Engineer

Großes Sprachmodell

neuronales Netz, an Text zur Folgetoken-Vorhersage trainiert

Was es ist

Ein großes Sprachmodell ist ein neuronales Netz, das an einer sehr großen Textmenge für eine einzige Aufgabe trainiert wurde: Ausgehend vom bisherigen Text weist es jedem Token, das als Nächstes folgen könnte, eine Wahrscheinlichkeit zu. Ein Token ist ein kurzes Textstück, oft ein ganzes Wort, manchmal ein Wortteil oder ein einzelnes Satzzeichen. Eine Antwort zu schreiben bedeutet, diesen einen Schritt in einer Schleife zu wiederholen. Die Software wählt anhand dieser Wahrscheinlichkeiten ein Token aus, hängt es an den Text an und fragt erneut, bis das Modell ein Token erzeugt, das das Ende signalisiert, oder eine Längengrenze erreicht ist. Das im Juli 2024 veröffentlichte NIST-Profil für generative KI fasst es in einem Satz zusammen: „LLMs sagen das nächste Token oder Wort in einem Satz oder einer Wortgruppe voraus." Die Abkürzung LLM, die das Profil selbst verwendet, steht für den englischen Ausdruck Large Language Model und bezeichnet dasselbe.

Die Architektur hinter den Modellen der letzten Jahre ist der Transformer, beschrieben in einer Forschungsarbeit zur maschinellen Übersetzung von 2017. Sie verzichtete auf die rekurrenten Schichten und Faltungsschichten, aus denen frühere Modelle bestanden, und setzte ausschließlich auf Attention: einen Mechanismus, mit dem jede Position im Text unmittelbar auf jede andere zugreifen kann. Dadurch ließ sich das Training wesentlich leichter auf viele Prozessoren verteilen, und das Modell der Forschungsarbeit wurde in viel kürzerer Zeit trainiert als seine Vorgänger.

„Groß" bezieht sich auf die Zahl der Parameter, also der Gewichte, die das Netz beim Training lernt. Eine Forschungsarbeit von 2020 beschrieb ein Modell mit 175 Milliarden Parametern, also zehnmal so vielen, wie jedes frühere nicht dünn besetzte Sprachmodell hatte, und zeigte, dass ein Modell dieser Größe anhand weniger ausgearbeiteter Beispiele in seiner Eingabe eine neue Aufgabe übernehmen konnte, ohne seine Gewichte im Geringsten zu verändern.

Ein Modell, das nur auf das Fortsetzen von Text trainiert wurde, tut genau das: Es setzt den Text fort. Das ist etwas anderes, als die gestellte Aufgabe zu erfüllen. Eine Forschungsarbeit von 2022 beschrieb den Schritt, der ein solches Modell zu einem Assistenten macht. Menschen schrieben die gewünschten Antworten auf eine Reihe von Anfragen, und das Modell wurde anhand dieser Antworten feinabgestimmt. Anschließend brachten Menschen die Antworten des Modells selbst in eine Rangfolge, und es wurde anhand dieser Rangfolgen weitertrainiert. Dieses Verfahren heißt bestärkendes Lernen aus menschlichem Feedback. In den Auswertungen der Forschungsarbeit bevorzugten Menschen die Antworten eines so trainierten Modells mit 1,3 Milliarden Parametern gegenüber denen des ursprünglichen Modells mit hundertmal so vielen Parametern.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein großes Sprachmodell?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Ein Agent wird an Ihre Systeme angebunden, an Fällen Ihrer eigenen Fachleute geprüft und von Ihren Mitarbeitenden beaufsichtigt — ohne Bindung an ein einziges Modell.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Repository-Prüfengine mit 268 Checks entwickelt — 268 Commit-Prüfungen und 146 Tests der Prüfungen selbst

Repository-Prüfengine mit 268 Checks entwickelt Kontakt aufnehmen →

Warum es wichtig ist

Ein Sprachmodell übernimmt Arbeit, die sich nur schwer mit festen Regeln abdecken lässt: eine frei formulierte E-Mail in den richtigen Ordner einordnen, die Felder aus einem eingescannten Lieferschein extrahieren, eine Antwort in der Sprache des Kunden entwerfen oder einen langen Gesprächsverlauf zusammenfassen. Bei Arbeit, für die früher ein Mensch jeden einzelnen Eintrag lesen musste, kann Software nun einen Entwurf erstellen, den ein Mensch prüft.

