



KALEIDOSCOPE  
Taking your content global

# Terminologie in der KI

Wie mit Terminologie der Output von  
LLMs und GenAI optimiert werden kann

Klaus Fleischmann, Christian Lang  
Kaleidoscope GmbH, 2025 03 29





# Agenda

1. Grundlagen
2. Prompt Engineering?
3. RAG?
4. Oder doch TAG?
5. Auswirkungen auf die Terminologearbeit
6. Fazit und Ausblick



# 01 Grundlagen

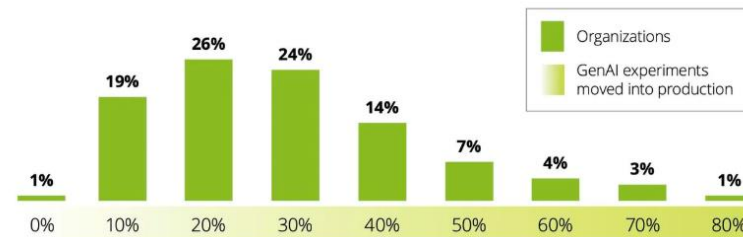
# KI – Der aktuelle Stand in Unternehmen



BCG

AI Adoption in 2024: 74% of Companies Struggle to Achieve and Scale Value

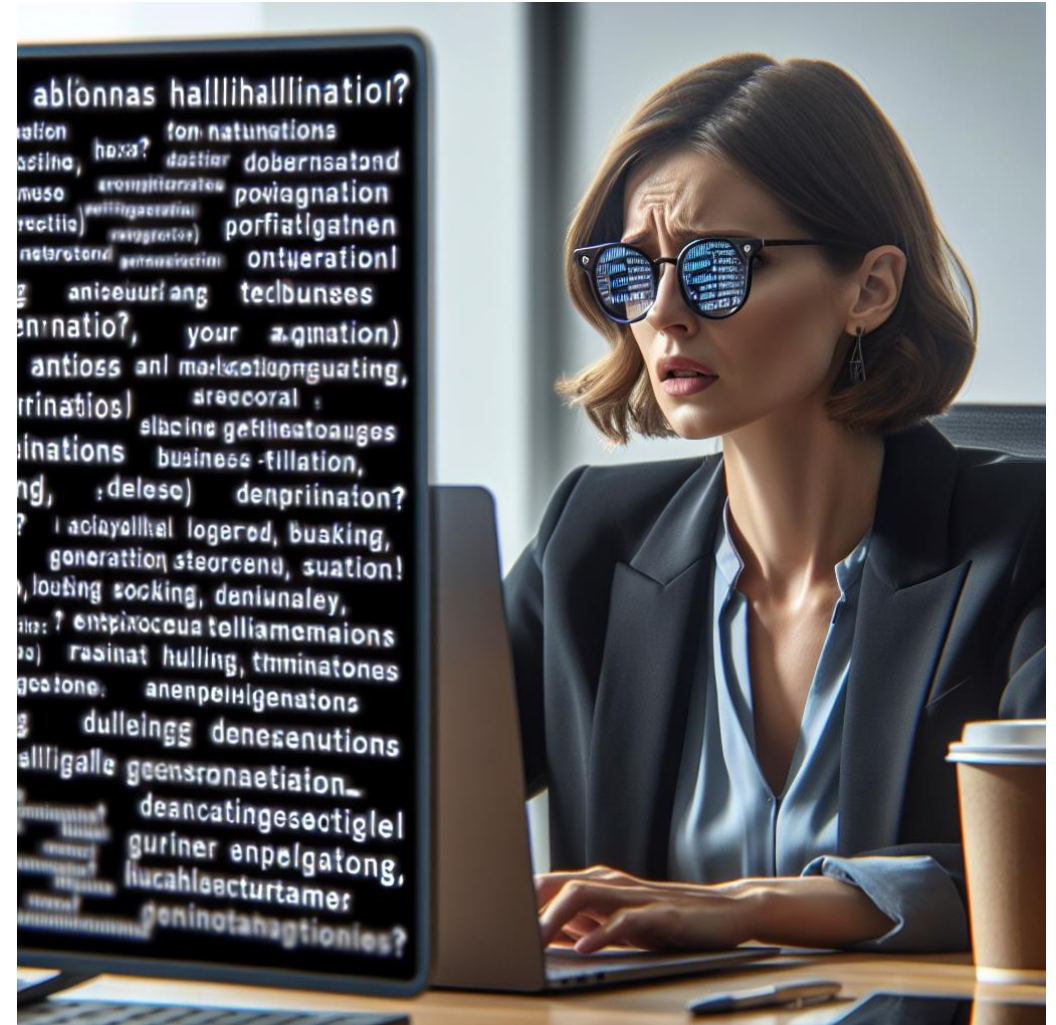
A large majority of organizations have deployed less than a third of their GenAI experiments into production



