

Formation RAG et fine-tuning des LLM : choisir, combiner, évaluer

Réf. SF34733

Apprenez à choisir entre RAG et fine-tuning, à les combiner et à prouver par la mesure lequel sert le mieux votre cas d'usage.

3 j
Durée

2 400 € HT
Inter / participant

0
Sessions à venir

Objectifs pédagogiques

- ✓ Choisir entre prompting, RAG, fine-tuning ou leur combinaison selon le besoin, les données disponibles et les contraintes de coût, de latence et de confidentialité
- ✓ Implémenter le fine-tuning d'un LLM open source avec LoRA et QLoRA sur un jeu de données métier
- ✓ Implémenter un pipeline RAG complet, de l'ingestion des documents à la génération de réponses sourcées
- ✓ Évaluer un système RAG et un modèle fine-tuné avec un jeu de test métier, des métriques adaptées et un LLM juge
- ✓ Optimiser la qualité de la recherche d'un RAG (découpage, recherche hybride, reranking, filtres de métadonnées) à partir de mesures
- ✓ Concevoir une architecture combinant RAG et fine-tuning et comparer ses performances à celles de chaque approche seule

Public visé & prérequis

Public visé Architectes techniques, Data scientists, Développeurs

Prérequis- Pratique courante de Python, notamment en notebooks - Notions d'apprentissage automatique : jeux d'entraînement, de validation et de test, surapprentissage, métriques d'évaluation - Première expérience d'utilisation d'un LLM, par API ou en open source

Programme détaillé

Adapter un LLM à son contexte : prompting, RAG ou fine-tuning

- Limites d'un LLM généraliste : connaissance figée, hallucinations, absence d'accès aux données internes
- Leviers d'adaptation : prompting, contexte long, RAG et fine-tuning, et ce que chacun modifie (les connaissances fournies au modèle ou son comportement)
- Apports et limites du fine-tuning : format, ton et tâches spécialisées, mais pas une connaissance factuelle fiable et à jour
- Critères de décision : nature du besoin, volume et fraîcheur des données, coût, latence, confidentialité et traçabilité des réponses
- Modèle open source ou API : choisir selon la tâche, l'hébergement visé et le budget GPU
- Mesure de référence : évaluer le modèle seul sur un premier jeu de questions métier avant toute adaptation

Activités pratiques :

- Mise en place de l'environnement et test d'un modèle open source et d'un modèle par API sur le cas métier fil rouge
- Repérage des questions qui échouent sans accès aux documents
- Application d'une grille de décision à plusieurs cas d'usage (assistant support, rédaction normée, classification)

Construire un pipeline RAG

- Architecture d'un RAG : indexation des documents en amont, puis recherche et génération au moment de la requête
- Ingestion des documents : extraction depuis PDF, Word, pages web et données structurées, nettoyage et ajout de métadonnées
- Stratégies de découpage (chunking) : taille, chevauchement, découpage selon la structure du document et effet sur la pertinence
- Modèles d'embedding : choix selon la langue, le domaine, la dimension des vecteurs et le coût
- Bases vectorielles : FAISS, ChromaDB, Qdrant ou pgvector, critères de choix et indexation
- Génération augmentée : prompt de génération, injection des passages, citation des sources et refus de répondre hors du corpus
- Outillage : framework d'orchestration (LangChain, LlamaIndex) ou code Python direct selon le contexte du projet

Activités pratiques :

- Construction d'un assistant RAG sur une base documentaire interne (procédures, FAQ, tickets)

- Comparaison de deux stratégies de découpage sur les mêmes questions

Améliorer la qualité de la recherche

- Diagnostic des échecs : distinguer un défaut de recherche (mauvais passages retrouvés) d'un défaut de génération
- Recherche hybride : combiner recherche lexicale et vectorielle pour les termes métier, les codes et les acronymes
- Reranking : réordonner les passages candidats avec un modèle dédié pour gagner en précision
- Filtres de métadonnées : restreindre la recherche par source, date, périmètre ou droits d'accès
- Transformation des requêtes : reformulation et décomposition des questions complexes
- Réglages et compromis : nombre de passages, taille du contexte, latence et coût

Activités pratiques :

- Ajout d'une recherche hybride et d'un reranker au pipeline, puis mesure du gain sur le jeu de questions
- Analyse d'erreurs sur les questions mal traitées et correction ciblée

