

Wissensbasierte KI

Knut Hinkelmann

Einleitung

In seinem Buch «Thinking, Fast and Slow» unterscheidet Daniel Kahneman (2011) zwischen zwei Denkmodi des Menschen: System 1 ist schnell, instinktiv und emotional; System 2 ist langsamer, überlegter und logischer. System 1 arbeitet automatisch und blitzschnell. Es trifft trifft alltägliche Entscheidungen unbewusst. Im Gegensatz dazu ist System 2 bewusst gesteuert und steht für das rationale Denken. Es ist eher langsam und wird aktiv, wenn es um komplexe Entscheidungen und Aufgaben geht. Dabei agieren die beiden Systeme nicht losgelöst voneinander, sondern beeinflussen sich gegenseitig. Rationale Entscheidungen des Systems 2 beeinflussen, wie unser System 1 die Welt wahrnimmt, und umgekehrt.

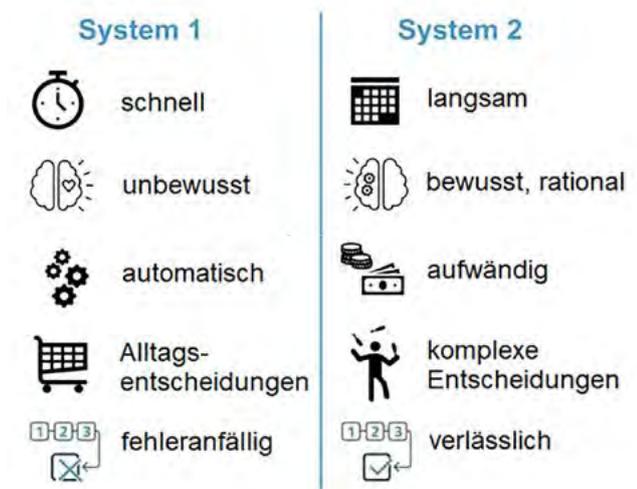


Abbildung 1: System 1 und System 2 nach (Kahneman, 2011)

In der künstlichen Intelligenz kann man zwei analoge Strömungen unterscheiden: die **wissensbasierte KI** und die **datenbasierte KI**. Dabei entspricht die datenbasierte KI dem System 1 und die wissensbasierte KI dem System 2.

In der datenbasierten KI werden durch maschinelles Lernen und Datenanalyse Zusammenhänge zwischen Daten automatisch erkannt und Modelle gebildet, die auf ähnliche Situationen angewendet werden. Intelligenz ist hierbei eine Frage von Daten. Je mehr Daten ein System zur Verfügung hat, desto intelligenter kann es sein.

In der wissensbasierten KI wird Wissen in Form von Regeln oder Zusammenhängen bereitgestellt und für logische Schlussfolgerungen verwendet. Hierbei ist Intelligenz eine Frage von Wissen. Je mehr Wissen ein System hat, desto intelligenter ist es.

Die wissensbasierte KI prägte lange Zeit die Geschichte der KI. Sie hatte einen Höhepunkt in den 1980er und 1990er Jahren mit dem Einsatz von Expertensystemen (Puppe, 2013). Erst mit der Verfügbarkeit grosser Datenmengen und dem rasanten Anstieg der Rechenleistung wurde die datenbasierte KI kommerziell nutzbar. Ähnlich wie System 1 beim Menschen ist jedoch die datenbasierte KI allein nicht in der Lage, alle Arten von Problemen zu lösen. Für rationale, zuverlässige und erklärbare Entscheidungen ist die wissensbasierte KI unerlässlich.

Die wissensbasierte KI automatisiert das symbolische Denken. Dabei wird Wissen durch Symbole repräsentiert. Inferenzregeln manipulieren die Symbole und können Schlüsse ziehen und neues Wissen herleiten. Die Fähigkeit zum symbolischen Denken ist ein Merkmal menschlicher Intelligenz. In der Entwicklung der Menschheit ist es noch relativ jung. Erste Hinweise auf symbolisches Denken gab es vor ungefähr 76 000 Jahren. Symbolisches Denken bezieht sich auf die Verwendung von Symbolen (z. B. Wörtern und Bildern) und mentalen Repräsentationen von Objekten oder Ereignissen zur Darstellung der Welt (Hockenbury & Hockenbury, 2002; Rathus, 2007).

Wenn Computer im Prinzip die gleichen Symbolsysteme verarbeiten können wie Menschen, können sie auch das symbolische Denken automatisieren. Das steckt hinter der von Allen Newell und Herbert Simon formulierten Physical Symbol Systems Hypothesis: «Ein physisches Symbolsystem verfügt über die notwendigen und ausreichenden Mittel für allgemeines intelligentes Handeln» (Newell & Simon, 1976).

Wissensrepräsentation und -verarbeitung

Wissen kann auf verschiedenen Ebenen formuliert, kommuniziert und verarbeitet werden:

- **Kognitiv/linguistisch:** Menschen drücken Wissen hauptsächlich in natürlicher Sprache aus. Allerdings ist die natürliche Sprache zur Repräsentation von Wissen auf dem Computer nicht geeignet, da ein und derselbe Sachverhalt unterschiedlich ausgedrückt werden kann.
- **Konzeptionell/logisch:** Die am besten untersuchte Form der formalen Wissensrepräsentation ist die Logik. In der Logik gibt es Inferenzregeln, um aus einer Menge von Formeln neues Wissen herzuleiten.
- **Ausführbar:** Um dem Computer Wissen zugänglich zu machen, ist eine Repräsentation in einer Form nötig, die der Computer «versteht». Mit «Verstehen» ist gemeint, dass der Computer das Wissen in geeigneter Weise verarbeiten kann. Viele Formalismen basieren dabei auf einer Variante der Logik und einem dazu passenden Verfahren zum automatischen Schliessen.

