

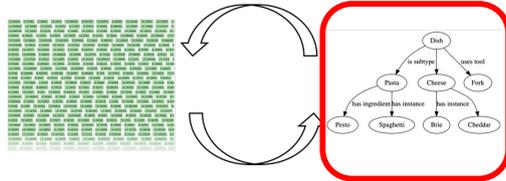
Überblick

Wissensgraphen 

Large Language Models (LLM) 

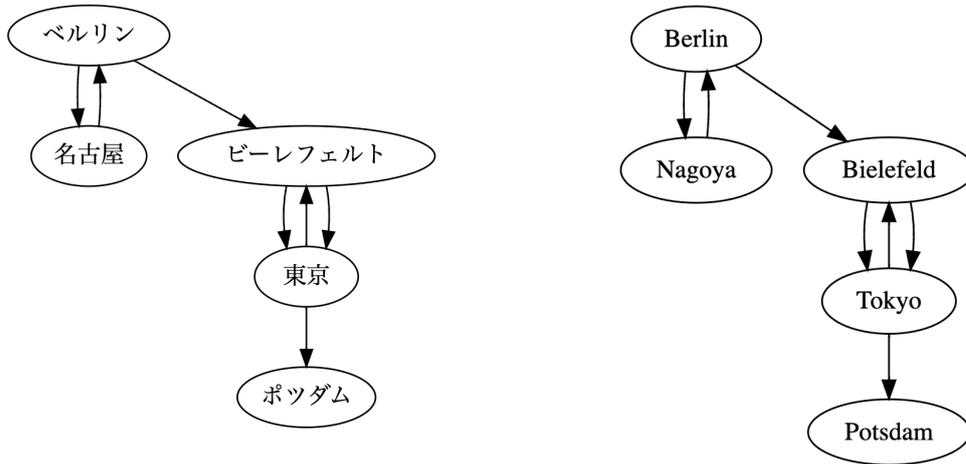
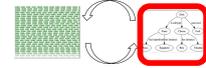
Retrieval, LLMs und Wissensgraphen 

Best Practices: Wissensgraphen und LLMs 



Wissensgraphen

Beispiel eines Wissensgraphen (anstatt einer Selbstvorstellung)

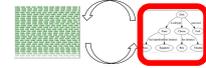


Public

4

Hier finden Sie zwei Varianten von Wissensgraphen, die meinen persönlichen Werdegang darstellen. Diese Folie veranschaulicht eine wichtige Eigenschaft von Wissensgraphen. Sie sind unabhängig von der menschlichen Sprache und stellen die Struktur in einer Domäne dar. Beschreibungen in verschiedenen Sprachen können in Wissensgraphen ausgedrückt werden. Hier sehen Sie ein Beispiel: einen Wissensgraphen mit einer Beschreibung auf Deutsch und denselben Wissensgraphen mit einer Beschreibung auf Japanisch.

Beispiel für einen Wissensgraph: Google Websuche



SAP
Unternehmen



Die SAP SE mit Sitz im baden-württembergischen Walldorf ist ein börsennotierter Softwarekonzern. Nach Umsatz ist SAP das größte europäische sowie weltweit eines der fünf größten Softwareunternehmen. Darüber hinaus handelt es sich bei SAP um das mit Abstand wertvollste börsennotierte deutsche Unternehmen. [Wikipedia](#)

Gründer: [Dietmar Hopp](#), [Hasso Plattner](#), [Claus Wellenreuther](#), [Hans-Werner Hector](#), [Klaus Tschira](#)

Hauptsitz: [Walldorf](#)

Umsatz: 30,87 Milliarden EUR (2022)

SAP
会社



SAP SEは、ドイツ中西部バーデン＝ヴュルテンベルク州にあるヴァルドルフに本社を置くヨーロッパ最大のソフトウェア会社である。フランクフルト証券取引所、ニューヨーク証券取引所上場企業。
[ウィキペディア](#)

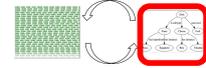
本部所在地: [ヴァルドルフ](#)

Public

5

Google hat den Begriff Knowledge Graph geprägt. Ein wichtiger Anwendungsfall für den Wissensgraphen ist nach wie vor die Websuche. Hier sehen Sie die Suchergebnisse von SAP Japan. Sie können sehen, dass derselbe Wissensgraph sowohl auf Japanisch als auch auf Englisch angezeigt werden kann. Dies unterstreicht die Bedeutung der Mehrsprachigkeit für Wissensgraphen.

Kernbestandteile von Wissensgraphen „Entitäten und Attribute“



Entitäten

SAP
Unternehmen

Die SAP SE mit Sitz im baden-württembergischen Walldorf ist ein börsennotierter Softwarekonzern. Nach Umsatz ist SAP das größte europäische sowie weltweit eines der fünf größten Softwareunternehmen. Darüber hinaus handelt es sich bei SAP um das mit Abstand wertvollste börsennotierte deutsche Unternehmen. [Wikipedia](#)

Gründer: Dietmar Hopp, Hasso Plattner, Claus Wellenreuther, Hans-Werner Hector, Klaus Tschira

Hauptsitz: Walldorf

Umsatz: 30.87 Milliarden EUR (2022)

Attribute

Entitäten

SAP
会社

SAP SEは、ドイツ中西部バーデン＝ヴュルテンベルク州にあるヴァルドルフに本社を置くヨーロッパ最大のソフトウェア会社である。フランクフルト証券取引所、ニューヨーク証券取引所上場企業。
[ウィキペディア](#)

本部所在地: ヴァルドルフ

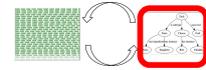
Attribute

Public

6

Die Hauptbestandteile eines Wissensgraphen sind Knoten und Kanten. Knoten werden verwendet, um Entitäten wie „SAP“ oder „Walldorf“ darzustellen. Durch Kanten werden Attribute definiert. Zum Beispiel wird das Attribut "Hauptsitz" verwendet, um eine Entität namens SAP mit einer Entität namens Walldorf zu verbinden.

Use Cases für Wissensgraphen



Websuche

🔍 SAPジャパン 東京



日本 東京の企業のオフィス

所在施設: 三井物産ビル

所在地: 日本、〒100-0004 東京都千代田区大手町1丁目2-1 三井物産ビル

営業時間: 営業時間外 · 営業開始: 9:00 ▶

Jetzt wollen wir einige gängige Anwendungsfälle für Wissensgraphen vorstellen.

Die Websuche ist nach wie vor ein wichtiger Anwendungsfall. Hier wird nach SAP Japan Tokyo gesucht. Die Suchergebnisse enthalten verschiedene Merkmale von SAP Japan, z. B. Standort und Öffnungszeiten.