Deloitte Consulting LLC

McKinsey & Company

Therein lies the challenge: the long-term potential of AI is great, but the short-term returns are unclear.



# KI – Der aktuelle Stand in Unternehmen



- Eigene LLMs zu trainieren ist aktuell nicht vertretbar
- Fertigen Modellen fehlt aber Wissen über das Unternehmen
- Daher halluzinieren sie bei fehlendem Kontext
- Und natürlich fehlt ihnen unsere spezifische Unternehmenssprache
- Das ist frustrierend für Anwender und Entwickler

# Aber...



“Soon after the first automobiles were on the road, there was the first car crash. But we didn’t ban cars—we adopted speed limits, safety standards, licensing requirements, drunk-driving laws, and other rules of the road.”

– Bill Gates, cofounder of Microsoft

Terminologie kann eine äußerst effektive “Leitplanke” auf dem KI-Highway sein.

– Klaus Fleischmann, founder of Kaleidoscope



# Was ist eigentlich „KI“



KLETTERZENTRUM INNSBRUCK  
Bergsport findet Stadt

10:15 - 10:45

Sprachbezogene Künstliche Intelligenz in Form  
von Large Language Models: Funktionsweise –  
fachkommunikatives Anwendungsspektrum –  
Kompetenzanforderungen

Ralph Krüger  
(TH Köln)





# 02 Prompt Engineering

# Was ist Prompt Engineering?



- "A prompt is a form of guidance or instruction provided to elicit a specific response or action."
  - Source: OpenAI GPT-4o
- GenAI LLMs werden meist mit der folgenden (chat-) Struktur fine-tuned:
  - System message (oder "System instructions" oder "Developer message" oder ...)
  - User message
  - Assistant message
  - (Tool oder Function)
- Gute Anleitungen für's Prompting:

<https://platform.openai.com/docs/guides/prompt-engineering>

<https://www.promptingguide.ai/techniques>

System message

```
You are a terminologist, tasked with replacing wrong terminology in the %LANGUAGE% text provided by the user. Replace the terms before || with the terms after ||. Example:

Terms to be replaced: cat||dog
runs||walks
Sentence to be corrected: "The cat runs quickly."
Expected output: "The dog walks quickly."

For some languages you might be provided with the gender in parentheses, make sure to use the correct article and grammar if it changes with the replacement.
Example:
Terms to be replaced: Maschine (feminine)||Rechner (masculine)
Sentence to be corrected: "Schalten Sie die Maschine ab."
Expected output: "Schalten sie den Rechner ab."

The user might provide additional instructions before the replacement terms and input sentence.
```

## System message für Termersetzung

Prompt

```
# Correct the sentence to be corrected below. Only return the correction. Before returning:
* Make sure all occurrences of the terms have been replaced.
* Adapt the sentence syntax to the replaced terms.
* Correct grammatical errors related to the replaced terms, but do not change anything else in the sentence.
* Avoid adding or deleting words in the sentence.
* Avoid changes beyond strictly necessary for sentence grammar.

Terms to be replaced: %SOURCETERM% || %TARGETTERM%

Sentence to be corrected: %SOURCETEXT%
```

