

Einführung in die künstliche Intelligenz

EKI04 – Wissensrepräsentation

Prof. Dr. A. del Pino

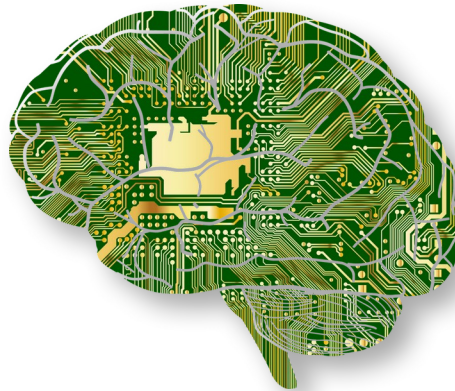


Bild: https://en.wikipedia.org/wiki/Machine_learning#/media/File:Anatomy-1751201_1280.png

Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

Die KI-Landkarte

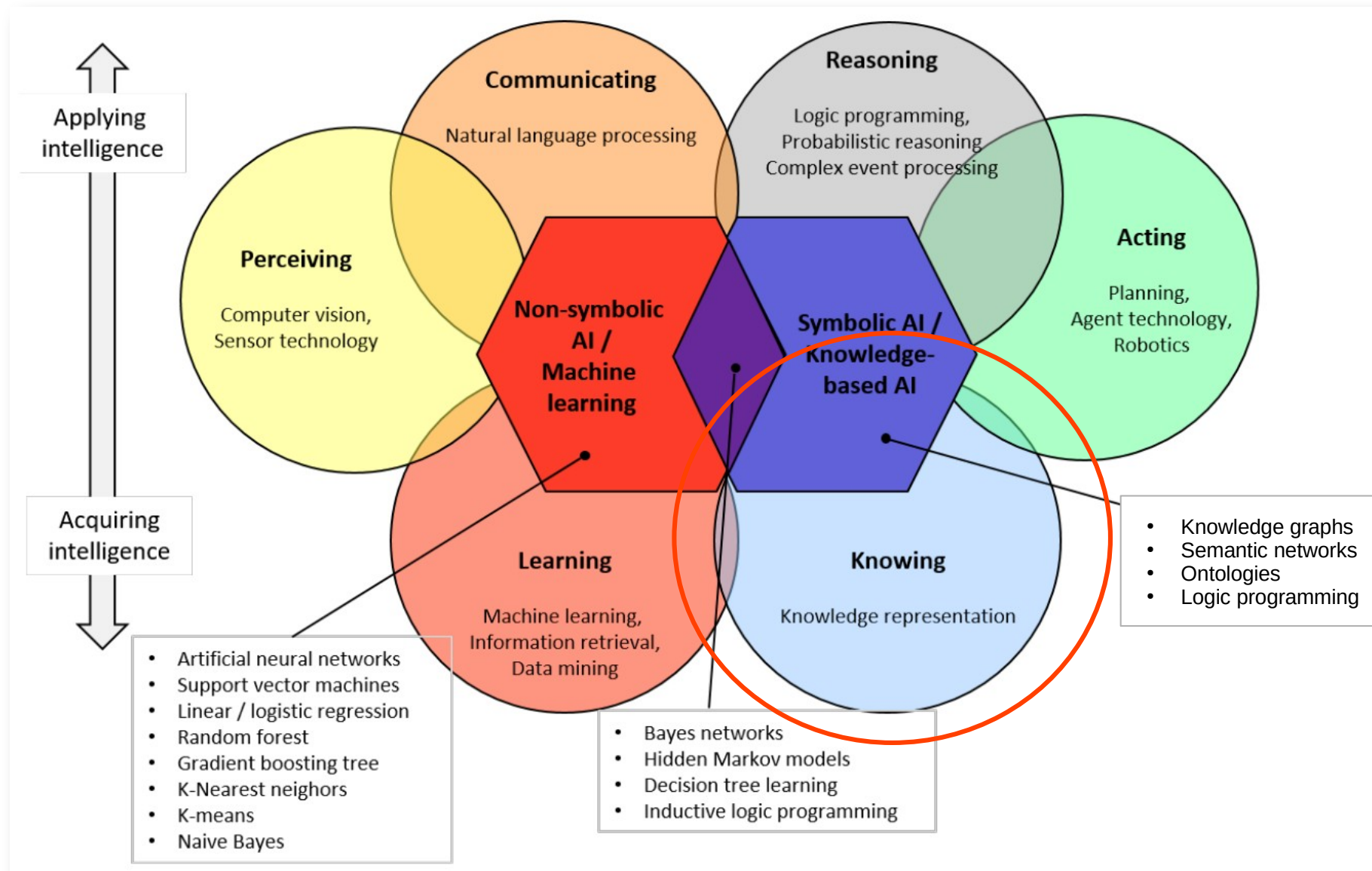


Bild: B. Humm "Applied Artificial Intelligence", S. 4

Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

Was wissen Sie über Filme?