Use Cases für Wissensgraphen



Websuche

🔍 SAPジャパン 東京

日本 東京の企業のオフィス

所在施設: 三井物産ビル

所在地: 日本、〒100-0004 東京都千代田区大手町1丁目2-1 三井物産ビル

営業時間: 営業時間外 · 営業開始: 9:00 ▶

Virtueller Assistent



「SAPジャパン 東京」

Public

8

Ein weiterer wichtiger Anwendungsfall für den Wissensgraphen ist der virtuelle Assistent. Virtuelle Assistenten können denselben Wissensgraphen verwenden wie die Websuche. Wie Sie an diesem Beispiel sehen können, besteht der Unterschied zwischen der Websuche und den virtuellen Assistenten darin, dass sie eine andere Benutzeroberfläche für den Wissensgraphen bieten.

Use Cases für Wissensgraphen



Websuche

🔍 SAPジャパン 東京

日本 東京の企業のオフィス

所在施設: 三井物産ビル

所在地: 日本、〒100-0004 東京都千代田区大手町1丁目2-1 三井物産ビル

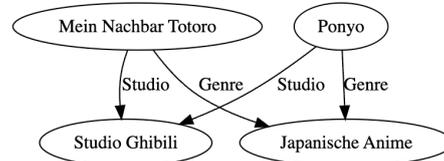
営業時間: 営業時間外 · 営業開始: 9:00 ▶

Virtueller Assistent



「SAPジャパン 東京」

Empfehlungssysteme



„Mein Nachbar Totoro“ → „Ponyo“

Public

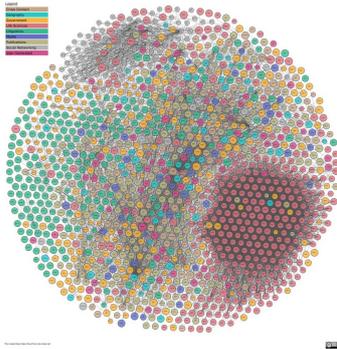
9

Ein weiterer häufiger Anwendungsfall für Wissensgraphen sind Empfehlungssysteme. Hier sehen Sie einen Wissensgraphen, der den Film „Mein Nachbar Totoro“ und den Film „Ponyo“ enthält. Auf der Grundlage eines solchen Wissensgraphen kann bei einer Suche nach „Mein Nachbar Totoro“ eine Empfehlung für „Ponyo“ ausgesprochen werden.

Öffentlich verfügbare und verlinkte Wissensgraphen



Linked Open Data – FAIR * Data



* <https://www.go-fair.org/fair-principles/>

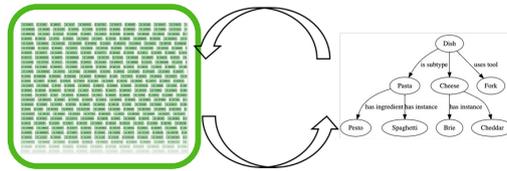
Public

Im Bibliotheksbereich vgl. z.B. [Semantic Web in Libraries](#)

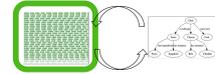
10

Es gibt eine zunehmend wachsende Menge von öffentlich verfügbaren Wissensgraphen, die auch unter dem Begriff „Linked Open Data“ bekannt sind. Hier sehen sie die Linked Open Data Cloud, ein automatisch generiertes Diagramm der LOD. Die Farben drücken verschiedenen Domänen aus. Bekannte generelle und miteinander verknüpfte Wissensgraphen sind Wikidata oder DBpedia, und verbreitete Vokabulare wie Schema.org. Auch die DNB stellt Katalogdaten im Format von Wissensgraphen bereit. Und die Bibliothekcommunity ist schon seit längerem unter anderen in der Tagungsreihe „Semantic Web in Libraries“ in diesem Bereich aktiv.

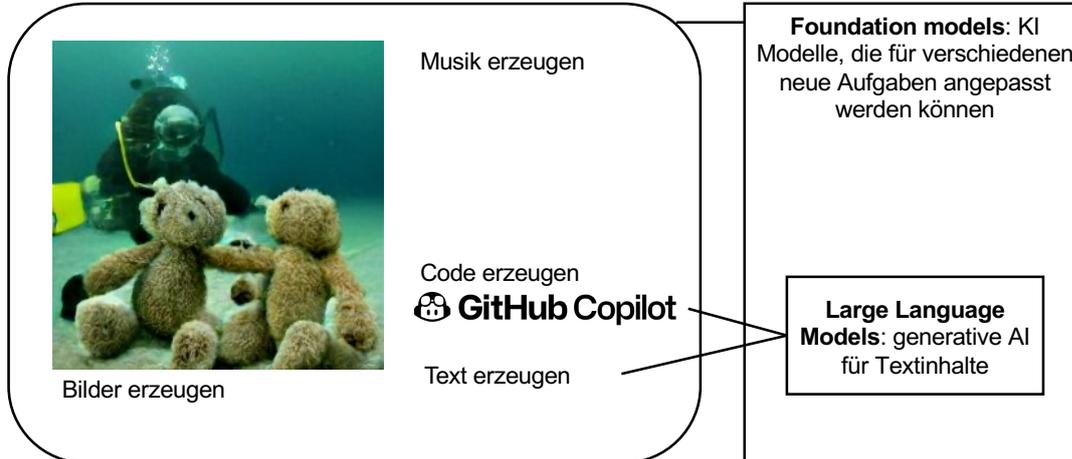
Große Sprachmodelle (LLM)



LLM im Kontext



Generative KI



Public

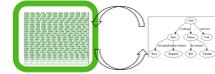
12

Große Sprachmodelle sind eine Art der „generativen KI“. Generative KI ist ein Ansatz zur Generierung von Inhalten. „Inhalte“ können Bilder sein, wie z. B. "Teddybären, die mit der Technologie von 1990 an einer neuen KI-Forschung unter Wasser arbeiten", oder Code, Musik usw.

Bei der Generierung werden Basismodelle verwendet. Das sind KI-Modelle, die an verschiedene und unvorhergesehene Aufgaben angepasst werden können. Große Sprachmodelle sind generative KI für Textinhalte.

In dieser Präsentation wird meist der Begriff „große Sprachmodelle“ verwendet. Aber was ich sage, kann auch auf Basismodelle im Allgemeinen angewandt werden.

