Mapbox etc...).

6.3. Enrichir l’information

Le succès d’une phase de retrieval dépend de la qualité des données qui seront ingérées dans le système.

6.3.1. Génération et enrichissement des métadonnées

Pour trouver la bonne donnée, il faut qu’elle existe, mais aussi qu’elle soit bien documentée et décrite. On retombe toujours sur le même constat, créer des fiches de métadonnées bien documentées et fournir un service de catalogage et de découverte est primordial pour le succès d’une plateforme de géodonnées.

L’IA peut aider à enrichir les métadonnées, ou même à les générer. Voici les tâches qu’elle peut accomplir :

- Enrichir le texte d’une description trop succincte. Ceci peut-être fait à partir d’un LLM, ou via une phase de retrieval qui peut s’inspirer de catalogues existants.
- Extraire des mots-clés, des thèmes et ou d’autres informations de la description.
- Générer une fiche de métadonnées à partir de la structure d’un jeu de données (noms des attributs) et de son contenu.

6.3.2. Feature labelling

La feature est l’objet géographique, que l’on peut venir enrichir de la même manière que ce qui est proposé pour les métadonnées. Cela fonctionne particulièrement si l’objet contient des descriptions sous forme de texte.

Par exemple, les itinéraires de randonnées de schweizmobile.ch contiennent de nombreux champs textuels. Tout d’abord une description étoffée de l’itinéraire, puis a des relations vers des points d’intérêt, des hôtels, des restaurants, des magasins de location de vélo ou tout autre éléments pouvant intervenir dans la préparation ou la réalisation d’une sortie outdoor.

A partir de la description textuelle d’un itinéraire, on peut extraire des informations non stockées en base comme :

- Faut-il porter son vélo ?
- Sommets que l’on a en visuel pendant l’itinéraire

Ces informations ne sont pas stockées directement sur une base de données, mais elles sont présentes dans le texte. Le feature labelling va permettre de créer des nouvelles colonnes dans la base de données avec ces informations, afin qu'elles soient indexées pour la recherche lexicale. Même si l’on met en place une recherche sémantique, préparer ces index améliorera grandement les performances du système.

Recherche classique à options: sommet Mt Cervin - Pas de portage de vélo

Recherche en langage naturel: Je recherche un parcours de VTT sans portage, 30km maximum, avec vue sur le Mont Cervin et un chouette restaurant pour faire une pause à midi.

6.4. Les relations sémantiques

Les interfaces de traitement du langage naturel utilisant des grands modèles de langage devraient non seulement faciliter la recherche de données, mais également révéler les relations sémantiques entre des réseaux d'objets géographiques à travers différents domaines.

Toutes les données peuvent être reliées entre elles par des liens sémantiques, ces liens sont représentés par des Graphes de Connaissance Géospatiaux (Spatial Knowledge Graphe SKG). L'IA et notamment les LLM sont capables d'extraire les interactions entre les données et révéler une prise de conscience spatiale dans ces domaines en constante évolution (notamment les domaines de l'économie, la santé publique et le climat).

En d'autres termes, les SKG, mis à disposition via des Infrastructures de Graphe Géospatiaux (GKI), viennent apporter une conscience spatiale aux LLM, décuplant leur potentiel de compréhension, d'analyse et d'anticipation des relations de cause à effet entre les données à composante géographique.

6.5. Piloter une application

La compréhension du langage naturel offre la possibilité de piloter une application par écrit ou oralement.

Les assistants personnels comme Siri ou Alexa sont des exemples d'assistants personnels généralistes, basés sur le langage oral.

Dans le même ordre d'idée, il est possible de commander une application à l'aide du langage écrit, ou alors de changer l'état d'une application en fonction d'un dialogue entre un utilisateur et un chatbot. Dans l'exemple de recherche sémantique que nous avons posté sur linkedin, on voit que lorsque l'utilisateur demande "La fibre optique est-elle disponible à Châtel-St-Denis ?", la carte se centre et zoom sur Châtel-St-Denis. La question de l'utilisateur à générer une recherche de layer et des actions (centre, zoom) sur la carte.