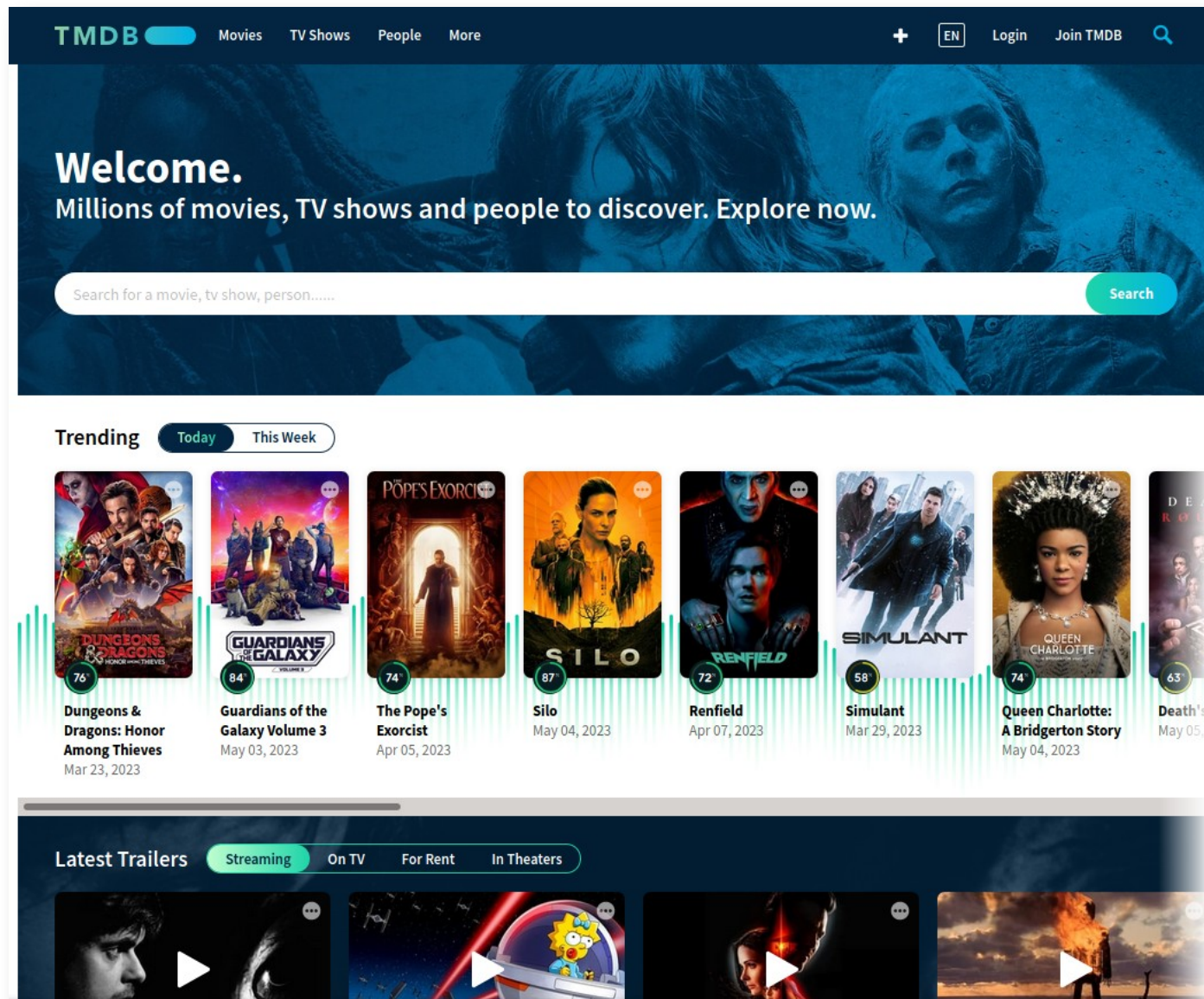


Bild: <https://www.themoviedb.org/>

Wissen über Filme

TMDB Movies TV Shows People More

Overview Media Fandom Share

A Beautiful Mind (2001)
 12 02/28/2002 (DE) • Drama, Romance • 2h 15m

79% User Score

Play Trailer

He saw the world in a way no one could have imagined.

Overview
 John Nash is a brilliant but asocial mathematician fighting schizophrenia. After he accepts secret work in cryptography, his life takes a turn for the nightmarish.

Ron Howard
 Director

Akiva Goldsman
 Writer

Now Streaming
 Watch Now

Top Billed Cast

 Russell Crowe John Nash	 Jennifer Connelly Alicia Nash	 Ed Harris William Parcher	 Paul Bettany Charles Herman	 Christopher Plummer Dr. Rosen	 Josh Lucas Martin Hansen
--	--	--	---	--	---