Die Eigenschaft, die das Modell nützlich macht, erschwert es zugleich, Anwendungen darauf aufzubauen. Seine Ausgabe entsteht gewöhnlich durch Stichprobenziehung, sodass dieselbe Eingabe bei verschiedenen Durchläufen unterschiedliche Antworten liefern kann. Sprachlich flüssig ist es unabhängig davon, ob seine Antwort stimmt. NIST bezeichnet selbstsicher vorgetragene falsche Ausgaben als Konfabulation, besser bekannt als Halluzination, und beschreibt sie als natürliche Folge der Bauweise dieser Modelle: Sie erzeugen Text, der zur Statistik der Texte passt, mit denen sie trainiert wurden. Das ist oft richtig und manchmal nicht. NIST weist auch darauf hin, dass ein Modell die

Begründungen oder Quellenangaben erfinden kann, die eine Antwort zu rechtfertigen scheinen. Eine falsche Antwort sieht genauso aus wie eine richtige. Wie oft ein Modell bei Ihrer Arbeit richtigliegt, lässt sich deshalb nur herausfinden, indem Sie es an Ihrer Arbeit messen.

Es gibt auch nicht das eine Modell, auf das man sich festlegen könnte. Regelmäßig erscheinen neue Modelle, zwei davon können sich bei derselben Eingabe sehr unterschiedlich verhalten, und der Anbieter des Modells, das eine Anwendung aufruft, entscheidet, wann es sich ändert. Ein Team, das auf einem Sprachmodell aufbaut, baut auf einem Bestandteil auf, den es nicht kontrolliert. Dauerhaft wertvoll ist alles rund um das Modell: die Fälle, an denen es gemessen wird, die Werkzeuge, die es aufrufen darf, und die Regeln dafür, was es selbstständig tun darf.

Das NIST-Profil nennt zwölf Risikobereiche für generative KI, darunter Konfabulation, Datenschutz, Informationssicherheit und geistiges Eigentum. Es wurde für Organisationen geschrieben, die Risiken steuern, und nicht für Entwickler. Dennoch ist es eine brauchbare Prüfliste für ein Team, das entscheidet, worauf eine modellgestützte Funktion zugreifen darf.