Für die automatische Wissensverarbeitung benötigt man also

- eine formale Sprache zur Wissensrepräsentation,
- einen Mechanismus zum automatischen Schliessen.

Diese Trennung der Wissensrepräsentation von der Wissensverarbeitung ist ein Grundprinzip der wissensbasierten KI. Dies spiegelt sich wider in der Architektur wissensbasierter Systeme (Abbildung 2). Ein wissensbasiertes System besteht aus einer Wissensbasis, die das explizit repräsentierte Wissen enthält, und einer Schlussfolgerungskomponente, die mit Hilfe von Inferenzregeln neues Wissen herleiten kann, um ein Problem zu lösen oder Entscheidungen zu treffen.

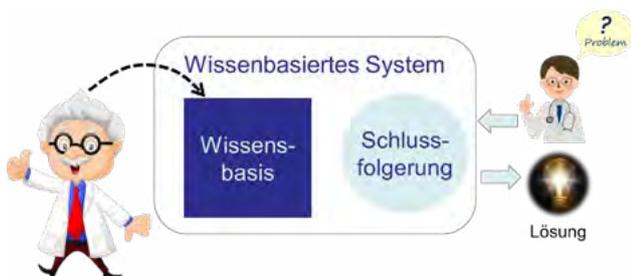


Abbildung 2: Aufbau eines wissensbasierten Systems

Wichtige Arten des Wissens, mit denen Menschen im Alltag häufig umgehen und die auch automatisiert verarbeitet werden können, sind

- relationales Wissen,
- regelbasiertes, logisches Wissen,
- prozedurales Wissen.

Relationales Wissen – Wissensgraphen

Relationales Wissen beschreibt Beziehungen zwischen Konzepten. Man kann dieses Wissen als Wissensgraphen darstellen. Ihr Hauptziel ist es, Daten leicht verständlich zu machen und Maschinen in die Lage zu versetzen, neue Beziehungen herzuleiten.

Die meisten Systeme unterscheiden zwischen Klassen (generische Konzepte) und Individuen. In Abbildung 3 sind die Individuen hellblau und die Klassen gelb dargestellt. Individuen stehen für Konzepte der realen Welt. Im Beispiel steht das Symbol Sokrates für den griechischen Philosophen und Athen für die Stadt, in der er gelebt hat. Individuen können Klassen zugeordnet werden. Im Beispiel ist Sokrates eine Instanz der Klasse Mensch, dargestellt durch die Relation type. Sie drückt aus, dass Sokrates ein Element der Menge aller Menschen ist. Mensch ist eine Unterklasse von Lebewesen, d. h. jeder Mensch ist auch ein Lebewesen. In Mengen ausgedrückt ist die Menge aller Menschen eine Teilmenge der Menge aller Lebewesen.

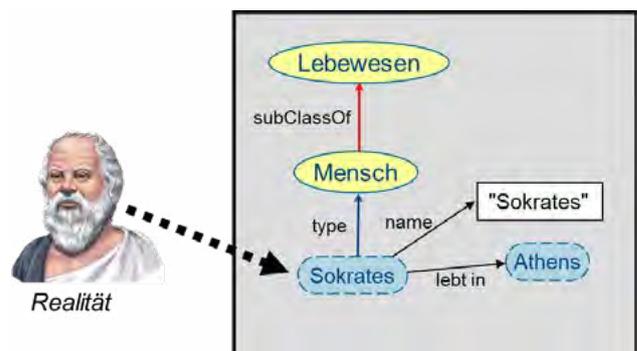


Abbildung 3: Konzepte eines Wissensgraphen

Wissensgraphen lassen sich auch als Tripel darstellen, die man wie Subjekt, Prädikat und Objekt eines Satzes lesen kann. Jedes Tripel repräsentiert eine Beziehung zwischen zwei Knoten eines Graphen. Der Graph in Abbildung 3 hat vier Beziehungen, wird also durch vier Tripel dargestellt.

Sokrates	lebt in	Athen
Sokrates	type	Mensch
Mensch	subClassOf	Lebewesen
Sokrates	name	«Sokrates»

Regelbasiertes Wissen

Regeln sind eine Form der Wissensdarstellung, die insbesondere bei der Entwicklung von Expertensystemen eingesetzt wird. Diese Darstellungsform hat sich als vorteilhaft herausgestellt, da sie auf eine für Menschen natürliche Weise Zusammenhänge oder auszuführende Aktionen beschreibt. Regeln bestehen aus einem Bedingungsteil und einer Schlussfolgerung oder einer Aktion, je nachdem, ob die Regel eine logische Abhängigkeit oder prozedurales Wissen darstellt.