Status
Released

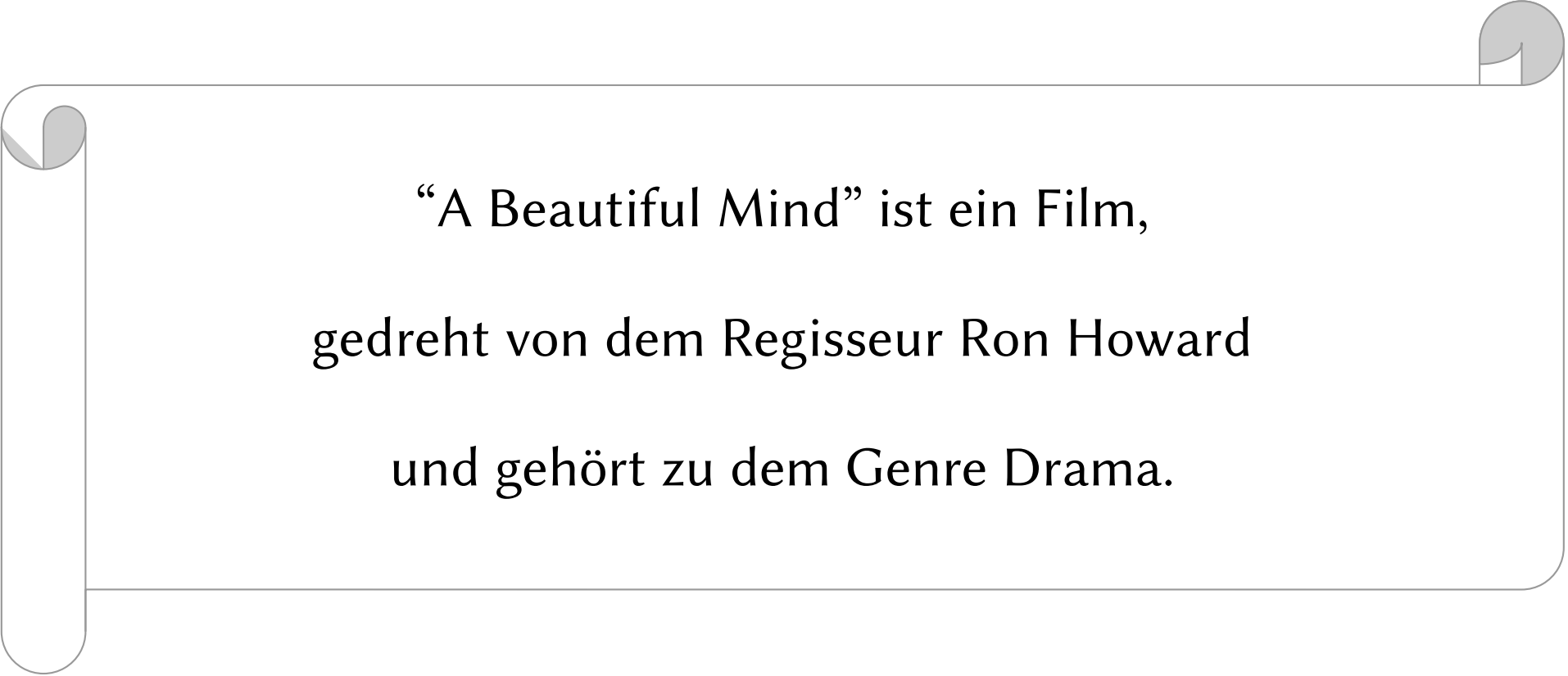
Original Language
English

Budget
\$58,000,000.00

Revenue
\$316,800,000.00

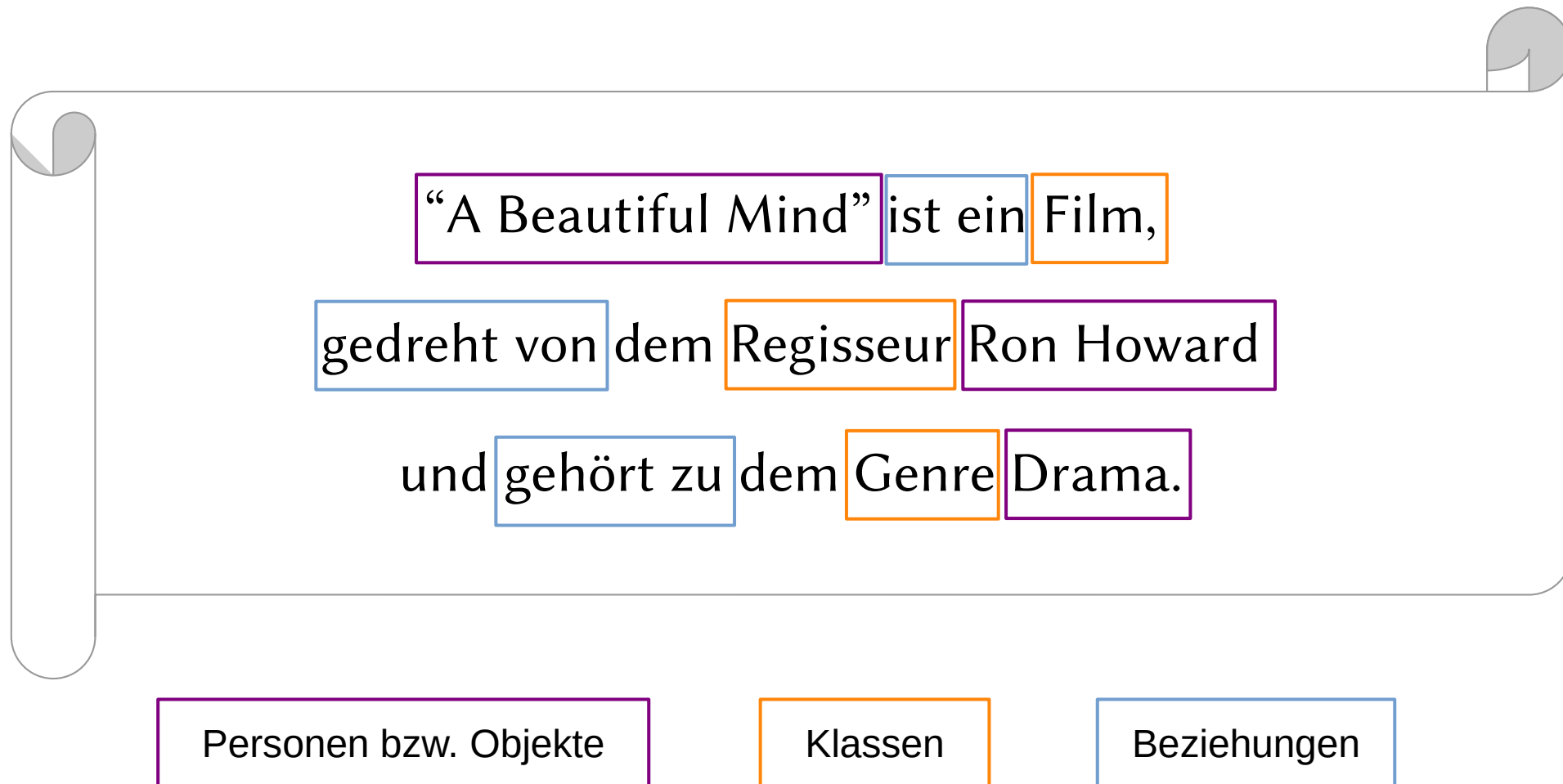
Bild: <https://www.themoviedb.org/movie/453-a-beautiful-mind>

Wissen über Filme

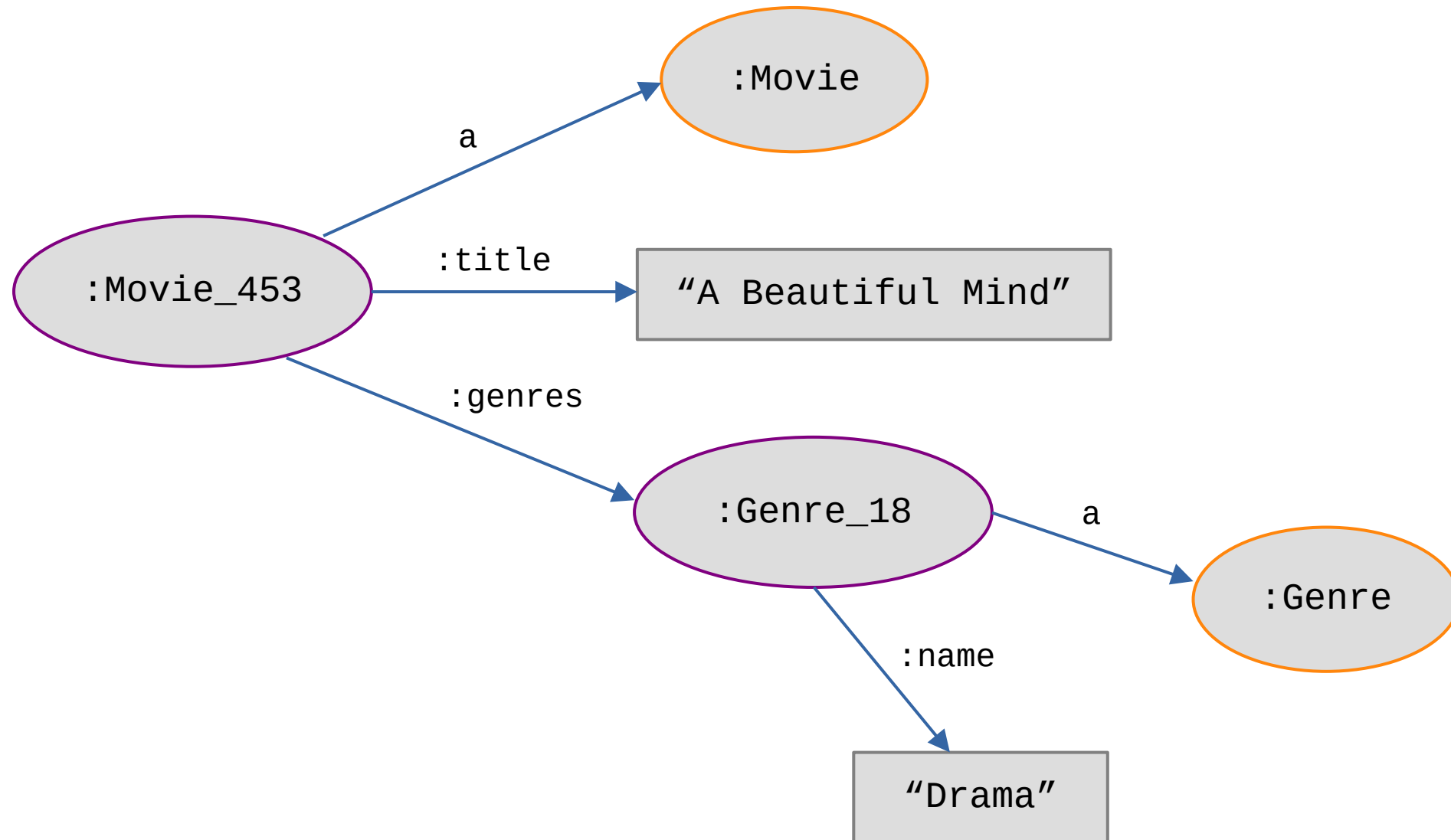


“A Beautiful Mind” ist ein Film,
gedreht von dem Regisseur Ron Howard
und gehört zu dem Genre Drama.