6.6. Assistant et Chatbot

Pouvant combiner plusieurs cas d'usage décrits ci-dessus, les LLM sont souvent utilisés comme chatbot avec lequel l'utilisateur peut interagir. Dans le contexte du géospatial, cet assistant peut trouver des données, interroger les données, piloter une application et maintenir globalement une discussion et interagir avec les éléments de la plateforme de géodonnées.

Mettre au point un chatbot nécessite de construire un RAG qui va garder une mémoire de la conversation. Souvent, on demande au LLM de reformuler la nouvelle requête de l'utilisateur à partir de l'historique enregistré.

Exemple

- *Je cherche des données sur les parcs nationaux suisses.*
- Oui, j'ai une couche de données sur le Parc national suisse.
- *As-tu d'autres données ??*

La seconde question sera transformée en :

- *As-tu d'autres données sur les parcs nationaux suisses, hormis la couche sur le Parc National Suisse*

Ensuite passée au LLM pour une phase de retrieval et de génération.

7. Applications géospatiales basées sur les LLM

Cette section liste des exemples d'utilisation de LLM dans le domaine des applications géospatiales. Le domaine étant en pleine ébullition, cette section sera obsolète avant que vous ayez terminé de la lire!

7.1. Recherche conversationnelle

7.1.1. GoogleMap

Utilisée principalement en voiture, sur téléphone, pour guider la navigation, l'application Google Map offre depuis le début de l'année 2024 des fonctionnalités basées sur l'interprétation du langage naturel, par texte ou via le micro de l'appareil. L'application va analyser la requête en langage naturel pour trouver des points d'intérêts aux alentours de la position de l'utilisateur. Elle propose des fonctionnalités de conversation (chatbot).

Pour en savoir plus.

7.1.2. Mapbox MapGPT

Exactement dans le même principe, le concurrent à Google Map pour la navigation, Mapbox, propose également des fonctionnalités de chatbot dans son application mobile. L'application propose d'avoir des conversations pour localiser des points d'intérêt, des routes, des itinéraires de façon dynamique et intégrer à la navigation actuelle.

Pour en savoir plus.

7.1.3. Natural Language Geocoding

Il s'agit d'utiliser le Natural Language Processing pour du geocoding (de la recherche de localisation). Cela peut être par point d'intérêt, par région administrative, par adresse ou tout autre façon de rechercher un lieu.

On peut imaginer de nouvelles possibilités par rapport à des services de geocoding classiques comme par exemple combiner plusieurs régions en incluant des opérations spatiales

- Au sud du Canada
- Entre la Seine et la Loire
- Etc...

Pour en savoir plus.

7.2. Recherche de données

7.2.1. Via un catalogue de données

Comme expliqué plus haut dans ce document, l'IA générative offre des fonctionnalités de recherche sémantique basée sur le sens du contenu et de la requête. Les points d'entrée principaux des données géographiques sont les catalogues de données (GeoNetwork, GeoServer, Opendatasoft et autre). Certaines de ces applications offrent des services de recherche qui peuvent être améliorées grâce à l'IA générative, pour les autres comme GeoServer qui ne fournit que des GetCapabilities, l'IA permet de disposer de services de recherche qui n'existent pas.

Le principe est d'ingérer ces catalogues (OGC API, CSW, GetCapabilities, API custom) dans une base de données vectorielle pour faire de la recherche sémantique sur n'importe quel catalogue de données.

Exemple de recherche sémantique sur GeoNetwork

7.2.2. Via l'API overpass

La récupération de certaines données, comme les données OpenStreetMap via l'API overpass peut être assez laborieuse du fait de la complexité de l'API. L'IA générative, qui a été entraînée sur la documentation d'Overpass est capable de transformer une requête en langage naturel en requête d'API Overpass avec

- Les données recherchées
- Des filtres
- Une localisation

Le tout est réalisé grâce à du prompt engineering permettant l'extraction des éléments recherchés dans la requête utilisateur et la génération de la requête HTTP overpass.

OSM étant la base de données libre la plus fournie, il est important de proposer la visualisation des données OSM ou OvertureMaps via LLM.