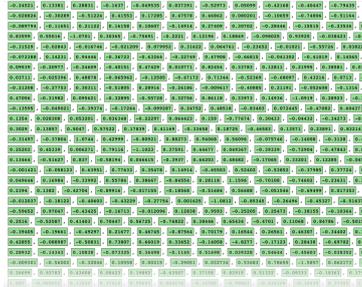
Wie funktionieren große Sprachmodelle? (2/2)



Positionen im 2D-Vektorraum



Sprachmodell Vektorraum: das Wort „happy“



What is a **tidy** thing to eat pasta with?

A **tidy** thing to eat pasta with is a **fork**.

- **Transformer:** Neu-Berechnung der Vektoren von Token („Wörtern“)
- **Aufmerksamkeit:** Berücksichtigung des vorherigen Kontextes

What is a **nice** thing to eat pasta with?

A **nice** thing to eat pasta with is **pesto**.

Public

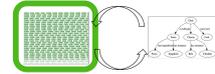
Quelle: [How AI chatbots like ChatGPT or Bard work – visual explainer](https://bbvcroft.net/llm). Interaktive Demo: <https://bbvcroft.net/llm>

14

In großen Sprachmodellen werden die Beziehungen zwischen Wörtern in abhängig von den vorherigen Wörtern in der Eingabe neu gewichtet und die Vektoren neu berechnet. Die Verknüpfungen zwischen zum Beispiel "tidy" mit "utility" und "tools", beeinflussen das Ergebnis für das Wort „with“ im ersten Satz. Es wird im Sinn von „unter Verwendung von“ oder „als Werkzeug“ interpretiert. Im zweiten Satz beeinflusst das Wort „nice“ die Interpretation von „with“ im Sinne von „gemeinsam mit“ oder „mit der Beilage“

Die Berücksichtigung des Kontextes wird als "Aufmerksamkeit" oder auch „Attention“ bezeichnet. Der Prozess der Neugewichtung von Beziehungen zwischen Wörtern und der Neuberechnung der Vektoren wird als "Transformer" bezeichnet. Die Prozesse werden auf jeden Teil einer Konversation angewendet. Dadurch wird die Bedeutung der Wörter in Bezug auf den Gesamtkontext immer wieder neu erfasst.

Funktionsweise von LLMs, und deren Folgen



Positionen im 2D-Vektorraum



Sprachmodell Vektorraum: das Wort „happy“



What is a **tidy** thing to eat pasta with?

A tidy thing to eat pasta with is a fork.

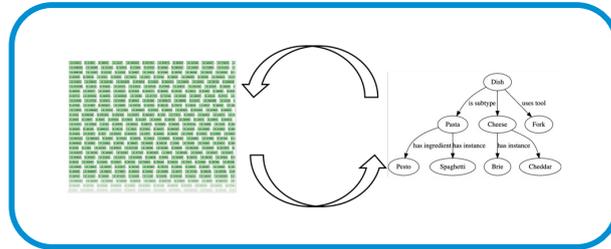
What is a **nice** thing to eat pasta with?

A nice thing to eat pasta with is pesto.

- Transformer: Neu-Gewichtung von Beziehungen zwischen Wörtern
- Aufmerksamkeit: Berücksichtigung des vorherigen Kontextes
- Milliarden von Parametern zur Neu-Gewichtung → rechenintensiv, langsam, teuer; wahrscheinlichkeitsbasiert

Quelle: [How AI chatbots like ChatGPT or Bard work – visual explainer](https://bbvcroft.net/llm). Interaktive Demo: <https://bbvcroft.net/llm>

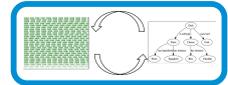
Die beschriebene Funktionsweise von großen Sprachmodellen hat Folgen. Zum einen, Vektorenberechnungen sind sehr aufwändig. Bei der ersten Version von ChatGTP nutzt die Neuberechnung von Gewichten 175 Milliarden Variablen, so genannte Parameter. Das bedeutet: rechenintensive, zeitaufwändige und teure Operationen. Zum anderen ist das Ergebnis wahrscheinlichkeitsbasiert: das Modell berechnet jeweils eine statistische Wahrscheinlichkeit für das nächste Wort in der Ausgabe.



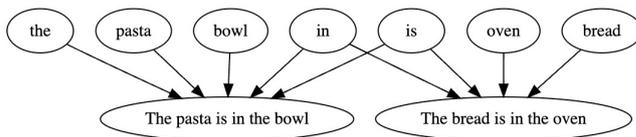
Retrieval, LLM und Wissensgraphen

Welche Rolle spielen nun LLM und Wissensgraphen für Informationsretrieval? Für diese Fragen wollen das Verhältnis von Retrievalansätzen beleuchten.

Vom invertierten Index zu Large Language Models (1/3)

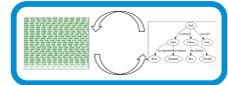


Invertierter Index

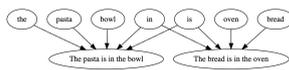


Eine grundlegende Form des Retrievals ist der invertierte Index. Der Vorteil ist, dass Worte eindeutig Dokumenten zugeordnet sind. Die Verarbeitung ist somit schnell und präzise. Ein Nachteil ist, dass die semantischen Beziehungen von Begriffen nicht erfasst sind.

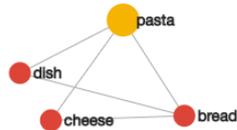
Vom invertierten Index zu Large Language Models (2/3)



Invertierter Index



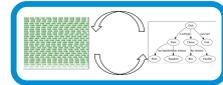
Wort Embeddings



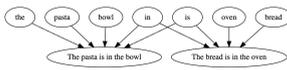
Quelle: [WebVectors: word embeddings online](#)
"a word is characterized by the company it keeps"

Word Embeddings erfassen die Bedeutung von Wörtern als Vektoren. Die Berechnung erfolgt anhand der Einbettung in den Kontext in großen Datenmengen. Man sieht sofort den Mehrwert für den Retrieval. Wörter wie „cat“ und „dog“ tauchen in einem ähnlichen Kontext wie „animal“ oder „rabbit“ auf. Diese Beziehungen lassen sich für eine semantische Suche verwenden. Der Kontext in einem konkreten Satz wird dabei allerdings noch nicht erfasst.

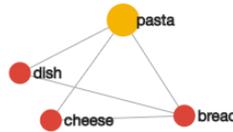
Vom invertierten Index zu Large Language Models (3/3)



Invertierter Index



Wort Embeddings



Quelle: [WebVectors: word embeddings online](https://www.wordembeddings.com/)
 "a word is characterized by the company it keeps"

LLM

What is a tidy thing to eat pasta with?

A tidy thing to eat pasta with is a fork.