Wissen über Filme repräsentieren



Wissensgraph (knowledge graph)



- Verwandte Begriffe: Semantisches Netzwerk, Linked Data, Ontologie, Thesaurus

Linked data: Tausende von Wissensgraphen

- Große Menge an Wissensgraphen
- Öffentlich verfügbar
- Tausende von Einträgen
- Standardisierte Formate
- Viele Domänen, z.B. Gesundheitswesen, Geographie, Medien, Soziale Netze
- Beispiele: Wikidata, dbpedia, YAGO, MeSH

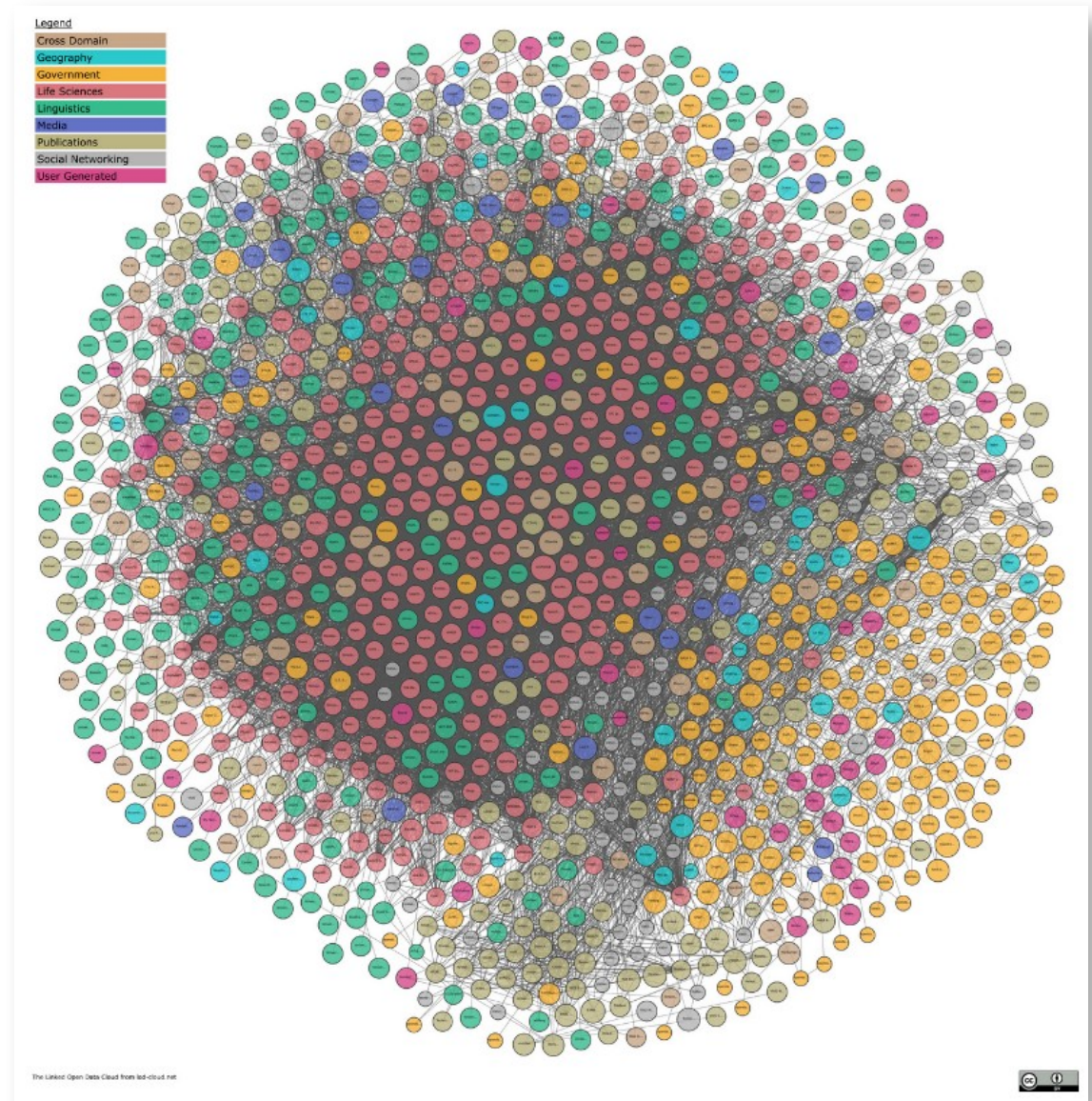


Bild: <https://lod-cloud.net/>

Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

Resource Description Framework (<https://www.w3.org/RDF/>)

- **RDF**: Formale Sprache um Wissensgraphen zu repräsentieren
- Spezifiziert durch das **World Wide Web Consortium (W3C)** als Teil der **Semantic Web** Initiative
<https://www.w3.org/standards/semanticweb>
- Benutzte Notation in diesem Kurs: Turtle <https://www.w3.org/TR/turtle>



Bild: <https://www.w3.org/RDF/>

RDF Ressourcen werden durch URIs repräsentiert

Definition eines **base namespace**
Erlaubt es, die Ressourcen abzukürzen

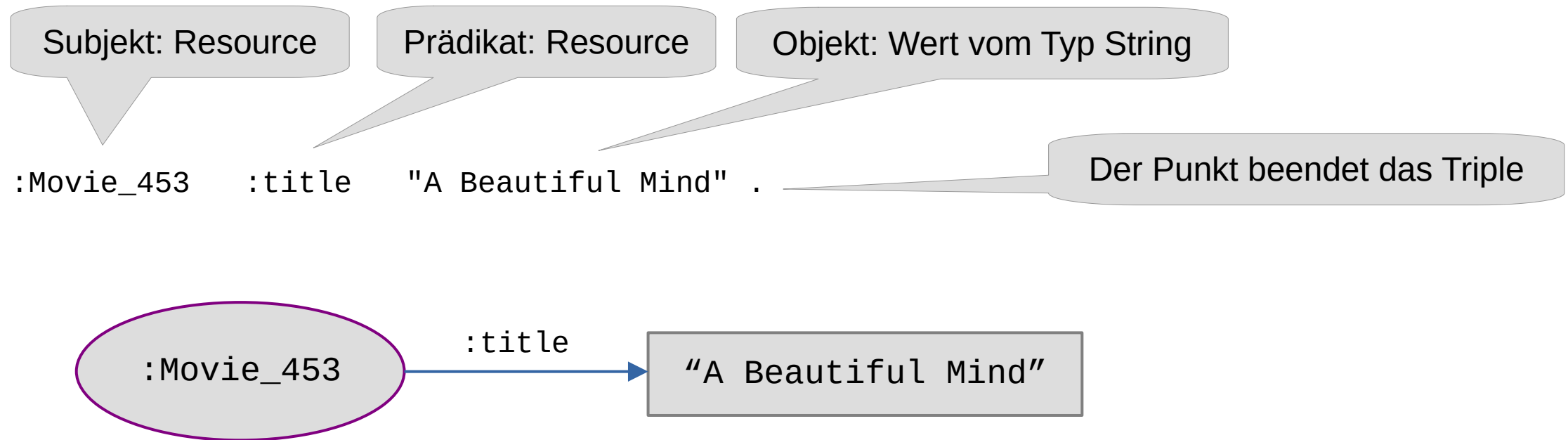
@base <https://www.themoviedb.org/kaggle-export/> .