7.3. Utilisation des données

Pour aller plus loin, l'IA générative est très utile dans l'extraction d'informations contenues dans les données, comme des tendances, des graphiques ou des agrégations. C'est une des utilisations les plus prometteuses des LLM pour le traitement des données.

7.3.1. Esri Chatbot

Le leader du SIG propriétaire, Esri, a présenté cette année ses avancées en termes d'intégration de l'IA au sein de leur suite d'outil. Il s'agit d'un chatbot permettant d'interagir et d'exploiter les données contenues dans les bases de données.

L'utilisateur peut chercher des données, et poser des questions en langage naturel dont la réponse peut être contenue dans un jeu de données.

Par exemple, "quand les poubelles passent-elles ?" va trouver les données sur le ramassage des poubelles, connaît notre adresse et va déterminer les jours.

L'application permet de chatter, de trouver des données, de répondre à des questions, de générer des cartes en affichant les données, d'exécuter des requêtes sur les données etc...

Pour en savoir plus

7.3.2. CARTO AI agents

CARTO est un leader mondial de la Geospatial Business Intelligence. Ils viennent de lancer leurs agents AI qui sont capables de traiter des problèmes complexes liés à des données géographiques. Le créneau de CARTO est vraiment l'analyse et le traitement de données, et intègre des workflow extrêmement avancés.

On peut demander par exemple "Quel est le pourcentage de marchandise que l'on peut livrer chez les banques couvertes par le projet JED en moins de 20 minutes ?". De nombreux traitements sont nécessaires pour répondre à ce besoin, d'où l'utilisation d'agents complexes.

Pour en savoir plus

Vidéo exemple de chatbot lié à la GeoBI.

7.3.3. Aino

Aino est un produit qui montre l'utilisation du langage naturel pour composer une carte à partir de données OpenStreetMap en y intégrant des traitements. C'est un projet start-up qui fait valoir l'utilisation de l'IA sur des applications cartographiques. Camptocamp réalise le même genre d'application dans le cadre de sa R&D, mais ne vise pas de produit.

Aino permet de trouver des données OSM sur une localisation, dans un rayon, sur un pays, il permet de faire de la distribution de points sur des régions géographiques, ou via une répartition H3 et fournit d'autres services classiques de traitement de données.

Pour en savoir plus

7.4. Enrichissement de données

7.4.1. Overture Maps

L'IA générative permet l'enrichissement de données, la catégorisation et le feature labeling sur les données vectorielles.

Par exemple, les données Overture Maps proviennent de différentes sources de données (OSM, ESRI, TomTom et d'autres). Les données doivent être agrégées et homogénéisées sur le schéma Overture Maps (ontologies, dictionnaires, thesauri). Pour cela, les données OpenStreetMap qui sont ingérées possèdent elles-mêmes une classification via des tags, dont la correspondance avec les champs Overture Maps est calculée grâce à l'IA.

Cet article illustre bien l'utilisation des Embeddings pour la labellisation des données Overture Maps.

7.5. Text2SQL

7.5.1. Overture Maps GPT

Un plugin ChatGPT permet, via du prompt engineering, de générer des requêtes SQL sur le schéma Overture Maps BigQuery de CARTO. Ces requêtes peuvent ensuite être utilisées dans des outils ou du code pour utiliser les données Overture Maps comme base de GeoBI.

Exemple: *Quelle est la répartition spatiale des chargeur de voitures électriques à Paris*

Comme on peut le constater, de nombreuses initiatives tentent d'appliquer les LLM au monde du géospatial. Hormis les grands éditeurs propriétaires, l'ensemble des travaux existants n'en sont encore qu'au stade expérimental. Nous avons à découvrir comment tirer partie de cette technologie, et à quelles fins l'utiliser. Globalement, l'objectif principal reste les outils de géo-intelligence, mais toutes les facultés de préparation et d'enrichissement de données laissent présager de nombreux usages destinés à améliorer la connaissance et l'utilisation des données géographiques.

8. Evaluations

Un défi majeur lors du développement d'applications basées sur le langage naturel est l'évaluation de la qualité de l'application.

