LLMs und die Generierung von Wissen: Eine technische Analyse

Inhaltsverzeichnis

1 Begriffsdefinitionen: Die Fundamente **2**
1.1 Was ist ein LLM (Large Language Model)? . . . 2
1.2 Unterscheidung: Information vs. Wissen . . . 2
1.3 Definition von „Wissenïn diesem Kontext . . . 3

2 Kann ein LLM Wissen generieren? **3**
2.1 Die generative Illusion . . . 3
2.2 Der Unterschied zwischen Wissen und Halluzination . . . 3
2.3 Randfall: Emergent Abilities . . . 4

3 Das Anwendungsszenario: Nur Modell vs. Modell mit Websuche **4**
3.1 Nur auf das Modell (Statisches Wissen) . . . 4
3.2 Modell mit Websuche (Dynamisches Wissen) . . . 4
3.3 Vergleich der Szenarien . . . 5

4 RAG vs. Websuche: Technische Unterscheidung **5**
4.1 Was ist RAG (Retrieval-Augmented Generation)? . . . 5
4.2 Was ist Websuche? . . . 5
4.3 Der Unterschied im Umfeld von LLM . . . 6

5 Alternativen und Randfälle **6**
5.1 Alternative 1: Fine-Tuning . . . 6
5.2 Alternative 2: Tool Use (Python Interpreter) . . . 6
5.3 Randfall: Datenlecks und Sicherheit . . . 6
5.4 Randfall: Langzeitgedächtnis . . . 7

6 Zusammenfassung und Fazit **7**

Einleitung

Diese Frage berührt das Herzstück der aktuellen KI-Debatte: den Unterschied zwischen statistischer Mustererkennung und echtem Verständnis. Um diese Komplexität angemessen zu beleuchten, ist eine präzise terminologische und technische Fundierung unerlässlich.

Im Folgenden wird die Frage *„Kann mit LLM Wissen generiert werden?"* systematisch aufgeschlüsselt, beginnend mit der Definition der Schlüsselbegriffe, über die Funktionsweise der Modelle bis hin zu den technischen Architekturen (RAG vs. Websuche).

1 Begriffsdefinitionen: Die Fundamente

Bevor wir die generative Kapazität von Modellen bewerten können, müssen wir klären, was wir unter den beteiligten Begriffen verstehen. Ohne diese Definitionen wäre die Diskussion auf einem semantischen Sandboden aufgebaut.

1.1 Was ist ein LLM (Large Language Model)?

Ein Large Language Model ist ein maschineller Algorithmus, der auf der Architektur des **Transformers** basiert. Technisch gesehen ist es ein neuronales Netz, das durch einen Prozess namens **Vorhersage des nächsten Tokens** (Next-Token Prediction) trainiert wird.

- **Funktionsweise:** Das Modell nimmt eine Eingabe (Prompt), zerlegt sie in tokens (Wortstücke) und berechnet basierend auf statistischen Korrelationen in seinem Trainingsdatensatz die Wahrscheinlichkeitsverteilung für das nächste Token.
- **Schlüsselmechanismus:** Der **Self-Attention-Mechanismus** erlaubt dem Modell, Beziehungen zwischen weit entfernten Wörtern im Satz zu verstehen (z. B. Subjekt-Objekt-Beziehung über viele Sätze hinweg).
- **Training:** Es wird überwacht (Supervised Fine-Tuning) und auf menschliche Werte abgestimmt (Reinforcement Learning from Human Feedback - RLHF), um die Ausgaben sicherer und nützlicher zu machen.
- **Wichtig zu wissen:** Ein LLM besitzt kein Bewusstsein, keine Intentionalität und keinen Zugriff auf eine externe Realität. Es ist ein hochkomplexer Text-Vorhersagemotor.

1.2 Unterscheidung: Information vs. Wissen

In der Informatik und Epistemologie werden diese Begriffe oft synonym verwendet, aber für die Bewertung von LLMs ist die Unterscheidung essenziell.

- **Information (Data/Information):** Dies sind rohe Fakten, Datenpunkte oder Symbole, die noch keinen semantischen Kontext haben.