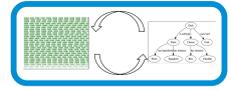
What is a nice thing to eat pasta with?

A nice thing to eat pasta with is pesto.

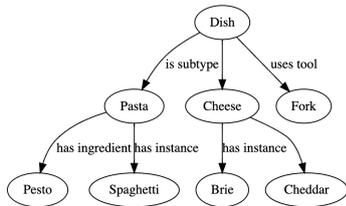
"attention is all you need"

Große Sprachmodelle sind dazu in der Lage. Sie stellen die beschriebenen Mechanismen von Aufmerksamkeit und Neuberechnung von Gewichten bereit. Dadurch kann man weitergehende, kontextuelle Beziehungen erfassen, welche über die Wortembeddings hinaus gehen, wie in dem Beispiel die Beziehung zwischen „tidy“ und „a fork“ oder „nice“ und „pesto“, und die beschriebene unterschiedliche Bedeutung von „with“.

Wissensgraphen und LLMs

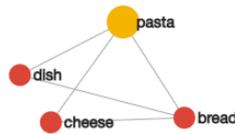


Wissensgraphen



Explizite Modellierung

Wort Embeddings



Quelle: [WebVectors: word embeddings online](#)
 "a word is characterized by the company it keeps"

Implizierte Modellierung

LLM

What is a tidy thing to eat pasta with?

A tidy thing to eat pasta with is a fork.

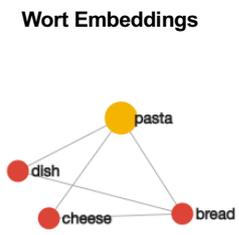
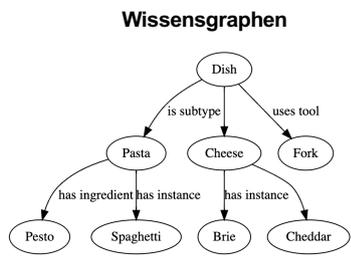
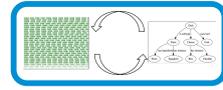
What is a nice thing to eat pasta with?

A nice thing to eat pasta with is pesto.

"attention is all you need"

Wissensgraphen sind eine explizite Form der Wissensmodellierung. Hier ist ein Wissensgraph für die Domäne „Essen“. Man kann dadurch die Zusammenhänge z.B. zwischen „Pesto“ und „Pasta“ erfassen. Man würde allerdings in dem Graphen nicht die Beziehung zwischen „nice“ und „pasta“ erfassen, oder die Bedeutung von „with“ im Kontext einer konkreten Verwendung.

Vorteile von Wissensgraphen, Nachteile von LLMs*



LLM

What is a tidy thing to eat pasta with?
A tidy thing to eat pasta with is a fork.

What is a nice thing to eat pasta with?
A nice thing to eat pasta with is pesto.

Quelle: [WebVectors: word embeddings online](https://webvisu.github.io/word2vecviz/)
 "a word is characterized by the company it keeps"

- ← **Präzision**
- ← **Einfach erweitert werden**
- ← **Erklärbarkeit**
- ← **Konsistente Ergebnisse**
- ← **Auf andere Sprachen anwendbar**
- ← **Domänenspezifisch („long tail“)**

LLMs are awesome, but...

- Halluzinationen
- Expensive to train & run
- Difficult to fix & update
- Hard to audit & explain
- Inconsistent answers
- Low resource languages
- Coverage gap on long tail

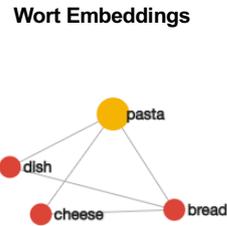
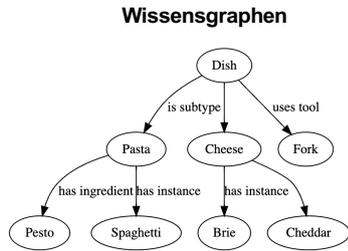
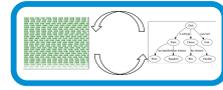
- Halluzination**
- Training & Ausführung teuer**
- Aktualisierung aufwändig**
- Ergebnisse schwer erklärbar**
- Inkonsistente Antworten**
- Sprachabdeckung nur fokussiert**
- Keine Domänenabdeckung („long tail“ fehlt)**

* Formuliert von Denny Vrandečić, Mitentwickler von [Wikidata](https://www.wikidata.org/)

Wissensgraphen haben verschiedene Vorteile. Das Wissen ist präzise, die Graphen können einfach erweitert werden, die Ergebnisse einer Suche im Graphen sind erklärbar. Die Ergebnisse sind zudem konsistent und auf andere Sprachen anwendbar, vgl. meinen Lebenslauf auf Japanisch. Zudem kann man mit Wissensgraphen domänenspezifisches „long tail“ Wissen, anders als bei LLMs, ohne umfangreiche Daten gut erfassen.

Denny Vrandečić, einer der Mitbegründer des Wissensgraphen „Wikidata“, hat in einem eindrucksvollen Video, das hier verlinkt ist, die Herausforderungen von LLMs beschrieben. Diese Folie fasst diese Herausforderungen zusammen. Besonders hervorheben kann man das Phänomen der Halluzination. Ein Beispiel ist die Anfrage vom 3. Dezember an ein Sprachmodell „Erzähle mir etwas zur Tagung "KI in Bibliotheken: Neue Wege mit großen Sprachmodellen". Das Modell antwortet, dass die Tagung bereits stattgefunden hat und ein voller Erfolg war, und sie wertvolle Impulse für die praktische Umsetzung von KI-gestützten Lösungen in Bibliotheken geliefert hat.

Nachteile von Wissensgraphen, Vorteile von LLMs



LLM

What is a tidy thing to eat pasta with?

A tidy thing to eat pasta with is a fork.

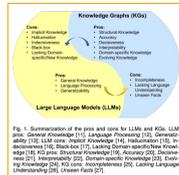
What is a nice thing to eat pasta with?

A nice thing to eat pasta with is pesto.