8.1. Particularité des applications basées sur le langage naturel

Lors du développement d'une application classique, les fonctionnalités sont généralement spécifiées à l'aide d'une série de cas d'utilisation qui définissent exactement le comportement de l'application en fonction des actions de l'utilisateur. Si l'utilisateur clique sur le bouton B, l'action A est exécutée et l'application se trouve dans l'état E. Les cas d'utilisation sont déterministes et non ambigus.

Dans le cas d'une application basée sur le langage naturel, l'utilisateur pose une question et le système lui répond. La question de l'utilisateur et la réponse du système peuvent être formulées de plusieurs manières différentes, sans qu'une formulation ne puisse être considérée comme juste et les autres fausses. Le cas d'utilisation fait donc correspondre un ensemble de formulations possibles pour une question à un ensemble de formulations possible pour une réponse. Il est ambigu et non déterministe.

Une application classique est généralement testée et évaluée à l'aide d'une suite de tests binaires qui sont soit réussis soit échoués. Pour une application basée sur le langage naturel, l'évaluation est beaucoup plus complexe. Il faut évaluer la pertinence de la réponse et la noter.

8.2. Evaluation et tests des applications basées sur le langage naturel

Dans certains cas, il est clair que la réponse à un type de question doit contenir un élément ou un mot précis. Dans ce cas, il est possible d'écrire des tests qui vérifient que la réponse du système contient le mot attendu.

Dans d'autres cas, il n'est pas possible d'attendre un mot précis dans la réponse. Dans ce cas, il est fréquent d'utiliser un LLM pour juger de l'adéquation de la réponse ou de la similarité de la réponse fournie avec la réponse attendue.

8.3. Importance de l'évaluation pour améliorer la qualité

Beaucoup de paramètres et de techniques peuvent influencer la réponse fournie par une application basée sur des LLM. Afin d'optimiser une application RAG et d'améliorer sa qualité, il est nécessaire d'être capable de mesurer l'impact des améliorations apportées à l'application. Ceci n'est possible que s'il existe une bonne mesure de la qualité de l'application.

Ce qui ne se mesure pas ne s'améliore pas (W. Edward Deming).

9. Limitations et considérations

Les LLM présentent un potentiel indéniable pour simplifier les problématiques liées à l'information géographique. Ils révèlent une vraie valeur sur certains domaines comme par exemple la recherche, l'aide à la génération de métadonnées, la compréhension et le traitement de gros volumes de données ou l'extraction d'information à partir de données structurellement peu complexes. Néanmoins, la mise en œuvre reste difficile à appréhender, et les LLM ne peuvent pas tout faire à la place d'un humain, leur usage reste pour le moment limité à des opérations assez basiques. Regardons leurs limites pour mieux comprendre comment les intégrer dans les applications géospatiales.

9.1. Mise en oeuvre

La plupart des systèmes basés sur les LLM, fondés sur du prompt engineering ou du RAG reposent sur les modèles les plus impressionnants et les plus puissants : les grands modèles généralistes comme GPT ou Claude. Ces modèles sont propriétaires, ils ont nécessité des centaines de millions d'euros d'investissement, et sont payants.

9.1.1. Mise en exploitation

Beaucoup de prototypes et applications expérimentales utilisent ces modèles pour impressionner, mais la mise en production de ces systèmes révèle d'autres considérations qui sont difficilement résolues sans moyens et compétences exceptionnelles.

Le passage en production est un enjeu fondamental:

- la mise à l'échelle
- le coût des licences propriétaires
- l'adhérences à des solutions tierces
- la sécurisation des données

Ces considérations imposent souvent le choix d'opérer soit même des modèles open source. Lama 3 est un modèle généraliste open source qui peut être utilisé mais il est souvent plus raisonnable d'opter pour des modèles plus spécifiques, plus petits. Trouver les bons modèles, les fine-tuner et les tester est une tâche complexe et fastidieuse. C'est un long parcours qui demande beaucoup d'investissement.

9.1.2. Complexité des modèles

Les grands modèles de langage sont généralement massifs, ce qui pose des problèmes lorsqu'il s'agit de déployer des solutions à échelle locale ou dans des environnements contraints.