## User message für Termersetzung

# Ein paar Beispiele



```
System Prompt
You are a translator and author. The user will provide text to be translated and indications which terminology to use.

# Task description
* Translate the text provided by the user from and into the language the user specifies.
* Make sure the translation sounds natural.
* The user specifies the translation direction by prefixing the text to be translated with the following string: `Translate {sourceLanguage} to {targetLanguage}`
* Use the definition to disambiguate the meaning of the terminology passed by the user and translate accordingly
* Only return the translation

# Terminology
* If available, the user will provide indication on what terminology to use
* Follow the suggestions provided within the <tag>-XML elements of the user message

## Terminology format
* The terminology will be provided in the format below within the <tag>-XML Elements of the user message:
```markdown
## Concept 1
* Definition of concept 1
### source term 1
#### Possible translations:
1. first possible translation
   * Usage note of possible translation 1
2. second possible translation
   * Usage note of possible translation 2
### source term 2
#### Possible translations:
1. first possible translation
   * Usage note of possible translation 1
2. second possible translation
   * Usage note of possible translation 2

## Concept 2
### source term 2
#### Possible translations:
1. first possible translation
...
* Note: not all terms will have a definition or usage note.

# Rules
* Use the definition to disambiguate the meaning of term pairs
* Follow the usageNote of each possible translation to choose the most suitable translation, if more than one translation is provided
* If the system returns a term that is not present in the source text, ignore the term.
```

## System message Übersetzung mit Anweisungen für TAG

```
System Prompt
Extract key terminology from the given text and present it in a markdown table format with the following columns: **Term**, **Definition**, **Context**.

# Steps
1. Analyze the provided text and identify key terminology that is domain-specific, technical, or central to the content.
2. Extract all unique terms that are relevant to the subject matter.
3. Pick one sentence from the original text verbatim to provide the **Context** for each term.
4. Present the extracted terminology in a markdown table.

# Output Format
The output should be a markdown-formatted table, with the following columns:

- **English term**: The key term extracted.
- **Context**: One sentence from the original text in which the term appears.
- **Definition**: A brief definition if available in the text; otherwise, leave blank.

- **Markdown Table Example:**

| Term | Definition | Context |
|-----|-----|-----|
| Term 1 | Brief definition or empty | Sentence containing the term |
| Term 2 | Brief definition or empty | Sentence containing the term |

- Always output the markdown directly in the text, not as a separate code window.

## Rules for Term column
- Ensure the terminology is unique and directly relevant to the text's subject.
- Always extract the terminology in dictionary form (usually infinitive for verbs and singular for nouns); i.e. if the term is "Large Language Models" extract "Large Language Model"
- Make sure to extract ALL terms present in the text
- Make sure to also include product names as terms

