Engineer

Grand modèle de langage

Réseau neuronal entraîné à prédire le jeton de texte suivant

Ce que c’est

Un grand modèle de langage est un réseau neuronal entraîné sur un très vaste corpus de textes pour faire une chose : à partir du texte déjà présent, attribuer une probabilité à chaque jeton susceptible de suivre. Un jeton est un court fragment de texte, souvent un mot entier, parfois une partie de mot ou un simple signe de ponctuation. Écrire une réponse consiste à répéter cette étape en boucle. Le logiciel choisit un jeton selon ces probabilités, l’ajoute au texte, puis recommence jusqu’à ce que le modèle produise un jeton d’arrêt ou qu’une limite de longueur soit atteinte. Le profil du NIST consacré à l’IA générative, publié en juillet 2024, le résume en une phrase : « Les LLM prédisent le jeton ou le mot suivant dans une phrase ou une expression. » L’abréviation anglaise LLM, pour large language model, est aussi celle qu’emploie le profil.

L’architecture des modèles de ces dernières années est le Transformer, décrit dans un article de 2017 sur la traduction automatique. Il a abandonné les couches récurrentes et convolutives des modèles précédents pour reposer uniquement sur l’attention, mécanisme qui permet à chaque position du texte de consulter directement toutes les autres. Cela a beaucoup facilité la répartition de l’entraînement entre de nombreux processeurs, et l’entraînement du modèle présenté dans l’article a pris bien moins de temps que celui de ses prédécesseurs.

« Grand » renvoie au nombre de paramètres, les poids que le réseau apprend pendant l’entraînement. Un article de 2020 décrivait un modèle en possédant 175 milliards, dix fois plus que tout modèle de langage dense antérieur. Il montrait qu’à cette taille, un modèle pouvait accomplir une nouvelle tâche à partir de quelques exemples résolus placés dans son entrée, sans modifier aucun de ses poids.

Un modèle entraîné uniquement à poursuivre un texte fait exactement cela : il poursuit le texte, ce qui diffère de l'exécution d'une demande. Un article de 2022 décrivait l'étape qui transforme un tel modèle en assistant. Des personnes ont rédigé les réponses souhaitées à un ensemble de demandes, et le modèle a été affiné sur ces réponses ; elles ont ensuite classé les réponses du modèle lui-même, qui a été entraîné davantage à partir de ces classements, selon une méthode appelée apprentissage par renforcement à partir de retours humains. Dans les évaluations de l'article, les personnes préféraient les réponses d'un modèle de 1,3 milliard de paramètres entraîné ainsi à celles du modèle original, qui en possédait cent fois plus.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un grand modèle de langage ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Un agent connecté à vos systèmes, testé sur des cas rédigés par vos experts, supervisé par votre équipe et indépendant d'un modèle unique.

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. Moteur de 268 contrôles pour le dépôt — 268 contrôles de commit et 146 tests des contrôles eux-mêmes

Moteur de 268 contrôles pour le dépôt Contactez-nous →

Pourquoi c'est important

Un modèle de langage prend en charge des tâches difficiles à couvrir par des règles fixes : classer un courriel librement rédigé dans le bon dossier, extraire les champs d'un bon de livraison numérisé, rédiger un projet de réponse dans la langue du client, résumer un long fil de discussion. Un travail qui exigeait auparavant qu'une personne lise chaque élément peut désormais être préparé par un logiciel, puis vérifié par une personne.

La propriété qui le rend utile complique aussi la construction de logiciels qui s'appuient sur lui. Sa sortie est généralement échantillonnée : une même entrée peut donc donner des réponses différentes d'une exécution à l'autre, et le texte reste fluide, qu'il soit juste ou faux. Le NIST appelle confabulation, plus connue sous le nom d'hallucination, une sortie erronée affirmée avec assurance, et la décrit comme une conséquence naturelle de la construction de ces modèles : ils produisent du texte conforme aux statistiques des textes d'entraînement, souvent exact et parfois faux. Le NIST précise aussi qu'un modèle peut inventer le raisonnement ou les références qui semblent justifier une réponse. Une

réponse fausse a exactement l'apparence d'une réponse juste : le seul moyen de savoir à quelle fréquence un modèle répond correctement pour vos tâches est donc de l'évaluer sur ces mêmes tâches.

Le choix ne se résume pas non plus à s'engager une fois pour toutes sur un modèle. De nouveaux modèles paraissent régulièrement, deux d'entre eux peuvent se comporter très différemment sur une même entrée, et le fournisseur du modèle appelé par une application décide quand il change. Une équipe qui s'appuie sur un modèle de langage construit avec un composant qu'elle ne contrôle pas. Le travail durable est tout ce qui l'entoure : les cas qui le mesurent, les outils qu'il peut appeler et les règles de ce qu'il peut faire seul.

Le profil du NIST nomme douze domaines de risque pour l'IA générative, dont la confabulation, la confidentialité des données, la sécurité de l'information et la propriété intellectuelle. Il s'adresse aux organisations qui gèrent les risques plutôt qu'aux développeurs, mais fournit une liste de contrôle utilisable pour décider à quoi une fonctionnalité fondée sur un modèle peut accéder.