9.2. Architecture

La solution privilégiée pour tirer partie du potentiel des LLM combiné à des données spécifiques est le RAG. Le RAG se base sur une phase de recherche de contenu dans une base de connaissance (phase de Retrieval). La qualité d'une application RAG dépend donc directement de la qualité de la base de connaissance sur lequel il s'appuie. La limitation d'un RAG réside dans la qualité des données, des documents, de la documentation, des métadonnées, du code et de tout ce qui peut composer la base de connaissance de notre système.

Il existe d'innombrables manières de construire un RAG : quelles étapes, quels prompts, quel type de recherche, quels modèles. Tous ces paramètres font que construire un RAG opérationnel nécessite une connaissance approfondie du domaine et des données et une grande quantité d'essais sur lesquels on itère pour tenter d'arriver à un résultat qui semble convenir.

9.3. Fiabilité

Il est assez difficile de tester la pertinence des réponses d'un LLM, surtout lorsqu'il s'agit de texte. Il faut d'ailleurs utiliser un LLM pour réaliser ces évaluations.

La recherche sémantique est plus facile à tester, tout comme la fiabilité d'un modèle de type Text2SQL qui permet d'interroger un jeu de données géographiques.

9.3.1. Estimer la performances du RAG

Globalement, tester la fiabilité d'un RAG est une tâche complexe, mais qu'il ne faut surtout pas négliger. Ces tests de fiabilité doivent être anticipés et utilisés tout au long de la réalisation du RAG pour l'affiner au fur et à mesure des itérations. Est-ce que les modifications que l'on réalise au cours du développement améliorent les résultats ou pas ? A quel moment peut-on juger que le système est opérationnel ?

9.3.2. Sensibilité du prompt engineering

Un LLM est non déterministe et peut fournir des réponses différentes si l'on pose plusieurs fois la même question. Élaborer les prompts, pour chaque phase du retrieval (extraction, reformulation etc...) est très délicat. Si l'on change un mot, une tournure, si l'on découpe nos appels, bref à chaque petit changement, les résultats peuvent varier significativement.

Développer un RAG permet rapidement d'avoir des résultats moyens, qui fonctionnent sur un cas spécifique. Mais avoir un RAG opérationnel, avec un taux de fiabilité conséquent est un travail extrêmement difficile, chaque petit gain de pertinence s'obtient au coût de nombreux efforts.

9.3.3. Erreurs de génération

Les LLM ne sont pas toujours capables de distinguer les données fiables des informations erronées, surtout lorsqu'ils traitent de grandes quantités de données hétérogènes. Par exemple, en matière de géospatial, un LLM pourrait mal interpréter une source de données ou fusionner des informations incompatibles entre elles :

- Les LLM peuvent donner des résultats peu fiables lorsqu'ils doivent interpréter des données géospatiales complexes, comme des cartes générées à partir de plusieurs couches de données.
- Il est difficile pour un LLM de toujours "savoir" quand il devrait dire "je ne sais pas", augmentant le risque de fournir des résultats incorrects.

9.3.4. Manque de données à jour et de connaissances contextuelles

Les LLM, tels que GPT-4 ou Llama, sont souvent basés sur des données figées à un moment précis. Cela signifie qu'ils ne disposent pas d'informations en temps réel sur les données géospatiales qui évoluent rapidement. Cela pose des problèmes dans plusieurs cas d'usage :

- **Données dynamiques** : Les données géospatiales, en particulier celles liées à des phénomènes en temps réel comme la météo, le trafic ou les catastrophes naturelles, sont souvent sujettes à des changements rapides. Les LLM entraînés sur des données statiques ne peuvent pas refléter ces changements, rendant leurs réponses obsolètes ou inexactes dans ces contextes.
- **Inexactitude des informations locales** : Les LLM peuvent générer des réponses basées sur des données d'entraînement anciennes ou incomplètes, conduisant à des erreurs sur des informations géospatiales comme les routes, les constructions récentes, ou les changements dans l'infrastructure.

9.4. Capacités

9.4.1. Taille du contexte

Les données géographiques peuvent être très volumineuses et ne peuvent pas être passées en totalité dans le contexte du LLM qui est limité. Une phase de génération à laquelle on envoie un jeu de données pour que le LLM en fasse l'analyse est donc bien à mesurer.

