

Wissensrepräsentation - Überblick

- Ontologien
- Knowledge graphs
- Symbolische KI

Was ist Wissen?

Um zu verstehen, was Wissen in einem KI-Kontext ist, müssen wir die Differenz zwischen einigen verwandten Konzepten unterscheiden:

- Daten - rohe Fakten oder Zahlen
- Informationen - Daten mit Kontext
- Wissen - zusammenhängende Fakten und Schlussfolgerungen

Daten

Bei den Daten kann es sich um Sensormesswerte, numerische Werte, Pixel-RGB-Werte usw. Handeln. Beispiel:

- “Temperatur”, 20, “Lüfter”, 1, “Zeit”, 08:30...

Dies sind rohe Fakten oder Daten.

Information

Sobald den Daten ein gewisser Kontext gegeben wird, werden sie zu Informationen.
Hier sind einige Beispiele:

- „Die um 08:30 Uhr gemessene Temperatur betrug 20,0 Grad“
- „Das Ventilator wurde um 08:30 Uhr eingeschaltet“

Wissen

Wissen wird aus Daten abgeleitet, indem Fakten durch Schlussfolgerungen miteinander verbunden werden. Beispiel:

- „Wenn die Temperatur 20 Grad erreicht, schalten Sie den Ventilator ein“

Datenrepräsentation (DR)

Die Datenrepräsentation (DR) hat das Ziel, rohe Fakten zu speichern und zu bearbeiten. DR hat eine niedrige Ebene, da es mit Bits, Strings und Tabellen arbeitet. Daten können in vielen Formaten gespeichert werden, darunter CSV, JSON und Binärdateien. DR wird in der Datenbankentwicklung, bei Sensoren und APIs eingesetzt.

Wissensrepräsentation (WR)

Die Wissensrepräsentation (WR) hat das Ziel, auf der Grundlage von Daten Schlussfolgerungen (Reasoning) und Verständnis zu ermöglichen. WR ist hochrangig, da sie sich mit Konzepten, Regeln und Logik befasst. Wissen kann in vielen Formen dargestellt werden, darunter Graphen, logische Regeln, Symbole und neuronale Gewichte. WR ist ein grundlegender Bestandteil der KI und wird in Expertensystemen, Chatbots, KI-Agenten und Large Language Models (LLMs) verwendet.

DR vs. WR

Analogie:

- DR – eine Bibliothek mit Büchern (rohe Fakten/Infos)
- WR – eine Bibliothekar der weiß was in die Bücher steht, und kann Fragen beantworten

Aspekt	DR	WR
Ziel	Rohe Fakten speichern und verarbeiten	Schlussfolgerungen (Reasoning) und Verständnis ermöglichen
Level	Low-Level (Bits, Strings, Tabellen)	High-Level (Konzepte, Regeln, Logik)
Format	CSV, JSON, binary	Graphen, logische Regeln, Symbole, neuronale Gewichte
Verwendungen	Datenbanken, Sensoren, APIs	Expert systems, Chatbots, KI-Agenten

Beispiele

DR im CSV-Dateiformat:

Wert, Sensor/Aktor, Zeit

“20”, “Temperatur”, “08:30”

“1”, “Lüfter”, “08:30”

WR im Logik:

T \rightarrow **L**

“Wenn der Temperatur 20 Grad erreicht, dann schalten die Lüfter an”

Ontologien

Definition: Eine formale Darstellung von Wissen als eine Menge von Konzepten und die Beziehungen zwischen ihnen.

Zweck: Ermöglichung eines gemeinsamen Verständnisses, semantischer Schlussfolgerungen und Interoperabilität.

In der Informatik und KI definiert eine Ontologie:

- Welche Arten von Dinge es gibt
- Wie sie sich zueinander verhalten

Bestandteile von Ontologien

Ontologien bestehen aus:

- **Klassen** sind Kategorien oder Arten von Dingen.
- **Instanzen** (Individuals) sind die tatsächlichen Dinge, die zu Klassen gehören - Beispiele aus der realen Welt.
- **Eigenschaften** (Relationen) definieren, wie sich Instanzen zueinander verhalten oder welche Eigenschaften sie haben.

Beispiel - Klassen

Klassen sind abstrakte Konzepten von Dinge:

- Person
- Stadt
- Universität
- Sensor

Beispiel - Instanzen

Instanzen sind die „Datenpunkte“ Ihrer Ontologie.:

- Hendrik (eine Instanz von Person)
- Berlin (eine Instanz von Stadt)
- HHN (eine Instanz von Universität)
- Temperatursensor (eine Instanz von Sensor)

Beispiel - Eigenschaften

Es gibt zwei Haupttypen von Eigenschaften:

- Objekt-Eigenschaften - Beziehen eine Instanz auf eine andere
Beispiel: `studiertAn(Hendrik, HHN)`
- Daten-Eigenschaften - Verknüpfen eine Instanz mit einem Wert
Beispiel: `hatWert(Temperatursensor, 20)`

Web Ontology Language (OWL)

OWL (Web Ontology Language) ist eine vom W3C entwickelte Standardsprache zur Definition und zum Austausch von Ontologien im Web. OWL wird verwendet, um Argumentation, Interoperabilität und maschinelles Verständnis komplexer Daten zu ermöglichen.

Resource Description Framework (RDF)

OWL baut auf RDF auf und verwendet **RDF-Triples** als zugrunde liegendes Datenmodell. OWL ontologies are ultimately expressed as RDF triples:

<Subjekt, Prädikat, Objekt>

Beispiel:

```
:Student rdfs:subClassOf :Person
```

Dies ist ein Beispiel für eine Relation zwischen zwei Klassen, beschrieben als RDF-Triple. Dies ist eine Form der Wissensrepräsentation.

Domäne (Domain)

Eine Domäne Legt fest, welche Art von Ding (Klasse) das Subjekt sein muss. Eine Beispiel:

```
:Hendrik rdfs:studiertAn :HHN
```

Die Klasse oder **Domäne** von der Subjekt Hendrik ist Person.

Wertebereich (Range)

Eine Wertebereich Legt fest, welche Art von Ding (Klasse) das Objekt sein muss.
Eine Beispiel:

```
:Hendrik rdfs:studiertAn :HHN
```

Die Klasse oder **Wertebereich** von der Objekt HHN ist Universität.

OWL Ontologie Beispiel

Klassen:

- Person
- Student (Subklass von Person)
- Lehrer (Subklass von Person)
- Modul

Eigenschaften:

- BelegtModul
- Unterrichtet

OWL Ontologie Beispiel

Wir können diese Ontologie als eine Menge von RDF-Triples beschreiben. Hier sind die Klassen:

- `:Person rdf:type owl:Class`
- `:Student rdfs:subClassOf :Person`
- `:Lehrer rdfs:subClassOf :Person`
- `:Modul rdf:type owl:Class`

OWL Ontologie Beispiel

..und die Eigenschaften:

- `:belegtModul rdf:type owl:ObjectProperty ;`
`rdfs:domain :Student`
`rdfs:range :Modul`
- `:unterrichtet rdf:type owl:ObjectProperty ;`
`rdfs:domain :Lehrer`
`rdfs:range :Modul`

OWL Ontologie Beispiel