WENN Bedingungen DANN Schlussfolgerung
WENN Bedingungen DANN Aktion

Der Bedingungsteil kann dabei aus mehreren Teilbedingungen bestehen. Verbreitet sind Schlussfolgerungsregeln für Entscheidungen, bei denen rechtliche oder formale Kriterien eingehalten werden müssen, z. B. die Annahme eines Versicherungsantrag oder eines Kreditantrags in einer Bank, die Zulassung von Studierenden zum Studium oder die Vergabe von Beihilfen.

Vereinfachtes Beispiel für eine Hypothek

Regeln: WENN Sicherheit vorhanden UND Eigenkapital
 ausreichend
 DANN Hypothek vergeben

WENN Eigenkapital grösser als 20 % und
 Eigenkapital aus Vorsorge kleiner als 10 %
 DANN Eigenkapital ausreichend

Fakten: Sicherheit vorhanden
 Eigenkapital 25 %
 Eigenkapital aus Vorsorge 8 %

Die Verarbeitung in einem regelbasierten System kann auf unterschiedliche Art und Weise erfolgen.

- Die Vorwärtsverkettung verarbeitet Regeln vorwärts von den Bedingungen zu den Resultaten. Sind die Bedingungen einer Regel erfüllt, werden Schlussfolgerungen gezogen oder Aktionen ausgeführt. Dies wird wiederholt, bis ein gewünschtes Ergebnis oder ein gewünschter Zielzustand erreicht ist. Im Beispiel kann aus den Fakten «Eigenkapital 25 %» und «Eigenkapital aus Vorsorge 8 %» mit der zweiten Regel geschlossen werden, dass das «Eigenkapital ausreichend» ist. Zusammen mit dem Fakt «Sicherheit vorhanden» kann dann mit der ersten Regel «Hypothek vergeben» abgeleitet werden
- Die Rückwärtsverkettung wird verwendet, um eine Hypothese zu prüfen oder eine Anfrage zu beweisen. Dabei werden die Regeln angewendet, bei denen die Schlussfolgerung mit der Anfrage übereinstimmt. Anschliessend wird die Bedingung der Regel als neue Anfrage genommen. Die Verarbeitung stoppt, wenn eine Anfrage mit einem Fakt übereinstimmt. Im Beispiel würde die Anfrage lauten, ob die «Hypothek vergeben» werden kann. Dazu muss überprüft werden, ob «Sicherheit vorhanden» ist und ob das «Eigenkapital ausreichend» ist. Die erste Bedingung ist durch den entsprechenden Fakt erfüllt. Die Bedingung «Eigenkapital ausreichend» kann mit der zweiten Regel bewiesen werden, da deren Bedingungen mit den Fakten «Eigenkapital 25 %» und «Finanzierung aus Vorsorge 8 %» erfüllt sind.

Im Folgenden werden verschiedene Formalismen zur Darstellung und Verarbeitung von Wissen anhand eines konkreten Beispiels vorgestellt. Das Beispiel bezieht sich auf die Zulassung zum Master-Programm Business Information Systems (BIS) an der Fachhochschule Nordwestschweiz FHNW. Es repräsentiert eine typische Anwendungskategorie von künstlicher Intelligenz: die Überprüfung von Anspruchsvoraussetzungen. Ähnliche Anwendungsfälle finden sich etwa bei der Bearbeitung von Kredit- oder Versicherungsanträgen sowie bei der Prüfung von Förderanträgen für Projekte.

Die Entscheidung über die Zulassung zum Studium wird auf Basis eines Antrags und Zusatzdokumenten getroffen. Ein vereinfachtes Kandidatenprofil ist in Abbildung 4 dargestellt.

MSc Business Information Systems Kandidatenprofil

Name: _____

Bachelor in: Wirtschaftsinformatik
 Betriebsökonomie
 Informatik
 anderer
 keiner

Note: _____

Universität: _____

Monate Praxiserfahrung: _____

Abbildung 4: Eingabedaten für den Aufnahmeentscheid

Die Aufnahmekriterien sind in Gesetzen und Studienreglementen geregelt. Etwas vereinfacht gelten die folgenden Kriterien:

Eine Kandidatin oder ein Kandidat wird zum Studium zugelassen, wenn sie/er einen Bachelor-Abschluss in Wirtschaftsinformatik (WI), Betriebsökonomie (BÖK) oder Informatik (Inf) hat. Zudem muss die Note mindestens 4.8 sein und die Person muss über mindestens 12 Monate Praxiserfahrung verfügen.

Diese Entscheidungskriterien werden im Folgenden in mehreren Formalismen dargestellt.

Entscheidungstabellen

Entscheidungstabellen sind eine verbreitete Form, Entscheidungsregeln zu definieren. Die Decision Model and Notation DMN ist ein Standard der Object Management Group, der es erlaubt, Entscheidungslogik mit Geschäftsprozessmodellen zu verknüpfen (OMG, 2024).

DMN besteht aus zwei Teilen: dem Entscheidungs-Anforderungsdiagramm (Decision Requirements Diagram) und der Entscheidungslogik. Im Anforderungsdiagramm in Abbildung 5 sieht man, dass die Zulassung auf dem Studienreglement beruht und dass sie drei Eingabedaten hat: den Bachelor-Abschluss, die Note und die Praxiserfahrung.