Il faut parfois plutôt envisager des modèles générant les requêtes SQL qui seront appelées pour générer les statistiques désirées soit même, au lieu de demander au LLM de le faire.

9.4.2. Requêtes complexes

Un LLM peut facilement extraire un graphique d'évolution d'une métrique d'un jeu de données, par contre, si l'on veut poser une question généraliste et que le LLM déduise seul les jointures à réaliser, les colonnes à utiliser, les agrégations à composer, les résultats seront beaucoup plus mitigés.

De plus, un des gros problèmes des LLM sur ce genre de sujet est l'itération. Il est très difficile de corriger la réponse d'un LLM. Il ne va pas réellement corriger sa réponse, mais en générer une autre qui peut être complètement différente (ce problème s'illustre très bien sur la génération d'images). De ce fait, l'itération pour atteindre la bonne requête SQL, complexe et fonctionnelle ne portera pas forcément ses fruits.

9.4.3. Modèles multimodaux

L'IA générative a montré d'excellents résultats en matière de génération de texte, d'image ou de vidéo. Mais ils comportent de grandes lacunes en matière de compréhension ou de génération de cartes.

Les LLM ne peuvent pas encore comprendre une carte fournie sous forme d'image. Ils ne sont pas non plus capables de générer une carte fiable à partir des données issues de la phase de retrieval.

9.4.4. Données géospatiales

La plupart des LLM sont entraînés sur de grandes quantités de texte généraliste, souvent dérivées du web, et ne sont pas intrinsèquement spécialisés dans les données géospatiales. Cela limite leur capacité à comprendre et à manipuler directement les données géographiques complexes sans une étape d'adaptation ou d'affinage. Par exemple :

- Les LLM peuvent comprendre des concepts simples comme des adresses ou des lieux connus, mais lorsqu'il s'agit de traiter des données vectorielles, de comprendre les projections cartographiques ou de manipuler des formats spécifiques (comme le GeoJSON, Shapefile ou PostGIS), leur compréhension est limitée.
- Le manque de connaissance des standards géospatiaux, tels que les spécifications de l'OGC (Open Geospatial Consortium), complique l'intégration des LLM dans des systèmes SIG existants sans recours à des processus supplémentaires comme le RAG.

9.4.5. Difficultés avec la précision géométrique et les calculs spatiaux

La manipulation précise des données géométriques est essentielle dans de nombreux systèmes géospatiaux. Cependant, les LLM ne sont pas conçus pour effectuer des calculs spatiaux avancés tels que l'intersection, la mesure de distances, ou le traitement de topologies complexes. Ils sont inefficaces pour :

- Calculer des **distances** précises entre des points sur des projections spécifiques (WGS84, Mercator, etc.).
- Gérer des **jointures spatiales** et des requêtes géographiques complexes, telles que la combinaison de multiples couches SIG ou l'utilisation de **PostGIS** pour exécuter des requêtes spécifiques.

9.4.6. Hallucinations

Un problème bien documenté des LLM est leur tendance à générer des réponses incorrectes ou trompeuses avec une grande confiance. Dans le domaine géospatial, ces "hallucinations" peuvent avoir des conséquences graves :

- **Informations géographiques fausses** : Un LLM pourrait inventer une localisation, une adresse ou des informations géographiques erronées qui n'existent pas dans les données réelles. Cela peut poser des problèmes dans des applications critiques comme la planification urbaine, les interventions d'urgence ou les systèmes de navigation.
- **Récupération incorrecte des données** : Même avec des architectures RAG pour extraire des données spécifiques, un LLM peut mal comprendre le contexte ou sélectionner des informations incorrectes pour générer une réponse, ce qui peut entraîner des erreurs.

