Engineer

Génération augmentée par recherche

Réponses générées avec des documents retrouvés par question

Ce que c’est

La génération augmentée par recherche répond à une question avec un modèle de langage en deux étapes. L’application cherche d’abord dans un ensemble de documents les passages les plus susceptibles de contenir la réponse. Elle les place ensuite dans l’entrée du modèle, à côté de la question, et celui-ci rédige sa réponse à partir de ces passages plutôt que de ce qu’il a absorbé pendant son entraînement. Le nom, généralement abrégé en RAG, vient d’un article de recherche de 2020 sur des tâches langagières exigeant des connaissances, comme la réponse à des questions ouvertes.

Cet article décrivait deux types de mémoire. Un modèle préentraîné conserve ce qu’il a appris dans ses paramètres, ce que les auteurs appelaient mémoire paramétrique. Un index de documents distinct, interrogé par un moteur de recherche entraîné, constitue la mémoire non paramétrique. Leur système associait un modèle séquence à séquence préentraîné à un index vectoriel dense de Wikipédia. Ils comparaient deux façons d’exploiter les résultats : l’une conditionne toute la réponse sur les mêmes passages retrouvés, l’autre peut utiliser des passages différents pour chaque token écrit. Le système a fait progresser l’état de l’art sur trois bancs d’essai de réponse à des questions en domaine ouvert. Les auteurs ont trouvé les textes produits plus spécifiques, plus variés et plus factuels que ceux d’un modèle comparable sans recherche documentaire.

Une application peut assembler le même schéma avec des éléments qu’elle possède déjà : un index de recherche sur ses propres documents et un grand modèle de langage appelé par API. La partie recherche est une recherche ordinaire, dont les deux familles apparaissent ici. La recherche éparse fait correspondre les mots, comme la recherche plein texte classe les documents avec une fonction de score telle que BM25. La recherche dense transforme la question et chaque passage en vecteurs de nombres puis renvoie les passages

aux vecteurs les plus proches : c'est la recherche vectorielle. Un article de 2020 sur la recherche dense de passages a constaté qu'un moteur dense entraîné sur un nombre modeste de paires question-passage dépassait une solide référence BM25 de 9 à 19 points de pourcentage sur la fréquence de présence du bon passage parmi les vingt premiers résultats. Un système peut exécuter les deux et fusionner leurs listes.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est la génération augmentée par recherche ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des données accessibles aux agents d'IA, une seule fiche par client, des accès limités au nécessaire et chaque réponse reliée à sa source.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Pipeline géographique accéléré de 50 fois — 50x d'accélération des pipelines de données géographiques

Pipeline géographique accéléré de 50 fois Contactez-nous →

Pourquoi c'est important

Un modèle de langage ne connaît que ce qui figurait dans ses textes d'entraînement. Ces textes s'arrêtent à une date fixe et n'ont jamais inclus vos contrats, vos tarifs ou le compte rendu de la semaine dernière. L'article RAG nomme deux limites des connaissances conservées dans les paramètres : il est difficile de montrer d'où vient une réponse et de mettre ces connaissances à jour. La recherche documentaire répond aux deux. La réponse peut citer les passages reçus, permettant au lecteur d'ouvrir la source pour vérifier. Les nouvelles connaissances atteignent le modèle dès qu'un document entre dans l'index, sans aucun entraînement.

Cela remet aussi la question des droits de lecture à sa juste place. La recherche documentaire est une recherche qui peut s'exécuter avec les droits de la personne posant la question. Le modèle ne voit alors que les passages que cette personne aurait pu ouvrir elle-même. Un modèle entraîné sur ces mêmes documents n'a pas cette frontière : tout ce qu'il a appris est accessible à tous ceux qui l'utilisent.

Le coût est que la réponse ne peut être meilleure que ce que renvoie la recherche. Un modèle recevant les mauvais passages en tire une réponse fluide, et un modèle ne recevant rien d'utile peut malgré tout répondre de mémoire.

L’essentiel du travail d’un système de génération augmentée par recherche porte donc sur les données et la recherche plutôt que sur le modèle.

À quoi cela ressemble en pratique

La trace ci-dessous suit une question dans un système de génération augmentée par recherche. La recherche s’exécute avec les permissions de l’utilisateur qui pose la question, chaque passage arrive avec son fichier et sa page d’origine, et les instructions demandent au modèle de répondre uniquement à partir des passages :

```
question: Quelles conditions de livraison s’appliquent à la commande 4471 ?
utilisateur: sales-3 (rôle: sales, région: north)

passages retrouvés, filtrés selon les droits de sales-3:
  [1] contracts/framework-2025.pdf, page 4
      "Les marchandises sont livrées à l’entrepôt de l’acheteur,
       sauf indication contraire dans la commande."
  [2] orders/4471.txt, ligne 12
      "Conditions de livraison: selon l’accord-cadre."

instructions:
  Répondez uniquement à partir des passages ci-dessous et citez-les par [n].
  S’ils ne répondent pas à la question, indiquez-le.

réponse:
  La commande 4471 renvoie à l’accord-cadre [2], selon lequel
  les marchandises sont livrées à l’entrepôt de l’acheteur [1].
```

Chaque partie de cette trace est une donnée vérifiable par le système. Un test peut confirmer que les bons passages ont été retrouvés pour une question connue. Un script peut vérifier que chaque `[n]` de la réponse renvoie à un passage effectivement retrouvé, et une personne lisant la réponse peut ouvrir la page 4 du contrat.