:Movie_453

Eine **Resource** in RDF. Abkürzung für
<https://www.themoviedb.org/kaggle-export/Movie_453>

Tripel (triples): Die Notation für Aussagen in RDF

- RDF Tripel bestehen aus drei Teilen:
 - Subjekt (subject): Eine Resource
 - Prädikat (predicate): eine weitere Resource
 - Objekt (object): Eine Resource oder ein Wert (string, int, float, date,...)
- Beispiel



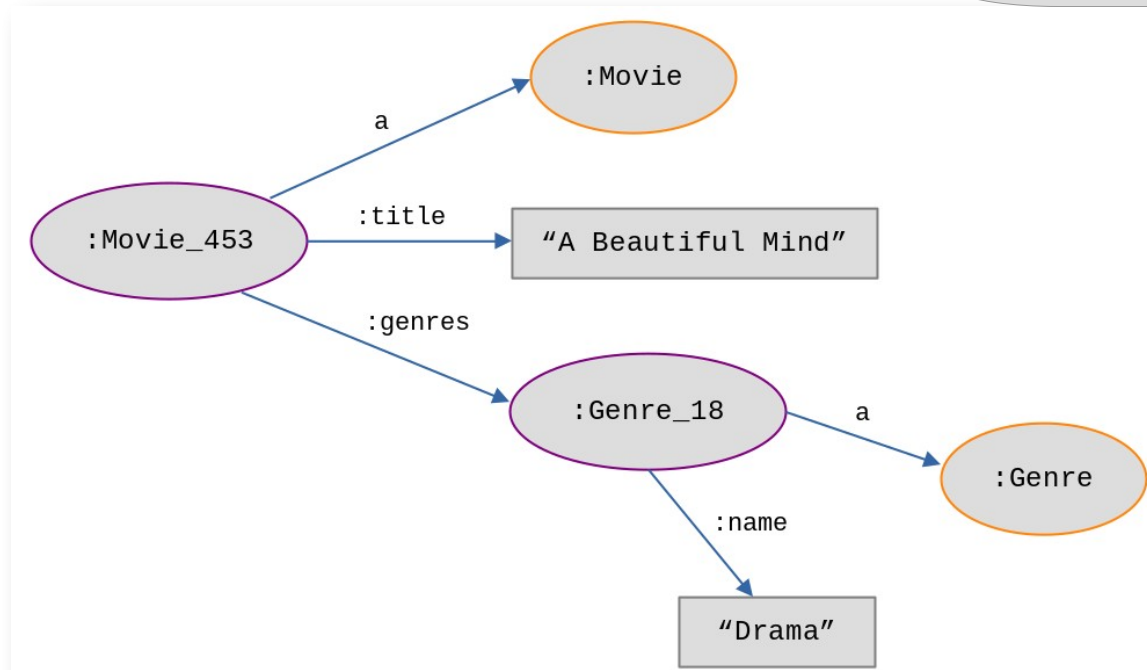
RDF Tripel repäsentieren Wissensgraphen

```
:Movie_453 a :Movie ;  
  :title "A Beautiful Mind" ;  
  :genres :Genre_18 .
```

Mit dem Semikolon kann man mehrere Tripel für das gleiche Subjekt verknüpfen

```
:Genre_18 a :Genre ;  
  :name "Drama" .
```

Das a steht für eine Instanz-Beziehung (Abkürzung für `rdf:type`)



Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

Der Datensatz: Kaggle Auszug der TMDB

<https://www.kaggle.com/c/tmdb-box-office-prediction/data?select=train.csv>

Playground Prediction Competition

TMDB Box Office Prediction

Can you predict a movie's worldwide box office revenue?

Kaggle · 1,395 teams · 4 years ago

Overview **Data** Code Discussion Leaderboard Rules Team Submissions **Late Submission** ...

Dataset Description

In this dataset, you are provided with 7398 movies and a variety of metadata obtained from The Movie Database (TMDB). Movies are labeled with `id`. Data points include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries.

You are predicting the worldwide revenue for 4398 movies in the `test` file.

Note - many movies are remade over the years, therefore it may seem like multiple instance of a movie may appear in the data, however they are different and should be considered separate movies. In addition, some movies may share a title, but be entirely unrelated.

E.g. *The Karate Kid* (`id: 5266`) was released in 1986, while a clearly (or maybe just subjectively) inferior remake (`id: 1987`) was released in 2010.

Also, while the *Frozen* (`id: 5295`) released by Disney in 2013 may be the household name, don't forget about the less-popular *Frozen* (`id: 139`) released three years earlier about skiers who are stranded on a chairlift...