## Rules for Definition
- Aim for clarity and conciseness in the definitions if provided.
- Never mention the term itself or the synonym in the definition.
- Start the definition without using an article.
- If no definition is available, leave the **Definition** field empty.
```

## System message Termextraktion



- Eingeschränkte Token-Zahl
  - Neuere Modelle können mittlerweile bis zu 128k oder sogar 1 Mio. Tokens verarbeiten.
- Zu viele Anweisungen/Kontext verwirren das Modell
- Lange Kontexte sind teuer/sehr ineffizient und verschlechtern Performance
  - Lost in the Middle: How Language Models Use Long Contexts (Liu et al., TACL 2024)
  - Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG (Jin et al. 2025)



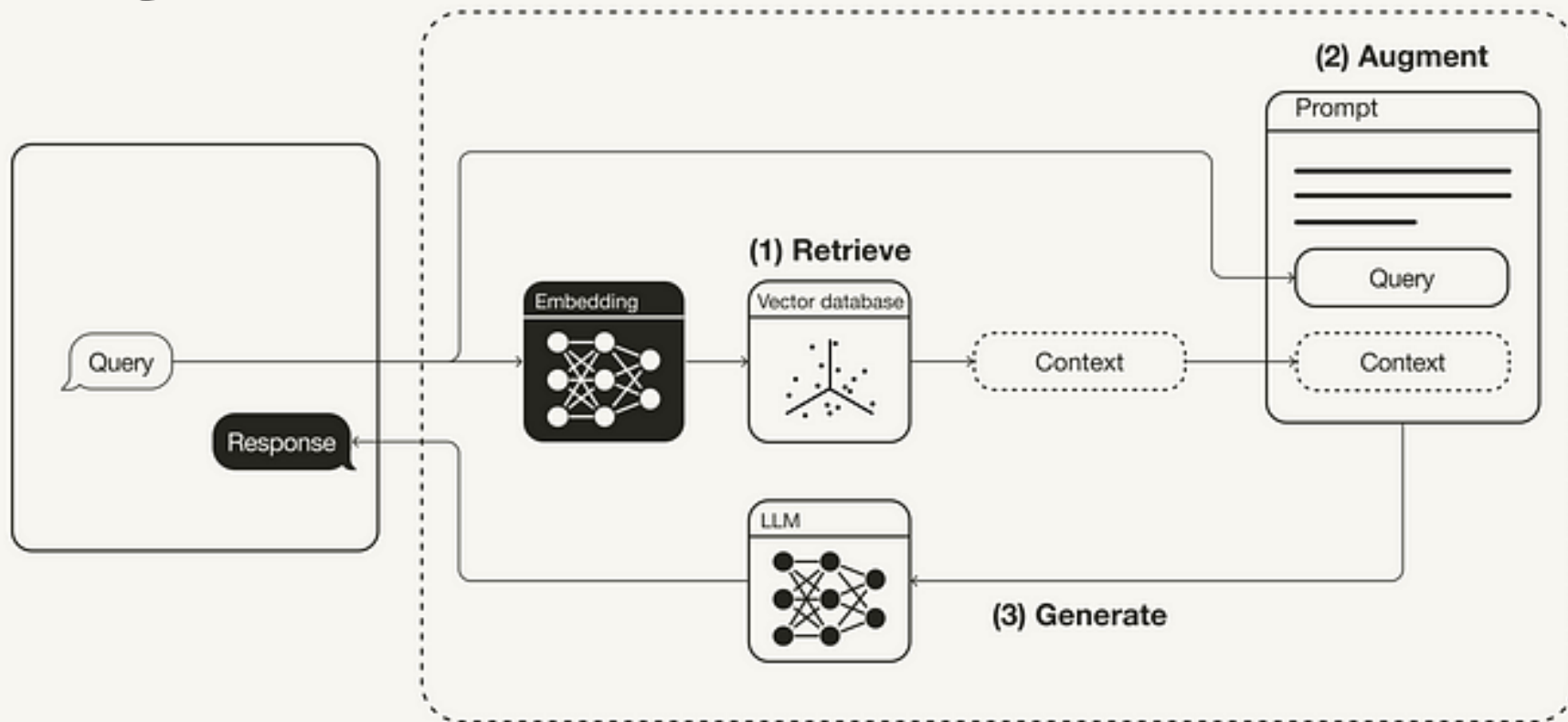
# 03

## Retrieval Augmented Generation (RAG)

# RAG - Der Teleprompter für die KI



## RAG



<https://towardsdatascience.com/retrieval-augmented-generation-rag-from-theory-to-langchain-implementation-4e9bd5f6a4f2>



# Nachteile von RAG für Terminologie



- Chunking
    - Wie soll Terminologie hochgeladen werden?
    - Wie kann ich sicherstellen, dass die besten „Chunks“ ausgewählt werden?
    - Cut-off führt zu unvollständigen Retrievals
  - Langsam und ineffizient
    - Der Retrieval-Prozess ist relativ langsam und rechenintensiv
    - Kein Echtzeit-Zugriff auf terminologische Daten