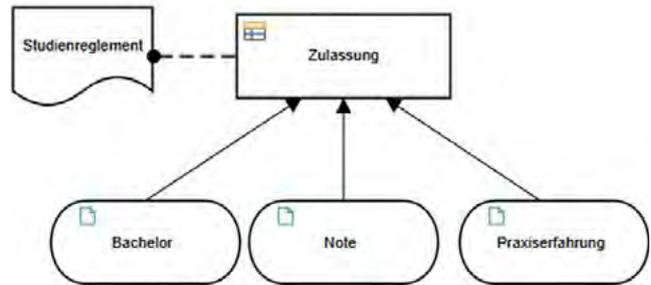


Abbildung 5: Anforderungsdiagramm für die Zulassung zu Master-Studium

Abbildung 6 zeigt die zugehörige Entscheidungstabelle. Jede Zeile entspricht einer Regel, die Spalten sind Eigenschaften, auf deren Basis die Entscheidung getroffen wird.

	inputs			outputs
	Bachelor	Note	Praxiserfahrung	Zulassung
A	Text	Number	# Number	Text "ja", "nein"
1	"Wirtschaftsinformatik"	>=4.8	>=12	"ja"
2	"Betriebsökonomie"	>=4.8	>=12	"ja"
3	"Informatik"	>=4.8	>=12	"ja"
4	"anderer"	-	-	"nein"
5	"keiner"	-	-	"nein"
6	-	<4.8	-	"nein"
7	-	-	<12	"nein"

Abbildung 6: Entscheidungstabelle für den Zulassungsentscheid

Die Modellierung der Entscheidung für die Studiengulassung ist sehr vereinfacht. DMN erlaubt weit komplexere Entscheidungen mit Teilentscheidungen, die in übergeordnete Entscheidungen einfließen, die Berechnung von Werten oder die Verwendung und Bearbeitung komplexer Datentypen.

Für die Auswertung von Entscheidungsregeln gibt es verschiedene Strategien. In der Tabelle von Abbildung 6 gibt es Situationen, in denen mehrere Regeln anwendbar sind. Wenn die Note geringer ist als 4.8 und die Praxiserfahrung kleiner

als 12, sind Regeln 6 und 7 anwendbar, beide liefern aber das gleiche Ergebnis. Falls es Konflikte zwischen Regeln gibt, muss man andere Strategien wählen, z. B. indem die Reihenfolge der Regeln berücksichtigt wird.

Regelbasierte Systeme

Regelbasierte Systeme ermöglichen gegenüber Entscheidungstabellen eine adäquatere Darstellung des Wissens. In der Entscheidungstabelle von Abbildung 6 ist die Note eine Eigenschaft des Bachelor-Abschlusses, während Praxiserfahrung eine Eigenschaft der Person ist. Diese Zusammenhänge gehen in der Entscheidungstabelle verloren.

In logikbasierten Regeln kann man diese Zusammenhänge explizit machen. In der folgenden Wissensbasis werden die Informationen über die Kandidat:innen als Fakten mit den Prädikaten `kandidatIn`, `bachelor`, `note` und `praxis` dargestellt. Hier ein Beispiel für den Kandidaten Peter Meyer, der einen Bachelor in Wirtschaftsinformatik (repräsentiert als `wi`) mit der Note 4.9 hat und über 24 Monate Praxiserfahrung verfügt:

```
kandidat(peterMeyer).
bachelor(peterMeyer).
note(peterMeyer,wi,4.9)
praxis(peterMeyer,24).
```

Die Entscheidungskriterien aus der Entscheidungstabelle in Abbildung 4 lassen sich als Regeln in Prolog wie folgt darstellen:

```
zulassung(K) :- kandidatIn(K),
                bachelor(K, wi),
                note(K,wi,N), N >= 4.8,
                praxis(K,P), P>=12.

zulassung(K) :- kandidatIn(K),
                bachelor(K, boek),
                note(K,boek,N), N >= 4.8,
                praxis(K,P), P>=12.

zulassung(K) :- kandidatIn(K),
                bachelor(K, inf),
                note(K,inf,N), N >= 4.8,
                praxis(K,P), P>=12.
```

Hier genügen drei Regeln, denn es muss nur dargestellt werden, unter welchen Bedingungen eine Zulassung erfolgt. Für alle Situationen, in denen keine Regel anwendbar ist, ist die Zulassung automatisch nicht möglich.

Bedingung und Schlussfolgerung einer Regel sind durch das Zeichen `:-` getrennt. In Prolog ist die Schlussfolgerung auf der linken Seite und die Bedingung rechts. Die erste Regel kann man lesen als: «Wenn ein Kandidat einen Bachelor in Wirtschaftsinformatik hat mit einer Note, die grösser oder gleich 4.8 ist, und die Praxiserfahrung des Kandidaten in Monaten grösser als 12 ist, dann bekommt der Kandidat eine Zulassung.»

Die Auswertung der Regeln erfolgt in Prolog durch Rückwärtsverkettung. Die Auswertung startet mit einer Anfrage, eingeleitet durch in Fragezeichen. Die Anfrage

```
?- zulassung(peterMeyer).
```

wertet aus, ob die durch `peterMeyer` repräsentierte Person eine Zulassung bekommt. Die Anfrage wird mit den Fakten zu `peterMeyer` (siehe oben) erfolgreich beantwortet.

Die Anfrage

```
?- zulassung(X).
```

ergibt als Antwort alle Kandidat:innen, die zugelassen werden können. Mit den obigen Fakten wäre `X = peterMeyer` eine Antwort.