Évaluer un système RAG

- Jeu de test métier : questions, réponses attendues et passages de référence représentatifs des usages réels
- Données de test synthétiques : générer des questions à partir du corpus et les faire valider par un expert métier
- Métriques de recherche : rappel et précision des passages retrouvés, rang du premier passage pertinent
- Métriques de réponse : fidélité aux sources, pertinence et exactitude par rapport à la réponse attendue
- LLM juge : protocole, grille de notation, biais connus et contrôle par échantillonnage humain
- Outils d'évaluation open source : automatiser les mesures et suivre les résultats d'une itération à l'autre

Activités pratiques :

- Constitution d'un jeu de test mixte, rédigé à la main et généré automatiquement
- Évaluation automatisée du pipeline et comparaison avant et après l'optimisation de la recherche

Fine-tuner un LLM avec LoRA et QLoRA

- Fine-tuning complet ou PEFT : pourquoi les méthodes qui n'entraînent qu'une petite partie des paramètres sont devenues la norme
- LoRA et QLoRA : principe des adaptateurs, quantification et réduction de l'empreinte GPU
- Préparation du jeu de données : format instruction / réponse (JSONL, datasets Hugging Face), nettoyage, déduplication, découpage entraînement / validation / test
- Volume et qualité des données : nombre d'exemples nécessaire, fuites de données et exemples contradictoires à éviter
- Entraînement supervisé avec l'écosystème Hugging Face : hyperparamètres essentiels et lecture des courbes de perte
- Surapprentissage et régressions : détecter la perte de capacités générales du modèle adapté
- Fine-tuning via l'API d'un fournisseur : alternative managée, coûts et contraintes sur les données

Activités pratiques :

- Entraînement d'un adaptateur QLoRA sur un jeu de données métier
- Comparaison du modèle de base et du modèle adapté sur le jeu de test

Combiner RAG et fine-tuning

- Répartition des rôles : le RAG apporte la connaissance à jour, le fine-tuning le format, le ton et le respect des consignes
- Fine-tuning du générateur pour le RAG : apprendre au modèle à exploiter les passages fournis et à écarter les passages non pertinents
- Adaptation du modèle d'embedding au domaine : améliorer la recherche sur un vocabulaire métier spécifique
- Évaluation comparative : modèle seul, RAG, modèle fine-tuné et combinaison mesurés sur le même jeu de test
- Lecture des résultats : qualité, coût, latence et effort de maintenance de chaque configuration
- Recommandation d'architecture : formaliser et argumenter le choix pour un cas d'usage donné

Activités pratiques :

- Construction d'un pipeline associant le modèle fine-tuné et le RAG
- Tableau comparatif chiffré des configurations et rédaction d'une recommandation argumentée

Mettre en production et faire vivre la solution

- Choix d'hébergement : API managée ou modèle auto-hébergé, effet sur le coût, la confidentialité et l'exploitation
- Servir un modèle adapté : chargement ou fusion de l'adaptateur LoRA, quantification pour l'inférence
- Observabilité : traces des requêtes et des passages retrouvés, coûts, latence et retours utilisateurs
- Mise à jour : réindexer les documents ou réentraîner le modèle, tests de non-régression

- Sécurité et conformité : injection de prompt via les documents, droits d'accès au corpus, données personnelles dans les corpus et les jeux d'entraînement
- Ouverture : RAG agentique et RAG sur graphe de connaissances, repères pour aller plus loin

Activités pratiques :

- Mise en place d'un suivi des requêtes et d'un test de non-régression sur le jeu de test
- Analyse des risques d'une solution RAG : injection, fuite de données, droits d'accès

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

4,9/5 — 65 avis

Rachelle Y. — Merci c'était super :)

Anthony G. — Formation très bien organisée, très claire Alison explique très clairement et est très à l'écoute, avec toujours une approche pratique essentielle. je n'avais jamais fait de Python et...

Ils nous font confiance

Cette année notre filière Intelligence artificielle & Data Science a accompagné plus de 250 entreprises dont :

SOPRA STERIA CEGEDIM RTE AKANEA AGRO SOFTWARE AKANEA DEVELOPPEMENT AMADEUS SAS GROUPE LDLC LBC FRANCE PROXIAD AXE SEINE SNCF - E.SNCF SOLUTIONS +240 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com