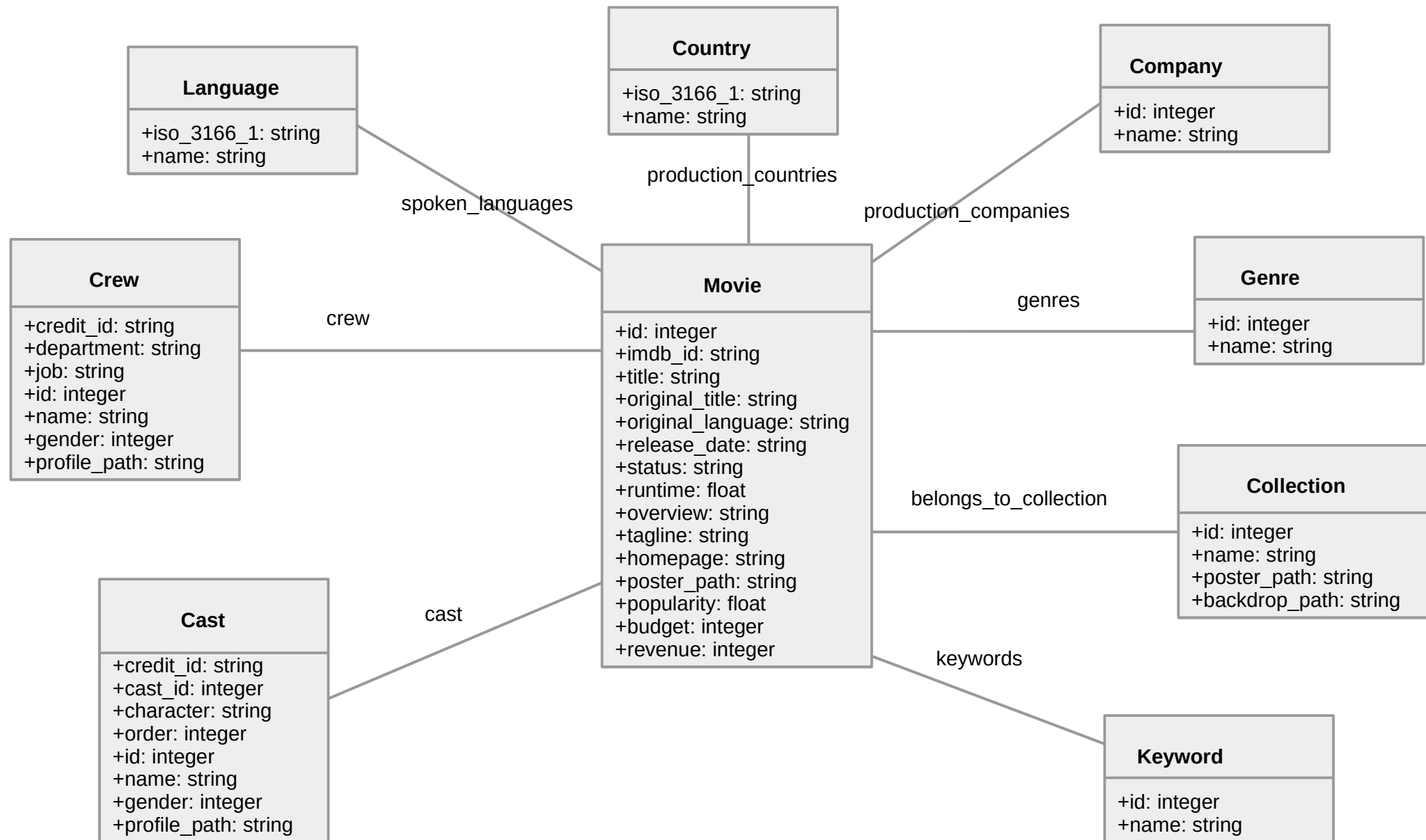
Files
3 files

Size
70.24 MB

Type
csv

Bild: <https://www.kaggle.com/competitions/tmdb-box-office-prediction/data>

Datenmodell (UML Klassendiagramm)



RDF Umwandlung: TMDb.ttl (mit speziellem Code für diesen Kurs)

The image shows a screenshot of a KWrite text editor window titled "TMDb.ttl — KWrite". The editor displays a JSON response from the TMDb API for the movie "Hot Tub Time Machine 2". The JSON is formatted with syntax highlighting and line numbers on the left. The response includes details such as the movie's title, genres, budget, IMDb ID, original language, overview, popularity, poster path, production companies, release date, runtime, spoken languages, status, tagline, keywords, cast, and crew. The status is "Released". The overview mentions that Lou, the 'father of the Internet', is shot by an assailant named Jacob and Nick, who fire up the time machine to save him. The cast list includes actors like James Van Der Beek, David Krumholtz, and James Van Der Beek. The crew list includes the director, James Van Der Beek, and the producer, James Van Der Beek. The status is "Released".

```
1102611 # Movie
1102612
1102613
1102614 :Movie_1 a :Movie ;
1102615     :id "1"^^xsd:integer ;
1102616     :belongs_to_collection :Collection_313576 ;
1102617     :budget "14000000"^^xsd:integer ;
1102618     :genres :Genre_35 ;
1102619     :imdb_id "tt2637294" ;
1102620     :original_language "en" ;
1102621     :original_title "Hot Tub Time Machine 2" ;
1102622     :overview "When Lou, who has become the 'father of the Internet,' is shot by an unknown
1102623     assailant, Jacob and Nick fire up the time machine again to save their friend." ;
1102624     :popularity "6.575392999999999"^^xsd:float ;
1102625     :poster_path "/tQtWuwvMf0hCc2QR2tkolwl7c3c.jpg" ;
1102626     :production_companies :Company_4 , :Company_60 , :Company_8411 ;
1102627     :production_countries :Country_US ;
1102628     :release_date "2/20/15" ;
1102629     :runtime "93.0"^^xsd:float ;
1102630     :spoken_languages :Language_en ;
1102631     :status "Released" ;
1102632     :tagline "The Laws of Space and Time are About to be Violated." ;
1102633     :title "Hot Tub Time Machine 2" ;
1102634     :Keywords :Keyword_4379 , :Keyword_9663 , :Keyword_11830 , :Keyword_179431 ;
1102635     :cast :Cast_52fe4ee7c3a36847f82afae7 , :Cast_52fe4ee7c3a36847f82afae8 ,
1102636     :Cast_52fe4ee7c3a36847f82afaf3 , :Cast_52fe4ee7c3a36847f82afaf7
1102637     , :Cast_52fe4ee7c3a36847f82afafb , :Cast_52fe4ee7c3a36847f82afaff ,
1102638     :Cast_5524ec51c3a3687df3000dbb , :Cast_5524ec8ec3a3687ded000d72 , :Cast_5524ec9bc3a3687df8000d13
1102639     , :Cast_5524eca892514171cb008237 , :Cast_5524ecb7925141720c001116 ,
1102640     :Cast_5524ecc3c3a3687ded000d74 , :Cast_5524ecd3c3a3687e11000ed3 , :Cast_5524ece6925141718d001009
1102641     , :Cast_5524ecf5c3a3687e08000dc2 , :Cast_555844da9251412afe0013a9 ,
1102642     :Cast_55872027c3a3683853005074 , :Cast_55efe971c3a368090c00cd1b , :Cast_55efe980c3a36871bf008176
1102643     , :Cast_55efe98e9251413e3201d316 , :Cast_58f2135ac3a3682e95008b91 ,
1102644     :Cast_59ac0240c3a3682cc802c399 , :Cast_59ac02cd925141079d02b1b4 ;
1102645     :crew :Crew_59ac067c92514107af02c8c8 , :Crew_52fe4ee7c3a36847f82afad7 ,
1102646     :Crew_5524ed25c3a3687ded000d88 , :Crew_5524ed2d925141720c001128 , :Crew_5524ed3d92514166c1004a5d
1102647     , :Crew_5524ed4bc3a3687df3000dd2 , :Crew_5524ed5a925141720c00112c ,
1102648     :Crew_5524ed85c3a3687e0e000f56 , :Crew_5524ed9fc3a3687e0e000f59 , :Crew_5524eda6c3a3687e03000d28
1102649     , :Crew_5524edb4925141720c00113d , :Crew_5524edc1925141727600102e ,
1102650     :Crew_5524edd192514171cb008257 , :Crew_555ad9be9251411e5b00d485 , :Crew_5677e93bc3a36816890087dc
1102651     , :Crew_5677e96a92514179e10093d0 , :Crew_5677e98492514179d2008cd9 ,
1102652     :Crew_5677e9d5c3a368168e009414 , :Crew_5677f89d9251417845001a61 , :Crew_5677f8b392514179dd0089fb
```

Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

SPARQL Protocol And RDF Query Language

- Standardisiert durch das W3C
- Aktuelle Version 1.1
- Spezifikation <https://www.w3.org/TR/sparql11-query/>

Einfache SELECT Query

Spezifiziert den base namespace

```
PREFIX : <https://www.themoviedb.org/kaggle-export/>
```

```
SELECT ?t
```

SELECT Teil: Query Variable (fängt mit '?' an)

```
WHERE {
```

```
?m a :Movie ;
```

```
    :title ?t .
```

```
}
```

QUERY Bedingung (WHERE Teil):
RDF Tripel mit Variablen legen ein
Muster fest.