Dieses Beispiel in Prolog dient dazu, das Prinzip regelbasierter Systeme zu erläutern. Es gibt eine ganze Reihe anderer Systeme, die neben der Rückwärtsverkettung auch Vorwärtsverkettung von Regeln erlauben. Dabei werden die Regeln auf die Fakten angewendet, um alle möglichen Konsequenzen herzuleiten. WIN-PROLOG von LPA¹ kombiniert Prolog mit Vorwärtsverkettung in einem System.

¹ www.lpa.co.uk

Wissensgraphen:

Strukturierung eines Anwendungsgebiets

Wie man am Beispiel von Prolog sieht, sind in Regelsystemen die Daten als Menge von Fakten repräsentiert. Wissensgraphen dienen dazu, einen Anwendungsbereich zu strukturieren, indem man die wesentlichen Konzepte identifiziert und in Beziehung setzt. Durch Zuordnung von Daten zu Konzepten kann man Daten eine Bedeutung geben. Daher werden die Technologien auch als **Ontologien** oder allgemein als **semantische Technologien** bezeichnet.

Im Folgenden wird das Vorgehen zur Entwicklung eines Wissensgraphen erläutert. Das Vorgehen orientiert sich der wohl bekanntesten Methode zu Entwicklung einer Ontologie (Noy & McGuinness, 2001), wurde aber in Bezug auf die Praktikabilität vereinfacht.

Das Vorgehen besteht aus fünf Schritten:

1. Bereich und Umfang der Ontologie bestimmen
2. Wichtige Begriffe identifizieren
3. Konzepthierarchie festlegen
4. Eigenschaften definieren
5. Individuen erfassen

Bereich und Umfang der Ontologie festlegen

In einem ersten Schritt wird festgelegt, welche Themenbereiche das KI-System umfassen soll. Dazu hat sich bewährt, sich über die Fragen zu einigen, die das System beantworten soll. Die Fragen werden Kompetenzfragen (competency questions) genannt (siehe Grüninger & Fox, 1994, 1995), da sie bestimmen, über welche Kompetenz das Systems verfügen soll.

Im Beispiel könnten die folgenden Kompetenzfragen gelten:

- Berechtigt die Vorbildung zum gewählten Studiengang an einer Wirtschaftshochschule?
- Ist die Praxiserfahrung ausreichend?
- Wer kann zu einem Masterstudiengang zugelassen werden?

Wichtige Begriffe identifizieren

Im nächsten Schritt geht es darum, die für die Wissensbasis relevanten Begriffe zu identifizieren. Diese leiten sich aus den Kompetenzfragen ab. Im Beispiel würde etwa folgende Überlegungen zu relevanten Begriffen führen:

Die Vorbildung hängt vom gewünschten Studienabschluss ab, wobei man Bachelor und Master unterscheidet. Die Fachhochschule Nordwestschweiz FHNW bietet Bachelor-Studiengänge in Betriebsökonomie, Wirtschaftsinformatik und Informatik und einen Masterstudiengang in BIS an. Für Bachelor-Studiengänge benötigt man eine Matura. Die passende Matura hängt vom Studiengang ab und der Hochschule ab. Es gibt sowohl technische als auch kaufmännische Berufsmatura.

Daraus ergibt sich folgende Liste von Begriffen: Studienabschluss, Bachelor, Master, Hochschule, Fachhochschule, Universität, Matura, Berufsmatura, Wirtschaftsinformatik, Informatik, Betriebsökonomie, FHNW.

Konzepte und Konzepthierarchie festlegen

Die in der vorherigen Phase identifizierten Begriffe werden nun strukturiert. Ein wichtiger Schritt ist dabei die Unterscheidung zwischen zwei verschiedenen Arten von Konzepten:

- Generische Konzepte
- Individuelle Konzepte

In diesem Schritt geht es darum, die generischen Konzepte zu erkennen und von den individuellen Konzepten zu unterscheiden. Individuelle Konzepte entsprechen materiellen und immateriellen Elementen der realen Welt. Generische Konzepte entsprechen Klassen oder Kategorien. Man kann sie sich als Mengen von Individuen vorstellen. Generische Konzepte kann man in Klassenhierarchien strukturieren.

Im Beispiel sind Hochschule, Universität und Fachhochschule generische Konzepte, wobei Hochschule einen Oberbegriff für Universität und Fachhochschule darstellt. Dies ist ein Beispiel für eine Klassenhierarchie. Studienabschluss, Bachelor und Master sind ebenfalls generische Konzepte. Wirtschaftsinformatik, Betriebsökonomie und Informatik sind Studiengänge an der FHNW. In Abbildung 7 sind die Konzepte graphisch dargestellt. Die gelben Konzepte sind Klassen, die blauen sind Individuen. Die roten Pfeile stellen Subklassen-Beziehungen dar, während die blauen Pfeile die Individuen ihren Klassen zuordnen.