9.5. Ethique et confidentialité

9.5.1. Confidentialité des données

Les LLM, en particulier ceux qui utilisent des modèles propriétaires hébergés dans le cloud, soulèvent des préoccupations majeures en termes de **protection de la vie privée** et de confidentialité des données géospatiales. Les utilisateurs doivent envoyer leurs données à des serveurs distants pour traitement, ce qui soulève plusieurs risques :

- **Fuites de données sensibles** : Les organisations qui manipulent des données géospatiales sensibles, telles que les infrastructures critiques, les réseaux d'eau, ou les installations militaires, peuvent être réticentes à utiliser des LLM si cela implique de partager ces données avec des tiers.
- **Conformité légale** : Les exigences de la RGPD (Règlement Général sur la Protection des Données) en Europe et d'autres lois sur la confidentialité imposent des limites strictes à la manière dont les données géographiques peuvent être traitées et stockées.

9.5.2. Biais géographiques

Les LLM sont aussi sujets à des biais dans les données sur lesquelles ils sont entraînés. Cela peut entraîner des biais géographiques :

- **Favoritisme des régions bien couvertes** : Les LLM peuvent être plus performants dans les régions où les données sont abondantes (comme les grandes villes) et moins fiables dans les zones rurales ou sous-représentées dans les jeux de données d'entraînement.
- **Erreurs culturelles et linguistiques** : Les différences dans la manière de nommer des lieux, des régions ou des infrastructures dans différentes cultures et langues peuvent conduire à des incompréhensions ou des généralisations incorrectes.

D'une manière générale, un grand modèle de langage présente des performances extraordinaires pour analyser du texte et répondre aux questions d'ordre général. Au-delà des problèmes d'éthique, de sécurité ou de mise en œuvre, il faut comprendre que ces modèles ingèrent mal l'information géographique non textuelle, ne sont pas à l'aise avec les coordonnées géographiques, les projections ou les opérations spatiales. Pour en tirer le meilleur parti, les LLM doivent être utilisés uniquement dans des phases de recherche ou de génération. Toutes les autres phases, appel à des APIs spécifiques, génération de requêtes SQL, traitements géospatiaux doivent être traités via des compétences pures géospatiales, en appui de modèles spécifiques au traitement d'une tâche plus simple.

Le RAG est globalement la meilleure alternative permettant de combiner le potentiel du LLM, avec l'expertise du géospatial.

Pour en savoir plus

- Retour d'expérience d'utilisation des LLM

10. Conclusions

L'arrivée de ChatGPT et des LLM bouleverse la société moderne. Certains y voient une menace, d'autres une opportunité extraordinaire de simplifier les tâches rébarbatives pour que les individus puissent se concentrer sur des fonctions à plus haute valeur ajoutée.

Les premières applications d'IA génératives sont impressionnantes, du chat en 2022, puis de la génération d'images ou de vidéos, maintenant une capacité à avoir un dialogue jonché d'émotions avec un humain, et quoi pour demain ? C'est une technologie en ébullition qui touche tous les secteurs de notre société.

Les acteurs du géospatial se sont engouffrés dans cet élan d'innovation et rêvent à proposer de nouveaux services basés sur les capacités de l'IA. Piloter une application par le dialogue, résoudre des problèmes sociétaux ou territoriaux de façon automatisée, traiter de grands volumes de données, en extraire de la valeur non soupçonnée, remplir automatiquement un catalogue de métadonnées, sont tous des exemples matérialisant les ambitions d'appliquer les LLM au monde du géospatial.

Il faut cependant rester très prudent, l'IA impressionne, mais elle doit prouver son utilité. Peu de produits sont en production, peu de startups sont rentables. C'est un milieu foisonnant certes, mais qui connaîtra probablement une grande crise dans les années à venir. Le parcours vers l'intégration des LLM doit être mesuré, réfléchi et prudent si l'on souhaite en tirer une réelle plus-value.

Certes, les grands acteurs du géospatial comme Google, Apple, Mapbox ou CARTO ont mis en place des systèmes en production pilotés par le langage naturel. Mais ces systèmes sont propriétaires, et très peu d'initiatives open source proposent aujourd'hui le même potentiel. L'utilisation des LLM révèle de nombreuses contraintes et difficultés, tant commerciales, éthiques ou techniques. Elle nécessite une grande maîtrise et compréhension du domaine, une veille technique accrue, autant de challenges qu'il faudra résoudre pour se faire une place dans cette jungle, tant de défis à résoudre pour que ces technologies deviennent pleinement opérationnelles et accessibles aux problématiques des plateformes de géodonnées.