Quelle: [WebVectors: word embeddings online](#)
 "a word is characterized by the company it keeps"

"attention is all you need"

- Unvollständige Abdeckung
- Kein sprachbezogener Zugang möglich
- Keine Erfassung von nicht neuen, nicht explizit modellierten Informationen



- Breite Wissensabdeckung
- Sprachbezogener Zugang möglich & flexibel
- Generalisierbarkeit

Public

Shirui Pan et al. [Unifying Large Language Models and Knowledge Graphs: A Roadmap](#)

22

Wissensgraphen halluzinieren nicht, sie haben aber auch Nachteile, welche der hier verlinkte Artikel zusammenfasst, und in Bezug zu LLMs setzt. Die Wissensgraphen sind zwar leicht erweiterbar, aber die Abdeckung einer Domäne kann manuell nur mit hohem Aufwand erreicht und durch fortwährende Aktualisierung sichergestellt werden. Wissensgraphen haben keinen flexiblen sprachbezogenen Zugang. Der vorliegende Wissensgraph hat alle Informationen für eine Frage wie „welche Beilage passt gut zu Pasta?“. Aber eine entsprechende Anfrage an den Graphen erfordert Kenntnisse in Querysprachen. Zudem bleiben nicht explizit modellierte Informationen im Wissensgraphen unberücksichtigt. Als Beispiel dient wieder die Beziehung zwischen „nice“ und „pesto“. Sie ist im Kontext einer Äußerung sinnvoll. Der Graph hat aber diesen Kontext nicht, und es fällt schwer alle denkbaren Kontexte im Wissensgraphen auszufordern.

Ein LLM kann helfen diese Nachteile auszugleichen. Es hat eine große Wissensabdeckung, auf Grund der umfangreichen Trainingsdaten. Einem explizit definierten Index kann man keine Fragen in natürlicher Sprache stellen. Explizit modellierte Informationen, die sich aus dem Kontext ergeben, wie der Bezug von „nice“ und „pesto“, kann das LLM inferrieren.

LLMs und Wissensgraphen: Forschung und Best Practices

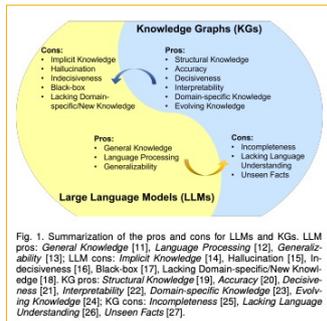
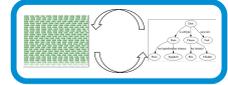


Fig. 1. Summarization of the pros and cons for LLMs and KGs. LLM pros: General Knowledge [11], Language Processing [12], Generalizability [13]. LLM cons: Implicit Knowledge [14], Hallucination [15], Indecisiveness [16], Black-box [17], Lacking Domain-specific Knowledge [18]. KG pros: Structural Knowledge [19], Accuracy [20], Decisiveness [21], Interpretability [22], Domain-specific Knowledge [23], Evolving Knowledge [24]. KG cons: Incompleteness [25], Lacking Language Understanding [26], Unseen Facts [27].

- Curt Cagle: [Nine ChatGPT Tricks for Knowledge Graph Workers](#)
- Dean Allemang: [AI's Woolf at the door – LLMs and Knowledge Graph, Figuring out vs. telling, LLM's Closing the KG Gap](#)
- Denny Vrandečić: [The Future of Knowledge Graphs in a World of Large Language Models](#)
- [Knowledge graphs and large language models](#) – chapter in Dagstuhl seminar report “Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century”, September 2022

Shirui Pan et al. [Unifying Large Language Models and Knowledge Graphs: A Roadmap](#) (2023)

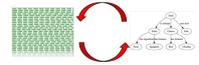
Public

23

Neben dem beschriebenen Artikel wurden in diesem Jahr viele Diskussionen veröffentlicht, welche die Ergänzung von Wissensgraphen und LLMs im Detail beschreiben.

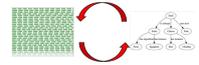
Was wir jetzt brauchen, sind Best Practices. Wie können Wissensgraphen und große Sprachmodelle voneinander profitieren? Welche Rolle spielen sie bei einer bestimmten Aufgabe? Wie kann man beide zusammen verwenden? In der Wissensgraphen-Gemeinschaft wurde in diesem Jahr eine Menge solcher Best Practices entwickelt. Vor allem die hier verlinkten Beiträge von Curt Cagle und Dean Allemang sind lesenswert.

Use Case getrieben vorgehen



- Braucht man LLMs und Wissensgraphen überhaupt?
- Szenarien in kleine Use Cases teilen: mit und ohne Wissensgraph oder LLM
- Kosten sind nach wie vor wichtig: Latenz und Euros
- Generell
 - Embeddings und Retrieval Augmented Generation (RAG) haben einen hohen Mehrwert, mit und ohne Wissensgraph
 - Durch die Nutzung von LLM Tools und Agenten kann Halluzination angegangen werden

Szenario: Retrieval im Kontext „Bibliothekskatalog“



DEUTSCHE NATIONAL BIBLIOTHEK

Kontakt | A-Z | Träger / Förderer | Datenschutz | Impressum | Hilfe | Mein Konto | English

KATALOG DER DEUTSCHEN NATIONALBIBLIOTHEK

Gesamter Bestand | Musikarchiv | Exilsammlungen | Buchmuseum

Suchformular zurücksetzen

nid=118540238 Finden → Expertensuche ?

Leichte Bedienung, intuitive Suche: Die Betaversion unseres neuen Katalogs ist online! → Zur Betaversion des neuen DNB-Katalogs

Noch nicht die passende Literatur gefunden? → Book a Librarian

Ergebnis der Suche nach: **nid=118540238**

Treffer 1 von 1

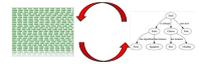
Link zu diesem Datensatz	https://d-nb.info/gnd/118540238
Person	Goethe, Johann Wolfgang von
Geschlecht	männlich
Andere Namen	Goethe, Johann Wolfgang (ADB)

Public

26

Als Szenario wählen wir passend zu dieser Veranstaltung den Bibliothekskatalog. Wir wollen sehen, in welchen Use Cases LLMs und Wissensgraphen Mehrwerte schaffen können.

Use Case „gezielte Suche“



- Suche nach ID „118540238“ braucht keine Wissensgraphen und LLM ☺
- Beispiel „Lebensdaten J.W.v. Goethe“: gesichertes Wissen kann man explizit ablegen, z.B. im Wissensgraph

KATALOG DER DEUTSCHEN NATIONALBIBLIOTHEK

Gesamter Bestand | Musikarchiv | Exilsammlungen | Buchmuseum

Suchformular zurücksetzen

nid=118540238 Finden → Expertensuche ?