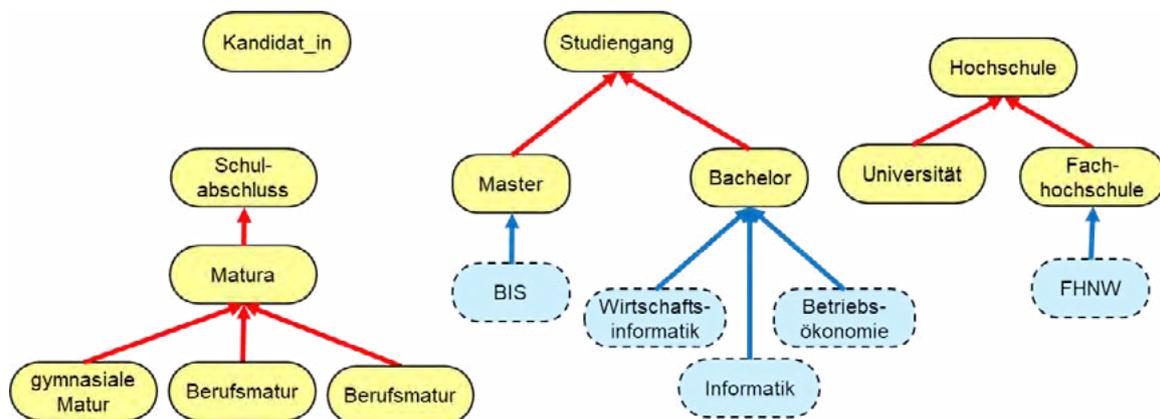
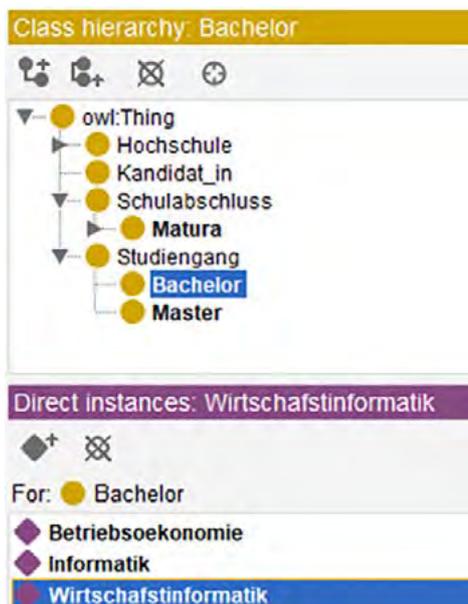


Abbildung 7: Abbildung Klassen und Individuen

Für die Definition von Wissensgraphen gibt es mächtige Werkzeuge. Abbildung 8 zeigt Benutzeroberfläche von Protege mit dem in Abbildung 7 definierten Wissensgraphen.

Wie oben erwähnt lassen sich Wissensgraphen auch als Tripel darstellen, die man wie Subjekt, Prädikat und Objekt eines Satzes lesen kann. Der in Abbildung 8 dargestellte Graph entspricht folgenden Triples:



Universitaet	subClassOf	Hochschule
Fachhochschule	subClassOf	Hochschule
Matura	subClassOf	Schulabschluss
Berufsmatur	subClassOf	Matura
Fachmatur	subClassOf	Matura
Gymnasiale Matur	subClassOf	Matura
Bachelor	subClassOf	Studiengang
Master	subClassOf	Studiengang
FHNW	type	Fachhochschule
Betriebsoekonomie	type	Bachelor
Informatik	type	Bachelor
Wirtschaftsinformatik	type	Bachelor
BIS	type	Master

Abbildung 8: Wissensgraph in Protege

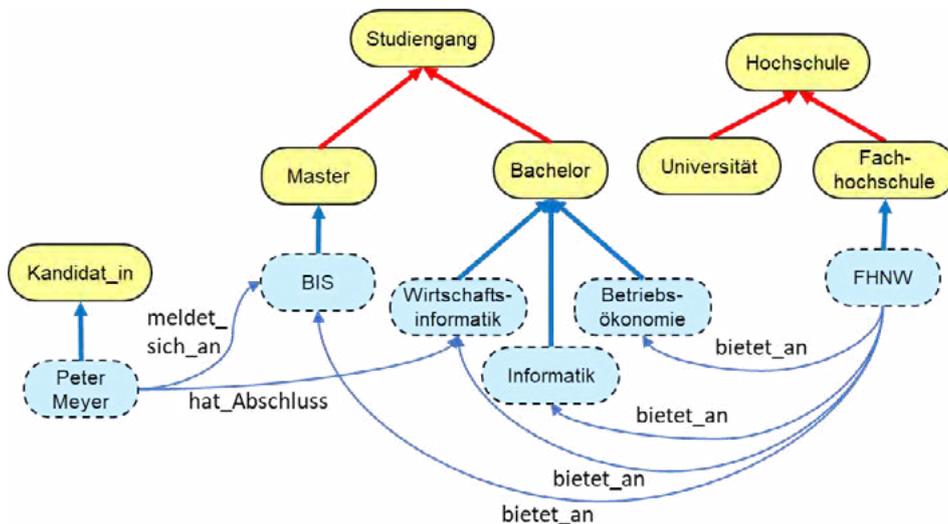


Abbildung 9: Objekt-Beziehungen zwischen Individuen

Beziehungen definieren

Neben den vordefinierten Beziehungen zum Aufbau einer Klassenhierarchie kann man in einem Wissensgraphen beliebige Beziehungen definieren. Man unterscheidet zwischen Datenbeziehungen (Data Properties) und Objektbeziehungen (Object Properties).