Les fondations nécessaires font l’objet de la page sur les données d’entreprise prêtes pour les agents IA. Elle décrit la transformation d’anciens PDF, de scans et d’archives de courrier en texte interrogeable, avec sa source conservée à côté, et l’origine gardée pour chaque ligne : système, chargement, fichier et page. Un agent peut ainsi citer l’enregistrement à l’origine d’un chiffre et une personne peut l’ouvrir. La page place aussi les règles dans les données plutôt que dans le

prompt : l'agent agit avec les privilèges de son utilisateur, jamais ceux d'un administrateur, et chaque lecture est journalisée avec la personne pour laquelle elle a été effectuée.

Où les problèmes surviennent

La recherche manque la cible et le modèle répond malgré tout. Quand le bon passage n'est pas retrouvé, le modèle reçoit quand même quelque chose et rédige quand même une réponse. Mesurez la recherche séparément : notez des questions réelles avec les passages qui y répondent et vérifiez leur fréquence de présence dans les résultats fournis au modèle. Demandez au modèle de le signaler lorsque les passages ne répondent pas à la question et incluez de telles questions dans vos tests.

Des passages qui coupent la réponse en deux. Découper les documents en morceaux de longueur fixe sépare un tableau de son en-tête et une clause de son exception. Chaque moitié retrouvée seule induit en erreur. Découpez selon la structure existante du document, ses sections, ses pages et ses lignes de tableau, en gardant son titre et son emplacement avec chaque morceau.

L'index ignore qui pose la question. Un index construit par un service capable de tout lire renvoie des passages de documents que la personne interrogeant le système n'a pas le droit d'ouvrir, et le modèle les lui résume. Filtrez selon les permissions de l'utilisateur à l'intérieur de la recherche, dans l'index ou la base de données, avant que quoi que ce soit n'atteigne le modèle. Une règle dans le prompt n'est pas un contrôle d'accès.

Des instructions cachées dans un document. Le texte retrouvé a été écrit par quelqu'un d'autre, et le modèle ne distingue pas de façon fiable un passage d'une instruction. Le profil du NIST pour l'IA générative décrit l'injection indirecte de prompt comme des attaques contre les applications fondées sur des modèles de langage, par l'intermédiaire de données susceptibles d'être retrouvées. Un article de 2023 a démontré de telles attaques sur des systèmes réels. Traitez chaque passage retrouvé comme une entrée non fiable, ne laissez aucun passage déclencher une action sans contrôle dans le code et ne donnez aux outils appelables par le modèle aucun droit supérieur à ceux de l'utilisateur.

Un index en retard. Un document est modifié ou supprimé, mais l'index garde ses anciens passages : le système répond donc avec assurance à partir d'un contrat devenu inapplicable. Actualisez l'index à chaque changement de source,

retirez les passages quand leur document est supprimé et affichez la date de chaque source à côté de la réponse.

Des citations qui ne disent pas ce qu'affirme la réponse. Un modèle peut citer le passage [2] pour une phrase que le passage [2] n'étaye pas. Le NIST relève que les citations générées peuvent elles-mêmes être inventées. Vérifiez dans le code que chaque citation renvoie à un passage retrouvé et faites comparer par des personnes un échantillon de réponses à leurs passages cités lors des tests réguliers.

Pour aller plus loin

- Génération augmentée par recherche pour les tâches de traitement du langage exigeant des connaissances — l'article de 2020 qui a présenté les modèles RAG et comparé leurs deux formulations pour la réponse aux questions et la génération.
- Recherche dense de passages pour répondre à des questions en domaine ouvert — un moteur dense entraîné sur des paires question-passage, comparé à BM25 pour trouver le passage contenant la réponse.
- NIST AI 600-1 : profil de l'intelligence artificielle générative — profil de risques du NIST pour l'IA générative, couvrant les inventions, la confidentialité des données et l'injection indirecte de prompt.
- Ce à quoi vous n'avez pas souscrit : compromettre des applications réelles intégrant des modèles de langage par injection indirecte de prompt — comment des instructions cachées dans les données retrouvées prennent le contrôle d'une application, avec des attaques sur des systèmes réels.

Glossaire

Choisissez selon vos contraintes réelles

Présentez un jeu de données représentatif, les questions auxquelles il doit répondre et le goulot d'étranglement actuel. Nous pouvons convenir de la manière de mesurer la qualité, le débit et la reprise.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est la génération augmentée par recherche ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/retrieval-augmented-generation/