  Beispiel: „Der Eiffelturm ist 330 Meter hoch."

- **Definition:** Eine nachweisbare Aussage über einen Zustand der Welt, die wahr oder falsch sein kann.
- **Wissen (Knowledge):** Dies ist Information, die kontextualisiert, verifiziert, strukturiert und in einen kausalen oder logischen Zusammenhang gesetzt wurde.

Beispiel: „Da der Eiffelturm 330 Meter hoch ist, muss er für den Bau von Brücken in der Nähe eine besondere Berechnung der Windlast erfordern."

- **Definition:** Die Fähigkeit, Informationen zu nutzen, um Schlussfolgerungen zu ziehen, Probleme zu lösen oder Entscheidungen zu treffen. Wissen impliziert **Verstehen und Anwendung**.

1.3 Definition von „Wissenïn diesem Kontext

Für die folgende Analyse definieren wir **Wissen** operativ wie folgt:

Definition

Wissen ist der Zustand, in dem eine Entität (hier: der Mensch oder das System) über eine Sorte von Fakten verfügt, deren Konsistenz überprüft werden kann, deren Ursprung bekannt ist und die es ihr ermöglicht, Prädiktionen über die Welt zu treffen oder Handlungen zu steuern.

Ein LLM *„generiert"* also keine Fakten im Sinne der Entdeckung einer physikalischen Wahrheit (wie ein Experiment), sondern es **synthetisiert** aus seiner internen Repräsentation (Gewichten) neue Textstrukturen, die für den Nutzer als Wissen wahrgenommen werden.

2 Kann ein LLM Wissen generieren?

Die Antwort ist nicht einfach mit Ja oder Nein zu beantworten, sondern erfordert eine Nuance zwischen *synthetisiertem Output* und *ontologischer Erkenntnis*.

2.1 Die generative Illusion

Ein LLM kann keine neuen physikalischen Gesetze entdecken, da es keine Daten der Welt in Echtzeit verarbeitet. Es kann jedoch **neues Wissen synthetisieren**:

- **Kontextualisierung:** Wenn ein Nutzer fragt: „Wie kann ich mein Budget optimieren, wenn ich X und Y tue?", generiert das LLM eine neue Zusammenführung von Informationen, die der Nutzer zuvor nicht selbst verknüpft hat.
- **Schlussfolgerung (Reasoning):** Durch Techniken wie *Chain-of-Thought* (Kette des Denkens) kann das Modell logische Schritte simulieren, die zu einer neuen Erkenntnis führen (z. B. mathematische Problemlösung).

2.2 Der Unterschied zwischen Wissen und Halluzination

Ein LLM kann eine plausible Antwort generieren, die faktisch falsch ist (Halluzination).

- **Szenario:** Das Modell behauptet, ein historisches Ereignis sei zu einem anderen Zeitpunkt passiert.
- **Fazit:** Das ist keine Generierung von Wissen, sondern statistische Fortsetzung eines Musters, das im Training existierte.

- **Erkenntnis:** Ein LLM ist also ein **Wiederkombinations-System**, kein **Erkenntnis-System**. Es kann Wissen *reproduzieren* und *transformieren*, aber nicht im Sinne von *Entdecken*.

2.3 Randfall: Emergent Abilities

Bei sehr großen Modellen treten sogenannte *Emergent Abilities* auf. Das sind Fähigkeiten, die nicht in den einzelnen Trainingsschritten sichtbar waren, aber bei der Größe des Modells auftauchen (z. B. logisches Multiplikieren).

- **Bedeutung:** Dies deutet darauf hin, dass das Modell eine interne Repräsentation der Welt entwickelt hat, die es nutzt, um *„Wissen"* zu simulieren. Es bleibt jedoch eine statistische Approximation und kein echtes Verständnis.

3 Das Anwendungsszenario: Nur Modell vs. Modell mit Websuche

Hier liegt der entscheidende Unterschied in der **Aktualität** und der **Datenbasis**.