Objektbeziehungen definieren Beziehungen zwischen Individuen einer Klasse. So kann man definieren, dass ein Studiengang an einer Hochschule angeboten wird und dass ein Studiengang zu einem Studienabschluss führt. In Abbildung 9 ist der Wissensgraph aus Abbildung 7 um entsprechende Beziehungen für die Individuen ergänzt.

Für jeden Wissensgraph kann definiert werden, welche Beziehungen für die Individuen einer Klasse erlaubt sind. Dies ermöglicht die Repräsentation von Wissen über den Anwendungsbereich. In unserem Beispiel würde man also definieren, dass die Beziehung `bietet_an` nur zwischen Hochschulen und Studiengängen gilt.

Man kann auf diese Art auch Beziehungen zwischen Individuen definieren, die noch nicht bekannt sind. So kann man definieren, dass Kandidat:innen sich für einen Studiengang anmelden und einen Schulabschluss besitzen. Abbildung 10 zeigt die Definition der im Wissensgraphen von Abbildung 9 verwendeten Beziehungen.



Abbildung 10: Spezifikation von Beziehungen zwischen Klassen in Protege

Individuen zuordnen

In realen Anwendungen enthält der Wissensgraph einige Individuen, die für die Anwendung wichtig sind. In unserem Beispiel sind dies die Studienabschlüsse, die angebotenen Studiengänge und die Hochschule. Andere Individuen kennt man erst zur Zeit der Anwendung. In unserem Beispiel wären dies die Kandidat:innen, die sich für einen Studiengang anmelden.

Durch die Zuordnung der Daten zu den Klassen eines Wissensgraphen bekommen die Daten eine Semantik. Wenn die Daten über Kandidat:innen dem Wissensgraphen zugeordnet werden, dann kann man Inferenzen über ihren Abschluss ziehen. So kann man prüfen, ob der Schulabschluss einer Matura entspricht, oder man kann filtern, welche Kandidat:innen sich für Studiengänge angemeldet haben, die zu einem Master-Abschluss führen.

Kombination von Regeln und Wissensgraphen

Zwar sind Wissensgraphen sehr gut geeignet, die Semantik eines Anwendungsbereichs adäquat zu repräsentieren, sie sind aber für die Repräsentation von Entscheidungen weniger gut geeignet. Die Kombination von Wissensgraphen mit Regeln erlaubt es, zusätzlich zur Semantik eines Anwendungsgebiets neues Wissen herzuleiten. Die folgende Regel stellt dar, dass ein:e Kandidat:in mit einem Bachelor-Abschluss in Wirtschaftsinformatik mit Note besser als 4.8 und mehr als 2 Jahre Praxiserfahrung zum Studiengang BIS zugelassen wird:

```
Kandidat_in(?K) ^
hat_Abschluss(?K, Wirtschaftsinformatik) ^
Bachelor(Wirtschaftsinformatik) ^
note(?A, ?N) ^ swrlb:greaterThan(?N,4.8) ^
praxis(?K,?P) ^ swrlb:greaterThan(?P,12)
-> zulassung(?X, BIS)
```

In der Semantic Web Rule Language SWRL werden Bedingungen und Konklusion durch das Symbol -> getrennt, das den Pfeil für die Implikation darstellt. Die Bedingungen sind links und die Schlussfolgerung rechts vom Pfeil. Die Bedingungen und die Schlussfolgerung sind dargestellt durch Prädikate für die Beziehungen und Klassen des Wissensgraphen. Die zweistelligen Prädikate wie `hat_Abschluss` entsprechen Beziehungen, einstellige Prädikate wie `Kandidat_in` oder `Bachelor` sind Klassen. Die Argumente von Prädikaten sind Individuen, z. B. `Wirtschaftsinformatik` und `BIS`.

Vages Wissen – Fuzzy-Logik

Die bisher behandelten Formalismen zu Wissensrepräsentation behandelten sicheres Wissen. Aussagen sind entweder wahr oder falsch. In der realen Welt hat man es oft auch mit Aussagen zu tun, deren Wahrheitswert nicht so genau zu bestimmen ist. Gute Beispiele dafür findet man in unseren Sinneswahrnehmungen. Begriffe wie salzig, kalt und heiss sind nicht eindeutig zu messen. Auch wenn man vielleicht den Salzgehalt oder die Temperatur messen könnte, so basieren unsere Verhaltensregeln eher auf sogenannten vagen Aussagen. Die Regel

Wenn es draussen kalt ist, ziehe ich eine Jacke an

ist für uns adäquat. Wir können sie anwenden, ohne die genaue Temperatur in der Regel angeben zu müssen. Zudem ist der Begriff «kalt» nicht klar abgegrenzt. Es gibt keine klare Grenze zwischen kalt, normal oder warm.

In Unternehmensanwendungen finden wir ebenfalls Beispiele für vage Begriffe und Regeln. So kann man einen Kredit vergeben, wenn der Antragsteller eine gute Bonität und niedrige Schulden hat.