Leichte Bedienung, intuitive Suche: Die Betaversion unseres neuen Katalogs ist online! → Zur Betaversion des neuen DNB-Katalogs

Noch nicht die passende Literatur gefunden? → Book a Librarian

Ergebnis der Suche nach: **nid=118540238**

Treffer 1 von 1

Link zu diesem Datensatz	https://d-nb.info/gnd/118540238
	Goethe, Johann Wolfgang von
	männlich
	Goethe, Johann Wolfgang (ADB)

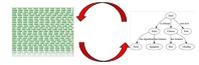
```
<https://d-nb.info/gnd/118540238> gndo:dateOfBirth "1749-08-28"^^xsd:date;
gndo:dateOfDeath "1832-03-22"^^xsd:date .
```

Public

27

Wenn man gezielt nach gesicherten Informationen einer ausgewählten Kategorie sucht, braucht man kein LLM. Hier ist ein Beispiel: wenn die ID „118540238“ für Goethe aus der DNB bekannt ist, reicht eine Datenbanksuche. Sie ist schnell, präzise und letztlich billig, im Vergleich zur LLM Verarbeitung. Weitergehende, gesicherte Informationen wie die Lebensdaten kann man auch im Wissensgraph ablegen. Dann wird das gesicherte Wissen leichter nachnutzbar.

Use Case „Suche via einfacher Wortvarianten“



Beispiel „Schriftsteller“

Tell me writing variants of Johann Wolfgang von Goethe in as many languages as you know. Store this variants in a JSON array. Each variant should consist of the variant writing and the name of the language in brackets. The name of the language should be written in English. The variants should also include non latin scripts, if needed.

LLM Prompt

LLM Ausgabe (ggf. mit Halluzinationen)

Johann Wolfgang von Goethe (Russian) – Ио́ханн Во́лфганг фон Гётте
 Johann Wolfgang von Goethe (Greek) – Γκέτε, Γίωργος Βόλφγκανγκ φου
 Johann Wolfgang von Goethe (Arabic) – يوهان فولفجانج فون غوته
 Johann Wolfgang von Goethe (Hebrew) – יוהאן וולףגאנג פון גוֹטֶה
 Johann Wolfgang von Goethe (Chinese Simplified) – 歌德, 约翰·沃尔夫冈·冯
 Johann Wolfgang von Goethe (Japanese) – ヨハン・ヴォルフガング・フォン・ゲーテ
 Johann Wolfgang von Goethe (Korean) – 요한 볼프강 폰 괴테
 Johann Wolfgang von Goethe (Hindi) – जोहान वोल्फगांग वॉन गोथे
 Johann Wolfgang von Goethe (Bengali) – জোহান বোলফগাং জন গটে

Sachbegriff	Schriftsteller
Quelle	M
Erläuterungen	Verwendungshinweis: Verwenden nur wo nicht vermeidbar; sonst s. Literatur; in der Regel wird mit dem Geographikum, nicht mit der Sprachbezeichnung verknüpft; mit einzelnen literarischen Gattungen wird in der Regel kein Kompositum gebildet, sondern mit der Gattungsbezeichnung verknüpft. Ggf. werden die vorhandenen Unterbegriffe sowie die entsprechenden weiblichen Bezeichnungen verwendet. Als Homonymenzusatz bei Personenschlagwörtern zugelassen.
Synonyme	Dichter Literaturproduzent Literat Homme de lettres

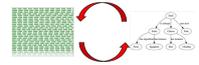
Varianten im WG „DNB Katalog“

Language	Label	Description	Also known as
English	writer	person who uses written words to communicate ideas and to produce written works	author writers authors penman penwoman
Japanese	著作家	職業のひとつ	著述家 著述業 文筆家 作家 文士
German	Schriftsteller	Person, die literarische Texte verfasst	Schriftstellerin Autor...

Varianten im WG „Wikidata“

Für allgemeine Entitäten findet man in öffentlich verfügbaren Wissensgraphen Wortvarianten, zum Beispiel für verschiedene Sprachen. Hier ist ein Beispiel für Goethe aus den Wissensgraphen Wikidata und dem DNB Katalog. Auch LLMs haben dieses Wissen. Hier sieht man eine Anfrage, einen LLM Prompt, um das Wissen zu erhalten. Die LLMs können bei allgemeinen Entitäten Lücken in den Wissensgraphen füllen. Die Gefahr von Halluzinationen ist aber gegeben. Die Anreicherung eines Suchindexes mit diesen Varianten erhöht also die Treffermenge, nicht aber die Präzision der Suche.

Use Case „Suche via semantischer Hierarchien“

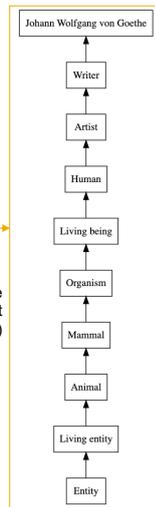


Beispiel „Konzepthierarchie“

I want to know something about concepts related to Johann Wolfgang von Goethe. I want to know all concepts that are broader than Johann Wolfgang von Goethe. For example, "human" is a broader concept, or "mammal" is an even broader concept. Create a list of these concepts. Have a maximum of 10 concepts. The most abstract concept should be "entity". Output the list as a file in the DOT language. Each concept should be encoded as a node. The broader concept relations should be encoded as arrows between the nodes.

LLM Prompt

LLM Ausgabe
(ggf. mit
Halluzinationen)



Ergebnis einer [DBpedia Query](http://dbpedia.org) zu übergeordneten Konzepten für „Goethe“

superclass

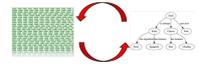
- <http://dbpedia.org/ontology/Person>
- <http://dbpedia.org/ontology/Animal>
- <http://dbpedia.org/ontology/Eukaryote>
- <http://dbpedia.org/ontology/Species>
- <http://dbpedia.org/ontology/Writer>

Public

29

Sowohl LLMs als auch Wissensgraphen können weitergehende Informationen für Retrieval bereitstellen. Hier sieht man einen LLM-Prompt um eine semantische Anreicherung mit Oberbegriffen zu erreichen. Man erhält eine Vielzahl von Begriffen, die wieder nicht alle relevant sind, wie „Mammal“. Die Anfrage an den DBpedia Wissensgraph erhält weniger Ergebnisse, dafür mit höherer Relevanz.

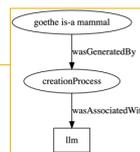
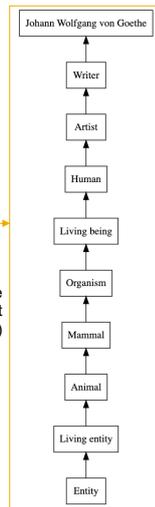
Exkurs: Wissensgraphen mit LLM erweitern



Beispiel „Konzepthierarchie“

LLM Prompt
I want to know something about concepts related to Johann Wolfgang von Goethe. I want to know all concepts that are broader than Johann Wolfgang von Goethe.

