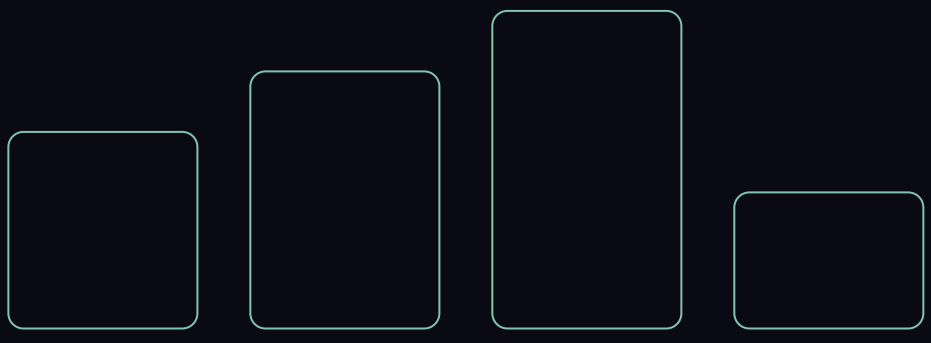
Édition Future.

Comprendre les LLM et travailler avec ses sources

Choisir un outil et vérifier les informations de ses documents.



GUIDE PRATIQUE

EXEMPLES ET EXERCICES CORRIGÉS

Votre parcours

Lisez les repères, puis réalisez les exercices avec vos propres données non confidentielles.

Avant de commencer

Ce cours s’adresse aux personnes qui utilisent déjà une conversation avec une IA et veulent comprendre ses limites. Vous n’avez pas besoin de programmer. Préparez deux petits documents non confidentiels et une question dont vous connaissez la réponse. Le projet final consiste à comparer deux offres fictives sans confondre chiffres, conditions et informations manquantes.

1. Distinguer le modèle et l’application

Un grand modèle de langue, ou LLM, transforme une entrée en une suite de tokens. Un token est une unité manipulée par le modèle : morceau de mot, mot, ponctuation ou autre élément selon le système. L’application qui l’entoure peut ajouter une interface, une recherche, des fichiers, une mémoire enregistrée et des outils. Les capacités visibles résultent de cet ensemble.

Cette distinction évite une confusion fréquente : une réponse accompagnée d’une recherche web n’est pas seulement issue des paramètres du modèle. Inversement, un texte très actuel en apparence ne prouve pas qu’une recherche a eu lieu. Demandez la source et ouvrez-la.

Les modèles peuvent réussir des tâches de raisonnement et échouer sur d’autres. Leur aisance verbale ne garantit ni exactitude ni compréhension de votre situation. Pour décider s’ils conviennent, observez leurs résultats sur les tâches qui vous intéressent plutôt que d’attribuer à tous les modèles une capacité ou une incapacité absolue.

Exemple fictif. Nora demande le délai de livraison de son fournisseur. Sans document ni accès au fournisseur, l’assistant n’a pas de base suffisante pour affirmer un délai. Avec un devis daté, il peut chercher la clause. Avec un outil de suivi de commande autorisé, l’application peut consulter un état plus récent. Ces trois situations n’offrent pas la même preuve.

2. Tokens, entraînement et inférence

Pendant l’entraînement, les paramètres du modèle sont ajustés à partir de données et d’objectifs d’apprentissage. Pendant l’inférence, le modèle utilise ses paramètres et le contexte de la demande pour produire une sortie. Fournir trois exemples dans une conversation guide cette sortie ; cela ne correspond pas, à lui seul, à réentraîner les paramètres.

Une mémoire enregistrée par l’application est encore autre chose : elle peut réintroduire des informations dans une demande ultérieure. De même, une politique fournisseur peut prévoir une utilisation ultérieure de certaines données pour améliorer ses services. Ne confondez pas ces mécanismes avec l’apprentissage instantané du modèle à chaque phrase.

Le nombre de tokens d’un texte dépend du tokenizer utilisé. Sans tokenizer identifié et exécuté, annoncer qu’un mot français vaut exactement quatre tokens serait trompeur. Pour préparer un budget d’API, utilisez le compteur de l’outil et les consommations réellement remontées.

Calcul pédagogique, avec tarifs entièrement fictifs. Si une API coûtait 1 € par million de tokens d’entrée et 4 € par million de tokens de sortie, une demande de 3 000 tokens d’entrée et 500 de sortie coûterait 0,003 € + 0,002 €, soit 0,005 €. Ce n’est le tarif d’aucun service cité. Ajoutez ensuite les autres frais éventuels : outils, recherche, stockage, nouvelle tentative. Un abonnement de conversation et une facturation API sont des offres différentes.