3.1 Nur auf das Modell (Statisches Wissen)

- **Einschränkung:** Das Modell kennt nur das, was es während des Trainings (z. B. bis Ende 2023 bei vielen Modellen) gelernt hat.
- **Nachteil:** Es weiß nichts über aktuelle Ereignisse, neue wissenschaftliche Studien oder interne Firmendaten.
- **Risiko:** Wenn das Modell auf Fakten reagiert, die nach dem Training entstanden sind, muss es diese *„erfunden"* (halluzinieren), um den Kontext zu erfüllen.
- **Anwendung:** Gut für allgemeine Fragen, kreatives Schreiben, Logik, die nicht auf aktuellen Fakten basiert.

3.2 Modell mit Websuche (Dynamisches Wissen)

Durch die Integration von Suchmaschinen (wie Google oder Bing) in den Prompt-Zyklus erhält das Modell Zugriff auf den *„Live"*-Internet.

- **Vorteil:** Das Modell kann auf aktuelle Nachrichten, Aktienkurse oder neue wissenschaftliche Artikel zugreifen.
- **Prozess:** Der Nutzer fragt nach etwas Aktuellem → Das System sucht im Internet → Extrahiert relevante Snippets → Das LLM generiert die Antwort basierend auf diesen Snippets.
- **Unterschied:** Hier wird das Modell nicht zum Wissensspeicher, sondern zum **Analyse-Interface** für externe Daten. Das Modell wird passiv, die Quelle wird aktiv.

Tabelle 1: Vergleich: Nur Modell vs. Modell mit Websuche

Merkmal	**Nur Modell (Statisch)** **Modell + Websuche (Dynamisch)**
Wissensbasis	Eingeschränkt auf Trainingszeitpunkt & Unbegrenzt (aktuell)
Verifizierung	Schwierig (keine Quelle) & Einfacher (Zitat von Quelle möglich)
Kosten	Gering (nur Inference) & Höher (API-Kosten für Suche)
Halluzinationsrisiko	Hoch bei aktuellen Fakten & Geringer (basierend auf Quellen)

3.3 Vergleich der Szenarien

4 RAG vs. Websuche: Technische Unterscheidung

Im Umfeld von LLMs werden beide Konzepte oft vermischt. Sie sind jedoch technisch und architektonisch unterschiedlich.

4.1 Was ist RAG (Retrieval-Augmented Generation)?

RAG ist eine **Architektur**.

1. **Retrieval:** Ein externes System (z. B. ein Vektor-Datenbank) durchsucht einen spezifischen Datensatz.
2. **Augmentation:** Die gefundenen Dokumente werden dem Prompt des LLMs hinzugefügt (Context Window).
3. **Generation:** Das LLM formuliert die Antwort.

- **Quelle:** Meist eine private, strukturierte Datenbank (z. B. PDFs eines Unternehmens, medizinische Akten, interne Handbücher).
- **Ziel:** Genauigkeit bei spezifischen Firmendaten, Datenschutz (Daten bleiben im System), Reduzierung von Halluzinationen durch Quellenbezug.
- **Beispiel:** Ein Anwalt fragt nach einem Paragraphen. Das System sucht im internen Dokumentenpool, gibt den Text an das LLM. Das LLM antwortet mit Zitat.

4.2 Was ist Websuche?

Websuche ist ein **Eingabe-Kanal**.

1. **Query:** Ein Suchbegriff wird an einen Such-Engine-Schnittstellen (API) gesendet.
2. **Ergebnisse:** Die Suchmaschine (z. B. Bing Search API) liefert eine Liste von URLs und Snippets.

3. **Integration:** Diese Snippets werden an das LLM gesendet (ähnlich wie RAG, aber ohne strukturierte Vektordatenbank, sondern unstrukturiertes Web).

4.3 Der Unterschied im Umfeld von LLM

- **RAG vs. Websuche:**
  - **RAG** ist oft **kontrolliert und strukturiert**. Es ist für Unternehmen, die ihre eigenen Daten nutzen wollen (Privacy First).
  - **Websuche** ist **offen und unstrukturiert**. Es ist für allgemeine Fragen, die keine internen Daten benötigen.
- **Kombination:** Oft wird beides kombiniert. Das LLM nutzt Websuche, um den *Kontext* zu finden (z. B. „Was ist die aktuelle Lage bei X?"), und nutzt RAG, um *interne Fakten* zu prüfen (z. B. „Wie hat unser Unternehmen X gelöst?").
- **Wichtig:** RAG kann *auch* Websuche nutzen, wenn die Suchmaschine als externe Datenbank eingebunden wird. Aber technisch ist RAG der **Prozess** der Retrieval-Augmentation, Websuche ist die **Quelle** der Daten.