    - Hoher Tokenverbrauch, da irrelevante Chunks mitgegeben werden
  - Retrieval ist schwer zu steuern
    - Nicht ohne weiteres möglich zu filtern
    - Schwer zu erörtern, welche Information von dem Modell tatsächlich genutzt wurde
    - Chunks werden mitunter in unsinnige Einheiten geteilt
    - Schwer zu Troubleshooten
    - Nicht vorhersehbar und etwas unzuverlässig
- Für IT und Entwicklung schwierig

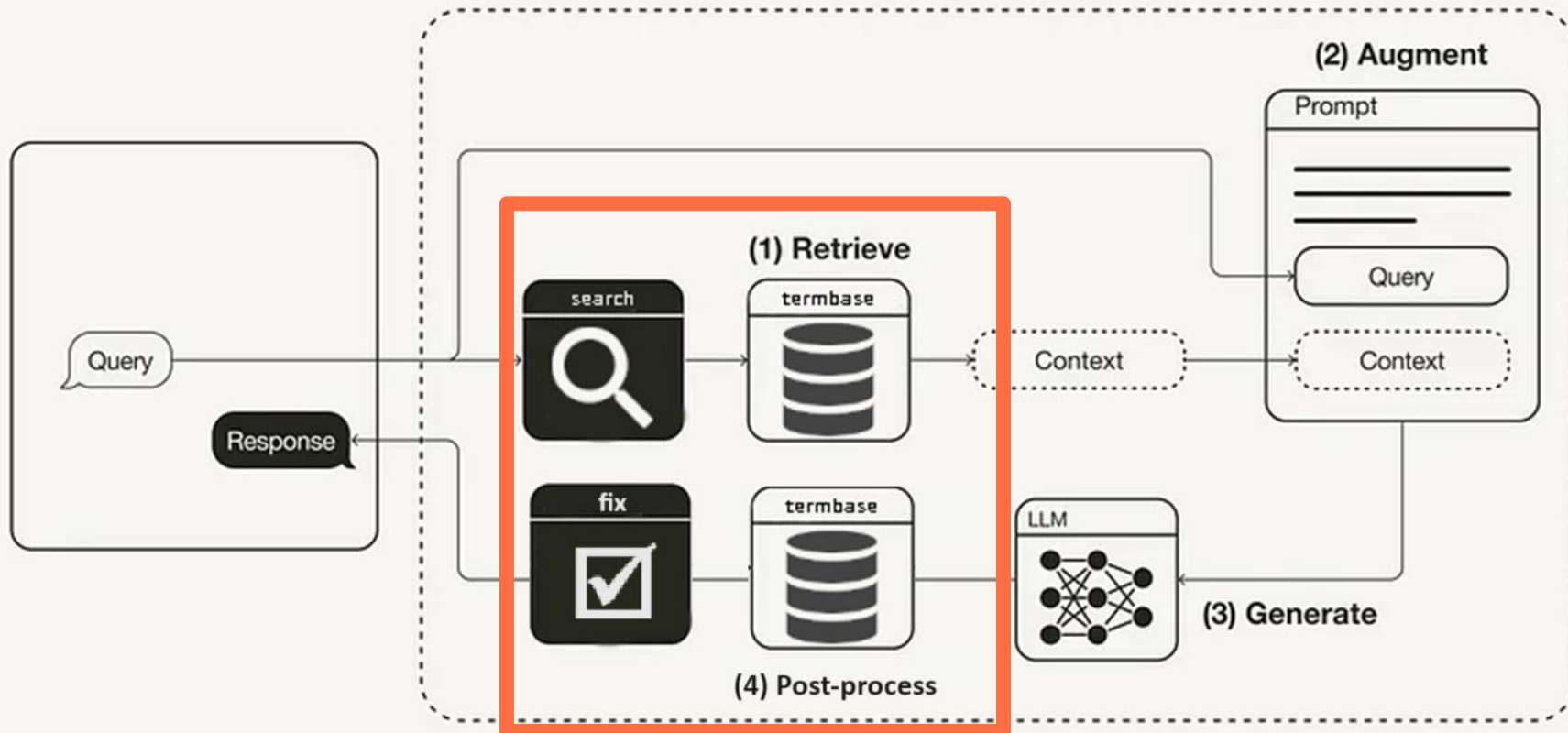


# 04 TAG

# Introducing TAG- Terminology Augmented Generation



## TAG



# Was ist TAG



## Strukturiertes Datenmodell

Termbanken verfügen über ein präzises Datenmodell, das relevante Informationen sauber strukturiert und abgreifbar speichert.

Daten

## Präziser Zugriff

Klassische Methoden der Terminologie ermöglichen deterministischen Zugriff mit genauer Suche, Filtermöglichkeit etc.

Zugriff

## Flexible Formate

LLMs benötigen leicht strukturierte Daten. Dies ist aus Termbanken erzeugbar. Beispiele sind Markdown, JSON aber auch Prosa.

Formate

## API-Zugriff in Echtzeit

Per API können Terminologie-Suchen sehr schnell und direkt erfolgen, statt langsamen Zugriff auf zuvor embeddete Inhalte.

Echtzeit



# TAG

Zum Nachlesen / Nachsehen:  
<https://youtu.be/SVzYEcPFJMs>

# Beispiele LLM-freundliche Formate



## Markdown (Use-Case Übersetzen)