Alle Zuweisungen an Variablen in
dem Wissensgraph, die genau auf
das Muster passen, werden als
Ergebnis der Query zurückgeliefert.

Query Ausführung mit Apache Fuseki

Apache Jena Fuseki datasets manage help server status

/TMDB

query add data edit info

SPARQL Query

To try out some SPARQL queries against the selected dataset, enter your query here.

Example Queries **Selection of triples** Selection of classes

SPARQL Endpoint: /TMDB/query

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

Prefixes: rdf rdfs owl xsd

```

1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
2 SELECT ?t
3 WHERE {
4   ?m a :Movie ;
5   :title ?t .
6 }

```

Table Response 3000 results in 0.189 seconds Simple view Ellipse Filter query results Page size: 50

	t
1	Cabin in the Sky
2	The Women
3	The Lost City of Z
4	Candy Razors
5	Earthquake
6	Out Cold
7	Videocracy
8	Lift

Datensatz

SPARQL Query

Ergebnis der Query

SELECT * um alle Query-Variablen zu selektieren

```
1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
2
3 SELECT *
4 WHERE {
5   ?s a :Movie ;
6   :title ?t .
7 }
```

Tabelle mit dem Ergebnis der Query:
Eine Spalte für jede Query-Variable (?s und ?t)

Table Response 3000 results in 0.099 seconds

Simple view ☐ Ellipse ☒ Enter query results Page size: 50

	s	t
1	<https://www.themoviedb.org/kaggle-export/Movie_1...	Cabin in the Sky
2	<https://www.themoviedb.org/kaggle-export/Movie_2...	The Women
3	<https://www.themoviedb.org/kaggle-export/Movie_2...	The Lost City of Z
4	<https://www.themoviedb.org/kaggle-export/Movie_1...	Candy Razors
5	<https://www.themoviedb.org/kaggle-export/Movie_2...	Earthquake
6	<https://www.themoviedb.org/kaggle-export/Movie_8...	Out Cold

Mehrere Query-Variablen

1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
 2
 3 SELECT ?t ?o
 4 WHERE {
 5 ?s a :Movie ;
 6 :title ?t ;
 7 :overview ?o .
 8 }

Query-Variablen

Ergebnis-Tabelle:
Eine Spalte für jede Query-Variable

Press CTRL - <spacebar> to auto

Table Response 2992 results in 0.164 seconds

Simple view Ellipse Filter query results Page size: 50

	t	o
1	Cabin in the Sky	Chronic gambler and carouser 'Little' Joe Jackson is shot by Domino Johnson at Jim Henry's gam...
2	The Women	Wealthy Mary Haines is unaware her husband is having an affair with shopgirl Crystal Allen. Sylv...
3	The Lost City of Z	At the dawn of the 20th century, British explorer Percy Fawcett journeys into the Amazon, where...
4	Candy Razors	Jose Manuel Gomez Perales, 'El Jaro', lives alone in Madrid, with no other company than his ban...
5	Earthquake	Earthquake is a 1974 American disaster film that achieved huge box-office success, continuing t...
6	Out Cold	Animal House meets Casablanca in this outrageous snowboarding comedy. Rick Rambis and his ...

Queries mit Bedingungen

```
1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
2
3 SELECT ?o
4 WHERE {
5   ?s a :Movie ;
6   :title "A Beautiful Mind";
7   :overview ?o .
8 }
```

Konkrete Werte als Prädikate oder Objekte schränken das Ergebnis ein

Table Response 1 result in 0.009 seconds

Simple view ☐ Ellipse ☐ Filter query results Page size: 50

	o
1	At Princeton University, John Nash struggles to make a worthwhile contribution to serve as his legacy to the world of mathematics. He finally makes a revolutionary breakthrough that will eventually earn him the Nobel Prize. After graduate school he turns to teaching, becoming romantically involved with his student Alicia. Meanwhile the government asks his help with breaking Soviet codes, which soon gets him involved in a terrifying conspiracy plot. Nash grows more and more paranoid until a discovery that turns his entire world upside down. Now it is only with Alicia's help that he will be able to recover his mental strength and regain his status as the great mathematician we know him as today..

Joins

```

1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
2
3 SELECT ?m ?t ?g ?n
4 WHERE {
5   ?m a :Movie ;
6     :title ?t ;
7     :genres ?g .
8   ?g :name ?n .
9 }

```

Wenn die gleiche Variable (hier ?g) an verschiedenen Stellen der Query Bedingung benutzt wird, erzwingt sie jedesmal den gleichen Wert

Table Response 7511 results in 0.195 seconds

Simple view ☒ Ellipse ☐ Filter query results Page size: 50

m	t	g	n
https://www.themoviedb.org/kaggle-export/Movie_1622	Cabin in the Sky	https://www.themoviedb.org/kaggle-export/Genre_14	Fantasy
https://www.themoviedb.org/kaggle-export/Movie_1622	Cabin in the Sky	https://www.themoviedb.org/kaggle-export/Genre_10402	Music
https://www.themoviedb.org/kaggle-export/Movie_2191	The Women	https://www.themoviedb.org/kaggle-export/Genre_18	Drama
https://www.themoviedb.org/kaggle-export/Movie_2191	The Women	https://www.themoviedb.org/kaggle-export/Genre_35	Comedy
https://www.themoviedb.org/kaggle-export/Movie_235	The Lost City of Z	https://www.themoviedb.org/kaggle-export/Genre_18	Drama

DISTINCT Queries

```

1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
2
3 SELECT ?n
4 WHERE {
5   ?m a :Movie ;
6     :title ?t ;
7     :genres ?g .
8   ?g :name ?n .
9 }

```

Table Response 7511 results in 0.088 seconds

Simple view ☒ Ellipse ☐ Filter query results

n
Fantasy
Music
Drama
Comedy
Drama
History
Adventure
Action
Crime
Drama
Thriller

```

1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
2
3 SELECT DISTINCT ?n
4 WHERE {
5   ?m a :Movie ;
6     :title ?t ;
7     :genres ?g .
8   ?g :name ?n .
9 }

```

Table Response 20 results in 0.091 seconds

Simple view ☒ Ellipse ☐ Filter query results

n
Fantasy
Music
Drama
Comedy
History
Adventure
Action
Crime
Thriller
Documentary
Foreign