Auch im Beispiel für die Zulassungskriterien zum Masterstudiengang gibt es vages Wissen. Tatsächlich definiert die Studienordnung nicht genau die Note 4.8 als Zulassungskriterium, sondern sie besagt, dass der Kandidat bzw. die Kandidatin mindestens eine gute Note haben muss. Die Grenze von 4.8 ist eine Interpretation dieser Aussage. Fuzzy-Logik erlaubt es, den Wert «gut» für eine Note nicht als absoluten Wert zu betrachten, sondern als Wert zwischen 0 und 1, wobei 1 für wahr und 0 für falsch steht:

- Eine Note, die kleiner ist als 4.6, ist nicht gut (Zugehörigkeitswert 0)
- Eine Note, die grösser ist als 5, ist definitiv gut (Zugehörigkeitswert 1)
- Eine Note zwischen 4.6 und 5 bekommt einen Zugehörigkeitswert zwischen 0 und 1.

Diese Zugehörigkeitsfunktion kann man auch graphisch darstellen (siehe Abbildung 11).

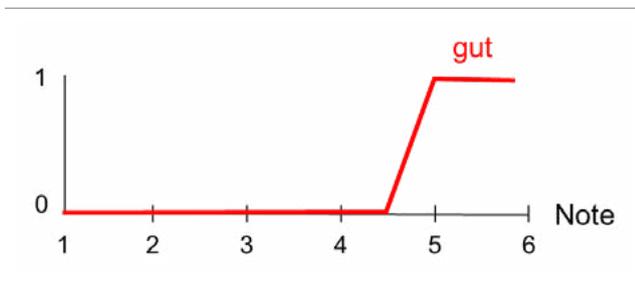


Abbildung 11: Zugehörigkeitsfunktion für gute Note

Analog kann man für eine Eigenschaft auch mehrere Fuzzy-Werte definieren. Abbildung 12 zeigt drei Zugehörigkeitsfunktionen für kurze, mittlere und lange Berufserfahrung. Jede Farbe entspricht einer Zugehörigkeitsfunktion.

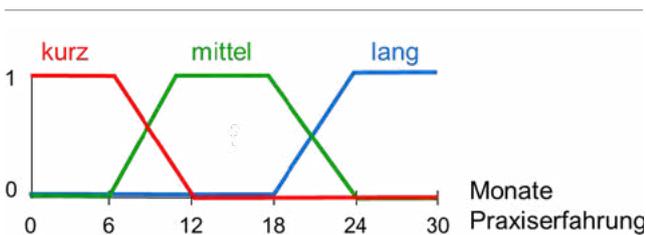


Abbildung 12: Zugehörigkeitsfunktionen für Arbeitserfahrung

Mit diesen Fuzzy-Werten kann man nun einfache Regeln formulieren:

WENN Note = gut UND Praxis = lang
DANN Zulassung = ja

WENN Note = sehr gut UND Praxis = mittel
DANN Zulassung = ja

WENN Note = gut UND Praxis = mittel
DANN Zulassung = ja

WENN Praxis = kurz
DANN Zulassung = nein

Für die Anwendung werden die konkreten Daten für die Noten und die Praxiserfahrung mit den Zugehörigkeitsfunktionen auf ihre Fuzzy-Werte abgebildet und dann die Regeln angewendet.

Fuzzy-Logik hat sich auch bewährt bei der Steuerung von Maschinen, wenn Daten mehrerer Sensoren miteinander verrechnet werden. Bekannte Beispiele sind Waschmaschinen, Heizungsregler und auch Bildstabilisatoren in Kameras.

Fazit

Analog zum Denken beim Menschen gibt es auch in der künstlichen Intelligenz die Unterscheidung von wissensbasierten und datenbasierten Schlussfolgerungen. Durch die wissensbasierte KI werden Erkenntnisse menschlicher Expert:innen einem Computer zugänglich gemacht. Dieses Wissen basiert auf Fachwissen und Erkenntnissen, die sich teilweise über Jahre entwickelt haben. Dies ist vor allem dann von Bedeutung, wenn Schlussfolgerungen nachvollziehbar und Ergebnisse erklärbar sein müssen.

In der wissensbasierten KI wird Wissen explizit in einer Wissensbasis repräsentiert und mit Inferenzverfahren ausgewertet. Dabei unterscheidet man zwischen relationalem und regelbasiertem Wissen. Weit verbreitete Formen der Wissensrepräsentation basieren auf klassischer Logik. Daneben gibt es eine Vielzahl weiterer Wissensrepräsentations- und Inferenzverfahren. Im Text wurde der Umgang mit vagem Wissen erwähnt. Andere Verfahren beschäftigen sich mit nicht-monotonem Schliessen, bei dem abgeleitetes Wissen wieder zurückgenommen werden kann, oder Produktionssystemen, bei denen Aktionen abgeleitet werden.

Literatur

- Grüninger, M., & Fox, M. S. (1994). The Role of Competency Questions in Enterprise Engineering. *Proceedings of the IFIP WG5.7 Workshop on Benchmarking – Theory and Practice*.
- Grüninger, M., & Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies. *Industrial Engineering (1995)*, 95, 1–10.
- Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.
- Newell, A., & Simon, H. A. (1976). Computer Science as Empirical Inquiry: Symbols and Search. *Communications of the ACM*, 19(3), 113–126.
- Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05.
- OMG. (2024). *Decision Model and Notation (DMN)* (Nummer May). Object Management Group OMG. www.omg.org/spec/DMN
- Puppe, F. (2013). *Einführung in Expertensysteme (2. Aufl.)*. Springer. doi.org/10.1007/978-3-642-76621-3