Wie es in der Praxis aussieht

Eine Funktion, die auf einem Sprachmodell aufbaut, beginnt mit schriftlich festgehaltenen Fällen, nicht mit einem Prompt. Jeder Fall besteht aus einer echten Eingabe und der Antwort, die eine mit der Arbeit vertraute Person akzeptieren würde. Die Fälle liegen in einer Datei unter Versionskontrolle neben dem 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
```

Welches Modell welche Art von Aufgabe übernimmt, ist eine Einstellung. So kann ein kleines, schnelles Modell die Extraktion übernehmen, während ein größeres die Antworten entwirft, für die Urteilsvermögen nötig ist:

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

Jede Änderung wird vor der Veröffentlichung gegen sämtliche Fälle geprüft, unabhängig davon, ob sie einen neuen Prompt, ein neues Werkzeug oder ein neues Modell betrifft. Das Ergebnis ist eine Bewertung für jeden Fall, die im Zeitverlauf aufbewahrt wird, damit die Korrektur eines Falls nicht unbemerkt einen anderen verschlechtert. Erscheint ein neues Modell, genügt zum Ausprobieren eine Änderung an einer Zeile in `routing.yaml` und ein Durchlauf über die Fälle. Die Entscheidung zum Wechsel beruht auf den Bewertungen und nicht auf der Ankündigung.

So beschreibt diese Website ihre individuellen KI-Agenten. Die Modellwahl ist Konfiguration statt Code, mit einem kleinen, schnellen Modell zum Einordnen und Extrahieren und einem großen für Aufgaben, die Urteilsvermögen verlangen. Ein Auftrag beginnt damit, dreißig bis fünfzig echte Beispiele mit den richtigen Antworten aufzuschreiben. Jede Änderung, ob an einem Prompt, einem Werkzeug oder dem Modell, wird vor der Veröffentlichung gegen alle Beispiele geprüft. Die Seite erklärt außerdem, dass das Modell austauschbar bleibt und der Code, die Fälle und die Daten dem Kunden gehören. Damit gehört der Agent dem Kunden weiterhin, wenn sich das darunterliegende Modell ändert.

Wo es schiefgeht

Selbstsichere Antworten, die falsch sind. Wird ein Modell nach einer Vertragsklausel gefragt, die ihm nicht vorliegt, beschreibt es womöglich trotzdem eine, im selben sicheren Ton wie bei einer richtigen Antwort, und nennt vielleicht eine Quelle, die nicht existiert. Geben Sie dem Modell die Dokumente, aus denen es antworten soll, verlangen Sie Quellenangaben und prüfen Sie im Code, dass jede Angabe auf ein echtes Dokument und eine konkrete Textstelle

verweist. Messen Sie die Fehlerquote anhand schriftlich festgehaltener Fälle und leiten Sie folgenreiche Antworten an einen Menschen weiter, bevor irgendetwas auf ihrer Grundlage handelt.

Anweisungen, die in den Daten versteckt sind. Eine Anwendung, die ein Modell E-Mails, Webseiten oder hochgeladene Dateien lesen lässt, lässt die Verfasser dieser Texte mit dem Modell sprechen. Eine Forschungsarbeit von 2023 zeigte dies an echten Anwendungen: Anweisungen in Inhalten, die das Modell voraussichtlich lesen würde, brachten es dazu, Daten preiszugeben und das Verhalten der Anwendung zu verändern. Das NIST-Profil führt indirekte Prompt Injection unter den Sicherheitsrisiken generativer KI auf. Behandeln Sie alles, was das Modell liest, als nicht vertrauenswürdige Eingabe. Setzen Sie Berechtigungen in den Werkzeugen durch, nicht im Prompt, geben Sie den Werkzeugen nur die Rechte der Person, für die das Modell handelt, und verlangen Sie die Zustimmung eines Menschen zu jeder Aktion, die sich nicht rückgängig machen lässt.

Ein Modell ändert sich unter Ihrer Anwendung. Eine neue Version oder ein anderer Anbieter kann dieselbe Eingabe anders beantworten. Eine Antwort, die letzten Monat noch den Test bestand, fällt heute durch, ohne dass sich Ihr eigener Code geändert hätte. Legen Sie eine genaue Modellversion fest, wo der Anbieter eine anbietet, und führen Sie bei jeder Änderung, auch bei einem Modellwechsel, die gesamte Falldatei aus, bevor die Änderung Nutzer erreicht.

Ausgaben, die Code nicht parsen kann. Ein Modell, das JSON liefern soll, gibt JSON mit einem freundlichen Satz dahinter zurück, lässt ein Feld weg oder schreibt eine Zahl als Zeichenkette. Beschreiben Sie die erwartete Struktur mit JSON Schema, prüfen Sie jede Antwort dagegen und behandeln Sie eine Antwort, die die Prüfung nicht besteht, als Fehler, der einen erneuten Versuch oder die Übergabe an einen Menschen erfordert, niemals als Daten.

Personenbezogene Daten in Prompts und Protokollen. Alles, was an ein gehostetes Modell gesendet wird, verlässt Ihre eigenen Systeme. Ein Protokoll mit Prompts und Antworten bewahrt personenbezogene Daten noch lange nach Abschluss der Anfrage auf. NIST ergänzt, dass Modelle sensible Inhalte aus ihren Trainingsdaten wiedergeben können, und bezeichnet dieses Problem als Datenmemorierung. Senden Sie nur die Felder, die eine Aufgabe benötigt. Wenden Sie auf Prompt-Protokolle dieselben Aufbewahrungs- und

Zugriffsregeln an wie auf die darin enthaltenen Daten und lesen Sie die Bedingungen des Anbieters zur Aufbewahrung und zum Training, bevor ihn personenbezogene Daten erreichen.

Weiterführende Lektüre

- NIST AI 600-1: Profil für generative künstliche Intelligenz — die im Juli 2024 veröffentlichte Ergänzung des NIST zu seinem AI Risk Management Framework: zwölf Risiken generativer KI, darunter Konfabulation, und empfohlene Maßnahmen für jedes davon.
- Attention Is All You Need — die Forschungsarbeit von 2017, die den Transformer einführte, die Architektur hinter den heutigen Sprachmodellen.
- Language Models are Few-Shot Learners — die Forschungsarbeit von 2020 über ein Modell mit 175 Milliarden Parametern, das anhand von Beispielen in seiner Eingabe neue Aufgaben übernimmt.
- Training language models to follow instructions with human feedback — Feinabstimmung anhand geschriebener Antworten und bewerteter Ausgaben: der Schritt vom Textvorhersagemodell zum Assistenten.
- Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection — in Daten versteckte Anweisungen, die ein Modell liest, vorgeführt an echten Anwendungen.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein großes Sprachmodell?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/large-language-model/