LLM Ausgabe
(ggf. mit
Halluzinationen)



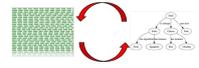
Erweiterung inkl.
Provenienz-
Informationen

Ergebnis von [DBpedia Query](#) zu übergeordneten Konzepten für Goethe

superclass
http://dbpedia.org/ontology/Person
http://dbpedia.org/ontology/Animal
http://dbpedia.org/ontology/Eukaryote
http://dbpedia.org/ontology/Species
http://dbpedia.org/ontology/Writer

Hier ein Exkurs zum Thema „Erweiterung von Wissensgraphen durch LLMs“. Das beschriebene Beispiel, mit LLMs Konzepthierarchien zu erzeugen, könnte dafür genutzt werden. Das gefährdet die Qualität der gesicherten Informationen. Eine Vorgehensweise ist bei der Erweiterung die Herkunft der Information explizit zu machen. Hier sieht man wie die Aussage „Goethe is-a mammal“ als durch ein LLM generiert gekennzeichnet werden kann. Das erlaubt bei weiteren Prozessen dann die ausgewählte Behandlung, z.B. Filterung von durch LLMs erzeugten gesicherten Informationen.

Exkurs: Wissensgraphen mit LLM erzeugen



System Prompt

You are an expert agent specialised in analysing person descriptions. Your task is to identify the entities and relations requested with the user prompt, from a given person description specification.

User Prompt

Use the following entity types:

```
# ENTITY TYPES:
entity_types = {
  "person": "https://schema.org/Person",
  "country": "https://schema.org/Country"
}
```

Use the following relation types:

```
relation_types = {
  "name": "https://schema.org/name",
  "birthPlace": "https://schema.org/birthPlace",
  "birthDate": "https://schema.org/birthDate",
  "deathDate": "https://schema.org/deathDate",
  "description": "https://schema.org/description"
}
```

Example input: ...
Example output: ...

(1) LLM Prompt

(2) LLM Ausgabe

```
{
  "@context": "https://schema.org",
  "@type": "Person",
  "name": "Johann Christoph Friedrich von Schiller",
  "birthPlace": "Germany",
  "birthDate": "1759-11-10",
  "deathDate": "1805-05-09",
  "description": "Johann Christoph Friedrich von Schi"
}
```

(3) Validierung der Ausgabe gegenüber Wissensgraph-Definitionen

The screenshot shows the Schema.org validator interface. On the left, the JSON-LD output from the LLM is pasted into the input field. On the right, the validator displays the resulting schema object for a 'Person' type. The schema includes fields for '@type', 'name', 'birthDate', 'deathDate', 'description', and 'birthPlace'. The 'description' field contains a detailed text about Johann Christoph Friedrich von Schiller.

Property	Value
@type	Person
name	Johann Christoph Friedrich von Schiller
birthDate	1759-11-10
deathDate	1805-05-09
description	Johann Christoph Friedrich von Schiller was a German poet, playwright, historian, philosopher, and physician. Schiller is considered by most Germans to be Germany's most important classical playwright.
birthPlace	Place
@type	Place
name	Germany

Beschreibung des Vorgehens: Automated Knowledge Graph Construction using ChatGPT

Vorteile der Nutzung standardisierter Wissensgraph-Vokabulare wie Schema.org:

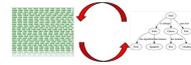
- LLMs haben viele Webinhalte mit den Vokabularen verarbeitet (vgl. Statistik unter [Webdatacommons](#)) → erhöht Qualität der Ausgabe
- Es gibt Tools zur Weiterverarbeitung (z.B. Validierung)

Public

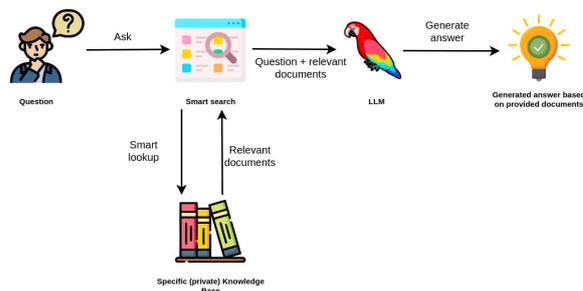
31

Mit LLMs lassen sich auch Wissensgraphen erzeugen. Hier ist ein Beispielworkflow, in dem die Prompts und insbesondere das Bereitstellen von Beispielen eine große Rolle spielen.

Use Case „(Dokument)daten mit LLM durchsuchen“: Retrieval Augmented Generation (RAG)



- Speichern interner (Dokument)quellen in Vektordatenbank



Quelle: [Knowledge Graphs & LLMs: Fine-Tuning Vs. Retrieval-Augmented Generation](#)

„Finde deutsche Schriftsteller die im 18. Jahrhundert in Frankfurt am Main geboren wurden“

„Finde deutsche Schriftsteller die im 18. Jahrhundert in Frankfurt am Main geboren wurden und die einen Kollegen aus Marbach am Neckar haben“

Abfrage via LLM, Ergebnis 😊

Abfrage via LLM, kein Ergebnis 😞

KATALOG DER DEUTSCHEN NATIONALBIBLIOTHEK

Gesamter Bestand | Musikarchiv | E-Kollaborationen | Buchmuseum

Suche:

Ergebnis der Suche nach: **nid=118540238**

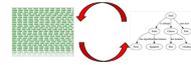
Treffer 1 von 1

Über die Deutsche Nationalbibliothek	Link zu diesem Datensatz	https://nbn-resolving.org/urn:nbn:de:hbz:5:1-118540238
Personen	Personen	Goethe, Johann Wolfgang von
Geschlechte	Geschlechte	infrisch
Andere Namen	Andere Namen	Goethe, Johann Wolfgang (ADB)

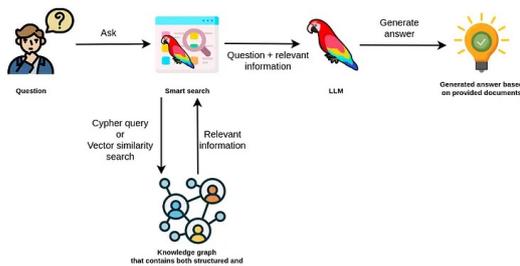
Ein immer häufiger auftretendes Muster in der Nutzung von LLMs ist Retrieval Augmented Generation. Zumeist Textdokumente oder auch einfache strukturierte Datensätze werden als Embeddings in einer Vektordatenbank gespeichert. Eine Nutzeranfrage kann dann gegenüber den Embeddings verarbeitet und das oder die passenden Dokumente gefunden werden. Die Anfrage „Finde deutsche Schriftsteller die im 18. Jahrhundert in Frankfurt am Main geboren wurden“ wird dann wahrscheinlich ein Ergebnis liefern, da die nötigen Informationen in einem Dokument bzw. Datensatz zu finden sind.