5 Alternativen und Randfälle

Um die Argumentation abzurunden, müssen wir Alternativen und Grenzen betrachten, die über den reinen LLM hinausgehen.

5.1 Alternative 1: Fine-Tuning

Anstatt Websuche zu nutzen, kann man das Modell durch **Fine-Tuning** auf neuen Daten anpassen.

- **Vorteil:** Das Modell *„lernt"* die neuen Fakten in seine Gewichte einbrennen.
- **Nachteil:** Die Kosten sind hoch (Rechenleistung), und das Wissen ist wieder statisch (altert).
- **Vergleich:** Websuche ist flexibler (keine Retraining-Kosten), RAG ist sicherer (Daten bleiben separat). Fine-Tuning ist teurer und weniger wartbar für aktuelle Daten.

5.2 Alternative 2: Tool Use (Python Interpreter)

Ein LLM kann nicht nur Text, sondern auch Code ausführen.

- **Szenario:** Das LLM berechnet eine mathematische Formel oder ruft eine API auf.
- **Wissen-Aspekt:** Hier wird Wissen durch *Berechnung* generiert, nicht durch Textvorhersage. Dies ist der einzige Weg, bei dem ein LLM *„echtes"* mathematisches Wissen beweisen kann, ohne sich darauf zu berufen, was es gelesen hat.

5.3 Randfall: Datenlecks und Sicherheit

Wenn ein LLM Websuche nutzt, kann es potenziell auf sensible Daten stoßen (z. B. bei schlecht gesicherten APIs).

- **RAG-Lösung:** Da RAG auf einer eigenen Datenbank basiert, kann man Zugriffskontrollen (ACL) um das RAG-System herum bauen. Websuche bietet hier keine Sicherheit.

5.4 Randfall: Langzeitgedächtnis

Ein LLM hat kein Gedächtnis. Es *„vergisst"* nichts, aber es kann nichts *neues* speichern.

- **Lösung:** Durch RAG kann man ein *„Langzeitgedächtnis"* simulieren. Man speichert neue Fakten in einer Vektordatenbank. Das LLM *„erinnert"* sich daran, indem es diese Daten beim Abruf findet.

6 Zusammenfassung und Fazit

Die Frage, ob LLMs Wissen generieren können, hängt von der Definition von *„Wissen"* ab.

1. **Technisch:** LLMs generieren keine neuen Fakten, sondern kombinieren statistische Muster. Sie sind **Wiederkombinationssysteme**.
2. **Pragmatisch:** Sie können **neues Wissen für den Nutzer** generieren, indem sie Informationen aus verschiedenen Quellen (RAG/Websuche) zu einer kohärenten, neuen Antwort verbinden.
3. **Dynamik:** Ein reines Modell ist statisch (Wissen ist eingefroren). Die Kombination mit Websuche macht das System dynamisch (Wissen ist flüssig).
4. **Architektur:** RAG ist die Methode zur Anbindung von spezifischen Daten, Websuche die Methode zur Anbindung des offenen Internets. Beide können kombiniert werden.

Fazit: Ein LLM ohne externe Anbindung (RAG/Websuche) ist ein **Wissens-Synthesizer** für das, was bereits existiert. Ein LLM mit Websuche und RAG ist ein **Wissens-Interface**, das Echtzeit-Zugriff auf die Welt hat. Es ist wichtig zu verstehen, dass das Modell selbst kein *„Wissens-Inhaber"* ist, sondern ein Werkzeug, das den Nutzer hilft, Informationen zu finden und zu strukturieren. Die Qualität des generierten Outputs hängt direkt von der Qualität der zurückgerufenen Informationen (Garbage In, Garbage Out) und der Fähigkeit des Modells, diese logisch zu verbinden.