À quoi cela ressemble en pratique

Une fonctionnalité fondée sur un modèle de langage commence par des cas écrits, avant une consigne. Chaque cas associe une entrée réelle à la réponse qu'accepterait une personne connaissant le travail, conservées dans un fichier versionné à côté du code :

```
# cases/delivery-notes.yaml
- id: note-0142
  input: scans/note-0142.txt
  expect:
    supplier: "Example Freight"
    reference: "DN-58213"
    pallets: 6
- id: note-0157
  input: scans/note-0157.txt
  expect:
    supplier: "Sample Haulage"
    reference: "DN-59002"
    pallets: 14
```

Le choix du modèle pour chaque type de travail est un réglage : un petit modèle rapide peut donc extraire les données, tandis qu'un plus grand rédige les réponses qui exigent du discernement :

```
# routing.yaml
jobs:
  extract-delivery-note:
    model: small-fast
    cases: cases/delivery-notes.yaml
  draft-claim-reply:
    model: large
    cases: cases/claim-replies.yaml
```

Chaque modification est évaluée sur tous les cas avant livraison, qu'elle porte sur une consigne, un outil ou un modèle. Le résultat est un score par cas, conservé dans le temps pour qu'une correction sur l'un ne puisse pas en casser discrètement un autre. Lorsqu'un modèle paraît, l'essayer demande de modifier une ligne de `routing.yaml` et d'exécuter les cas ; la décision de changer repose sur les scores plutôt que sur l'annonce.

C'est ainsi que ce site décrit ses agents d'IA sur mesure. Le choix du modèle relève de la configuration plutôt que du code, avec un petit modèle rapide pour le classement et l'extraction, et un grand pour le discernement. Une mission commence par la rédaction de trente à cinquante exemples réels avec les bonnes réponses, et chaque modification de consigne, d'outil ou de modèle est évaluée sur tous ces exemples avant livraison. La page précise aussi que le modèle reste remplaçable et que le code, les cas et les données appartiennent au client : l'agent reste donc le sien quand le modèle sous-jacent change.

Où les problèmes surviennent

Des réponses assurées mais fausses. Un modèle interrogé sur une clause de contrat qu'il n'a pas sous les yeux peut tout de même en décrire une, avec le ton assuré d'une réponse correcte, et citer une source inexistante. Fournissez les documents sur lesquels il doit fonder ses réponses, demandez-lui de les citer et vérifiez dans le code que chaque référence pointe vers un document et un passage réels. Mesurez le taux d'erreur sur des cas écrits, et faites examiner par une personne les réponses ayant des conséquences avant qu'une action ne s'appuie dessus.

Des instructions cachées dans les données. Une application qui laisse un modèle lire des courriels, des pages web ou des fichiers téléversés laisse les auteurs de ces textes lui parler. Un article de 2023 l'a montré sur des applications réelles : des instructions placées dans du contenu que le modèle risquait de lire lui faisaient divulguer des données et modifier le comportement de l'application. Le profil du NIST cite l'injection indirecte de consignes parmi les risques de sécurité de l'IA générative. Traitez tout ce que le modèle lit comme une entrée non fiable, imposez les permissions dans les outils plutôt que dans la consigne, ne donnez aux outils que les droits de la personne pour qui le modèle agit et exigez l'approbation d'une personne pour toute action irréversible.

Un modèle qui change sous vos pieds. Une nouvelle version ou un autre fournisseur peut répondre autrement à une même entrée : une réponse acceptée le mois dernier échoue aujourd'hui sans modification de votre code. Fixez une version précise du modèle chaque fois que le fournisseur le permet et évaluez-le sur tous les cas du fichier à chaque modification, y compris un changement de modèle, avant que cette modification n'atteigne les utilisateurs.

Une sortie que le code ne peut pas analyser. Un modèle auquel vous demandez du JSON renvoie du JSON suivi d'une phrase aimable, omet un champ ou écrit un nombre sous forme de chaîne. Décrivez la structure attendue dans JSON Schema, vérifiez chaque réponse par rapport à ce schéma et traitez tout échec comme une erreur, à réessayer ou à confier à une personne, jamais comme des données.

Des données personnelles dans les consignes et les journaux. Tout ce qui est envoyé à un modèle hébergé quitte vos systèmes, et un journal de consignes et de réponses conserve les données personnelles longtemps après la fin de la requête. Le NIST ajoute que les modèles peuvent répéter des éléments sensibles de leurs données d'entraînement, problème qu'il appelle mémorisation des données. N'envoyez que les champs nécessaires au travail, soumettez les journaux de consignes aux mêmes règles de conservation et d'accès que les données qu'ils contiennent, et lisez les conditions du fournisseur sur la conservation et l'entraînement avant de lui transmettre des données personnelles.

Pour aller plus loin

- NIST AI 600-1 : profil de l'intelligence artificielle générative — le complément de juillet 2024 au cadre de gestion des risques de l'IA du NIST : douze risques de l'IA générative, dont la confabulation, et des actions proposées pour chacun.
- L'attention suffit — l'article de 2017 qui a présenté le Transformer, l'architecture des modèles de langage actuels.
- Les modèles de langage apprennent à partir de peu d'exemples — l'article de 2020 sur un modèle de 175 milliards de paramètres qui accomplit de nouvelles tâches à partir d'exemples dans son entrée.
- Entraîner les modèles de langage à suivre des instructions grâce aux retours humains — affinement sur des réponses rédigées et des sorties classées, l'étape du prédicteur de texte à l'assistant.
- Ce à quoi vous n'aviez pas souscrit : compromettre des applications réelles intégrant des LLM par injection indirecte de consignes — des instructions placées dans les données lues par un modèle, démontrées sur des applications réelles.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'un grand modèle de langage ? » 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/large-language-model/