Was man damit nicht erreichen kann: Anfragen erfolgreich stellen, die auf miteinander verknüpften Informationen aus verschiedenen Quellen beruhen. „Finde deutsche Schriftsteller die im 18. Jahrhundert in Frankfurt am Main geboren wurden und die einen Kollegen aus Marbach am Neckar haben“. Diese Anfrage müsste Informationen aus dem Datensatz über Goethe und über Schiller berücksichtigen. Diese semantischen Verknüpfungen sind aber nicht in den Vektorembdings gespeichert.

Use Case „(Dokument)daten mit LLM durchsuchen“: RAG mit Wissensgraphen & WG Queries



- Speichern interner (Dokument)quellen in Vektordatenbank
- Verknüpfen mit Wissensgraphen



Quelle: [Knowledge Graphs & LLMs: Fine-Tuning Vs. Retrieval-Augmented Generation](#)

„Finde deutsche Schriftsteller die im 18. Jahrhundert in Frankfurt am Main geboren wurden und die einen Kollegen aus Marbach am Neckar haben“

Abfrage via LLM

Ergebnis

KATALOG DER DEUTSCHEN NATIONALBIBLIOTHEK

Gesamter Bestand | Musikarchiv | E-Sammlungen | Buchmuseum

Suche: nid=118540238

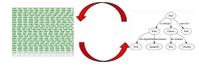
Ergebnis der Suche nach: **nid=118540238**

Treffer 1 von 1

Über die Deutsche Nationalbibliothek	Link zu diesem Datensatz	https://d-nb.info/gnd/118540238
Person	Person	Goethe, Johann Wolfgang von
Geschlecht	Geschlecht	männlich
Andere Namen	Andere Namen	Goethe, Johann Wolfgang (ADB)

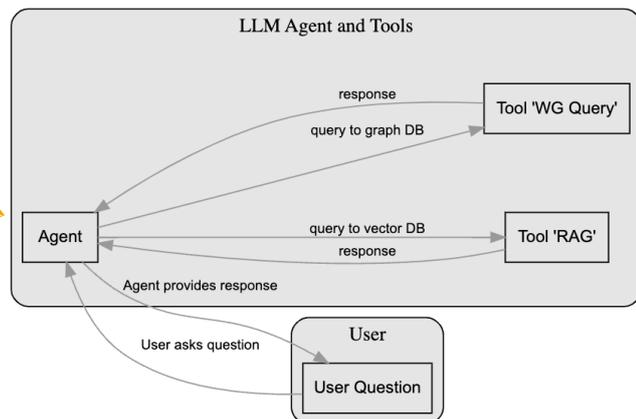
Eine derartige Funktionalität lässt sich durch die Verbindung von Retrieval in der Vektordatenbank und Verarbeitung im Wissensgraphen erzielen. Durch die Suche in der Vektordatenbank können die Entitäten wie „Goethe“ oder „Schiller“ erfasst werden. Im Wissensgraph kann dann für diese Entitäten eine Query ausgeführt werden, welche Information wie die Geburtsorte der beiden Schriftsteller berücksichtigt.

Use Case „Verarbeitung dynamisch gestalten“: LLM Tools & Agenten



„Finde deutsche Schriftsteller die im 18. Jahrhundert in Frankfurt am Main geboren wurden und die einen Kollegen aus Marbach am Neckar haben“

„Johann Wolfgang von Goethe“



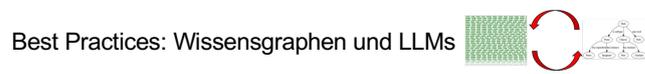
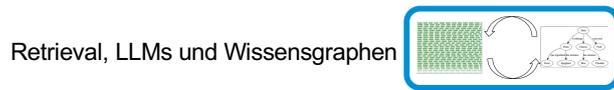
Public

34

Eine derartige Kombination von LLM und Wissensgraphen ist sehr mächtig. Die Herausforderung ist aber, dass es vor einer Anfrage nicht vorhergesagt werden kann welche Verarbeitung nötig ist. Hierfür gib es im Kontext von LLM zwei relevante Mechanismen: Tools und Agenten.

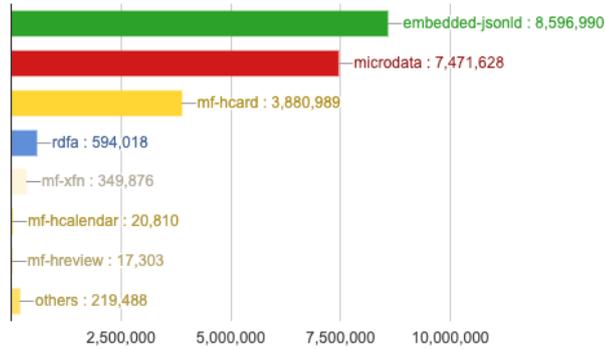
Tools ermöglichen es den LLMs, externe Verarbeitungen anzustoßen. Diese können beliebig sein, zum Beispiel eine Websuche oder die Abfrage einer Datenbank. Agenten erlauben den dynamischen Aufruf von Tools. Auf dieser Folie ist skizziert, welche Tools für die vorher beschriebene, komplexe Frage über Goethe und Schiller hilfreich sein können. Das sind dann ein Tool für Retrieval Augmented Generation, und ein Tool für eine Abfrage von Wissensgraphen. In welcher Reihenfolge die Tools aufgerufen werden und wie ihre Zwischenausgabe weiterverarbeitet wird, entscheidet der Agent.

Überblick



Web Data Commons Data Sets – October 2022

Domains with Triples



**Zum Schluss ein Aufruf: veröffentlichen Sie ihre Daten FAIR*
Sie tragen zur Qualität von LLMs bei 😊**

* <https://www.go-fair.org/fair-principles/>

Public

37

Zum Schluss ein Aufruf, insbesondere an die Bibliothekscommunity. Veröffentlichen Sie Ihre Daten im Web FAIR. LLMs beruhen zu einem großen Teil auf Webinhalten, inkl. Wissensgraphen. Diese fließen also auch in LLMs ein. Sie tragen so zur Qualität von LLMs bei. Um mit den Worten von Denny Vrandečić zu schließen: in einer Welt unbegrenzter Mengen von Inhalten ist Wissen sehr wertvoll.