```
* · Source · Term · 1 · --> · Target · Term · 1 LF
—> * · Definition: · Definition · of · concept LF
—> * · Usage · note: · Usage · note LF
* · Source · Term · 2 · --> · Target · Term · 2 LF
—> * · Definition: · Definition · concept LF
* · Source · Term · 3 · --> · Target · Term · 3 LF
```

## JSON/JSONL (Use-Case Korrektur)

```
{"concept1": {"forbidden · term · 1": "allowed · term · 1", LF
"Definition": "Definition · of · concept · 1"}} LF
{"concept2": {"forbidden · term · 2": "Allowed · term · 2", LF
"Definition": "Definition · of · concept · 2", LF
"grammaticalGender": "Grammatical · Gender · allowed · term"}} LF
```

## YAML (Use-Case Generierung)

```
-- Concept1: Preferred · term2 LF
  -- Definition: Definition · of · concept LF
  -- Admitted synonyms: Admitted · term1 | SEO · term1 LF
  -- Forbidden terms: Forbidden · synonyms LF
-- Concept2: Preferred · term2 LF
  -- Definition: Definition · of · concept LF
  -- Admitted synonyms: SEO · term2 LF
  -- Forbidden terms: Forbidden · synonyms
```

## Prosa (Use-Case NMT)

In neueren NMT Systemen kann Kontext zur Desambiguierung mitgegeben werden, zB:

```
Translate 'Welle' as 'shaft' if it refers to the element transmitting power
from the engine to the propeller.
Translate 'Welle' as 'wave' if it refers to the movement of the ocean
```

# Vorteile von TAG



- TAG ist effizient in der Verarbeitungsgeschwindigkeit und im Tokenverbrauch, was es schneller und kostengünstiger macht als RAG.
- TAG greift in Echtzeit auf aktuelle Terminologie zu.
- TAG ermöglicht präzisen und deterministischen („terminologischen“) Zugriff auf die Daten, mit Suchmethoden, Analysemethoden, Filtern etc.
- TAG kennt das Metadatenmodell der Termbank
- TAG kann extrem flexible Zielformate zurückgeben
- TAG ist einfach zu implementieren und für Anwender leicht verständlich.
- TAG schlägt die Brücke zwischen Terminologie-Methoden und KI-Entwicklung.



# 05 Auswirkungen auf die Terminologiewerk

# Welche Auswirkungen hat das auf mein Terminologiemanagement?



- Die kurze Antwort: KEINE
- Im Gegenteil: Endlich haben wir eine gute Begründung warum wir immer Wert gelegt haben auf saubere Daten und Dinge wie Definitionen, Kontextsätze, Verwendungsstatus etc.
- Es gelten die üblichen Regeln für Terminologie
- Die lange Antwort: FAST Keine



# Beobachtungen zu hilfreichen Feldern



Konzept-Ebene	Sprach-Ebene	Term-Ebene
Administrative Information	Workflow-Status	<b>Verwendungstatus</b>
<b>Definition (+Quelle)</b>	<b>Definition (+ Quelle)</b>	<b>Verwendungshinweis</b>
<b>Fachgebiet</b>	<b>Erklärung</b>	Wortart
Anmerkung	Anmerkung	<b>Grammatikinfos</b>
<b>Beziehung</b>	Abbildung	<b>Kontext</b>
		<b>Beispiele</b>
		Anmerkung
		Quelle
Hinweis für KI?	Hinweis für KI?	Hinweis für KI?

# Beobachtungen zu Metadaten



- Englisch für Feldnamen und Feldwerte, in natürlicher Sprache
- Klassifikationen:
  - Bestehende Taxonomien verwenden
  - „Internas“ und Akronyme vermeiden
- Verwendungsstatus
  - Eindeutige und verständliche Werte
  - Standards verwenden
- Grammatikinfos
  - Z.B. feminin, maskulin, neutrum
  - Oder den Artikel anführen („der“, „die“, „das“)
- Term-Typen sind hilfreich
  - Akronym, Kurzform etc.

	Oncology
	Immunology
	Anesthesiology
	Radiology
	Emergency medicine
	Internal medicine
	Neurology

	Admitted
	Preferred
	Deprecated

	Feminine
	Masculine
	Neuter



# 06 Fazit und Ausblick

# Terminologie ist die Grundlage



- Terminology is the very foundation of all semantic science and semantic data. Particularly in LLMs where we are mainly dealing with words. **Terminology is the source of truth for working with words.**
- If we build a semantic stack like taxonomies or ontologies without a solid foundation in terminology, it will fall apart like a house of cards.

Michael Iantosca  
[www.thinkingdocumentation.com](http://www.thinkingdocumentation.com)



- KI im Unternehmen muss übergreifend gedacht werden.
- Terminologen können extrem hilfreiche Daten beisteuern, idealerweise mit TAG (Auch die Technische Dokumentation kann ideal vorbereitete Daten für RAG liefern)
- Terminologie ist eine ausgezeichnete Basis für Knowledge Graphs oder andere Semantic Stacks, idealerweise sind sie sogar integriert. Suchen wir diese Teams!
- KI verbessert den Business Case für Terminologie:
  - Reduzierte Aufwände für Terminologearbeit.
  - Höherer Nutzen der Terminologie.
  - Früherer und vor Allem höherer ROI

# Noch Fragen?



Klaus Fleischmann

CEO Kaleidoscope GmbH &  
Eurocom Translation Services GmbH

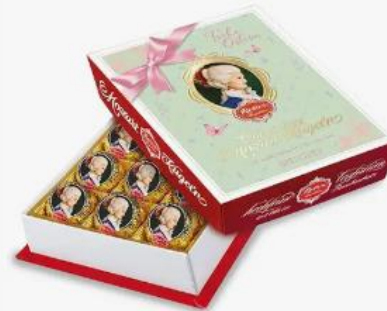


Christian Lang

Connecting language and  
technology



A **Mozartkugel** (German: [ˈmoːtʰsʰʌʁtˌkuːɡl]  <sup>ⓘ</sup>; English: "Mozart ball"; pl. *Mozartkugeln*) is a small, round **sugar confection** made of **pistachio**, **marzipan**, and **nougat** that is covered with **dark chocolate**. It was originally known as *Mozart-Bonbon*, created in 1890 by **Salzburg** confectioner **Paul Fürst** (1856–1941) and named after **Wolfgang Amadeus Mozart**. Handmade *Original Salzburger Mozartkugeln* are manufactured by Fürst's descendants up to today, while similar products have been developed by numerous confectioners, often industrially produced.



Sponsored ⓘ

Reber, Constanze Mozart Balls  
Baroque, Pack of 12, Mozart  
Balls Made of Alpine Milk an...

Chocolate

3.6  (7)

€12<sup>96</sup> (€54.00/kg)

€5.00 Direct debit code, valid 48  
hours

 prime

FREE delivery **Mon, 31 Mar**