Vermeidet
Duplikate in den
Ergebnissen

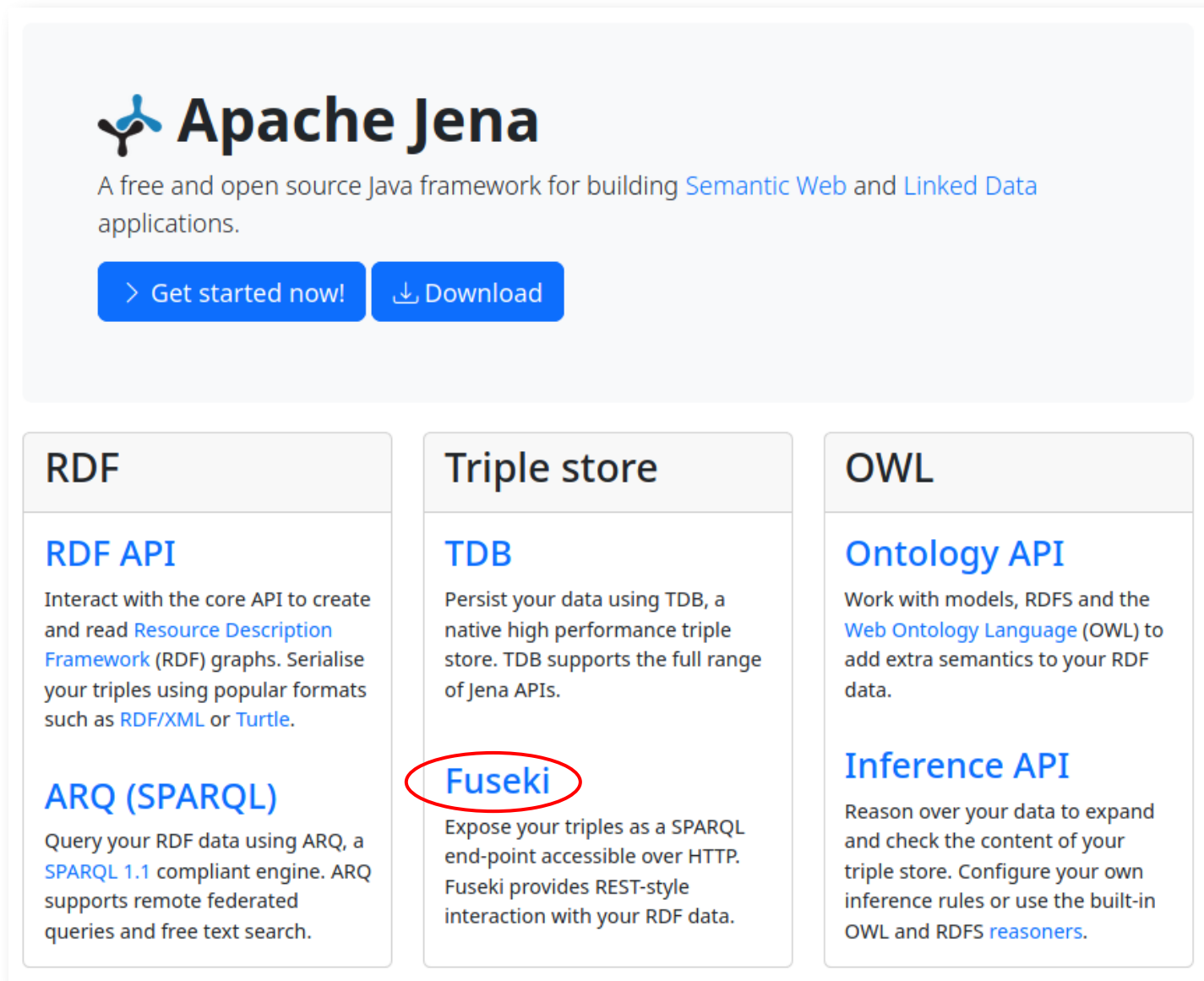
Weitere SPARQL Konzepte

- ASK Queries: Prüft eine Bedingung und liefert einen Wahrheitswert als Ergebnis zurück
- COUNT Queries: Zählt die Anzahl der Ergebnisse
- FILTER: Ermöglicht es, zusätzliche Bedingungen zu formulieren, z.B. für Datentypen
- OPTIONAL: Spezifiziert optionale Werte
- EXISTS / NOT EXISTS: Negation
- GROUP BY, HAVING: Aggregation
- ORDER BY: Sortierung
- Subqueries
- CONSTRUCT Queries: Erzeugt neue RDF-Statements als Produktionsregeln
- ...
- Details finden man hier: <http://www.w3.org/TR/sparql11-query>

Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

Apache Jena: Java Framework für Linked Data



The screenshot shows the Apache Jena website. At the top, there is the Apache Jena logo and a description: "A free and open source Java framework for building Semantic Web and Linked Data applications." Below this are two buttons: "Get started now!" and "Download". The main content is organized into three columns: RDF, Triple store, and OWL. The RDF column contains "RDF API" and "ARQ (SPARQL)". The Triple store column contains "TDB" and "Fuseki" (which is circled in red). The OWL column contains "Ontology API" and "Inference API".

Apache Jena
A free and open source Java framework for building [Semantic Web](#) and [Linked Data](#) applications.

[Get started now!](#) [Download](#)

RDF	Triple store	OWL
RDF API Interact with the core API to create and read Resource Description Framework (RDF) graphs. Serialise your triples using popular formats such as RDF/XML or Turtle .	TDB Persist your data using TDB, a native high performance triple store. TDB supports the full range of Jena APIs.	Ontology API Work with models, RDFS and the Web Ontology Language (OWL) to add extra semantics to your RDF data.
ARQ (SPARQL) Query your RDF data using ARQ, a SPARQL 1.1 compliant engine. ARQ supports remote federated queries and free text search.	Fuseki Expose your triples as a SPARQL end-point accessible over HTTP. Fuseki provides REST-style interaction with your RDF data.	Inference API Reason over your data to expand and check the content of your triple store. Configure your own inference rules or use the built-in OWL and RDFS reasoners .

Quelle: <https://jena.apache.org/>

Apache Jena Fuseki SPARQL Server

Apache Jena Fuseki

Apache Jena Fuseki is a SPARQL server. It can run as an operating system service, as a Java web application (WAR file), and as a standalone server.

Fuseki comes in two forms, a single system "webapp", combined with a UI for admin and query, and as "main", a server suitable to run as part of a larger deployment, including [with Docker](#) or running embedded. Both forms use the same core protocol engine and [same configuration file format](#).

Fuseki provides the SPARQL 1.1 [protocols for query and update](#) as well as the [SPARQL Graph Store protocol](#).

Fuseki is tightly integrated with [TDB](#) to provide a robust, transactional persistent storage layer, and incorporates [Jena text query](#).

Contents

- [Download with UI](#)
- [Getting Started](#)
- [Running Fuseki with UI](#)
 - [As a standalone server with UI](#)
 - [As a service](#)
 - [As a web application](#)
 - [Security with Apache Shiro](#)
- [Running Fuseki Server](#)
 - [Setup](#)
 - [As a Docker container](#)
 - [As an embedded SPARQL server](#)
 - [Security and data access control](#)
 - [Logging](#)
- [Fuseki Configuration](#)
- [Server Statistics and Metrics](#)
- [How to Contribute](#)
- [Client access](#)
 - [Use from Java](#)
 - [SPARQL Over HTTP](#) - scripts to help with data management.
- [Extending Fuseki with Fuseki Modules](#)
- [Links to Standards](#)

The Jena users mailing is the place to get help with Fuseki.

[Email support lists](#)

Download Fuseki with UI

Releases of Apache Jena Fuseki can be downloaded from one of the mirror sites:

[Jena Downloads](#)

and previous releases are available from [the archive](#). We strongly recommend that users use the latest official Apache releases of Jena Fuseki in preference to any older versions.

Fuseki download files

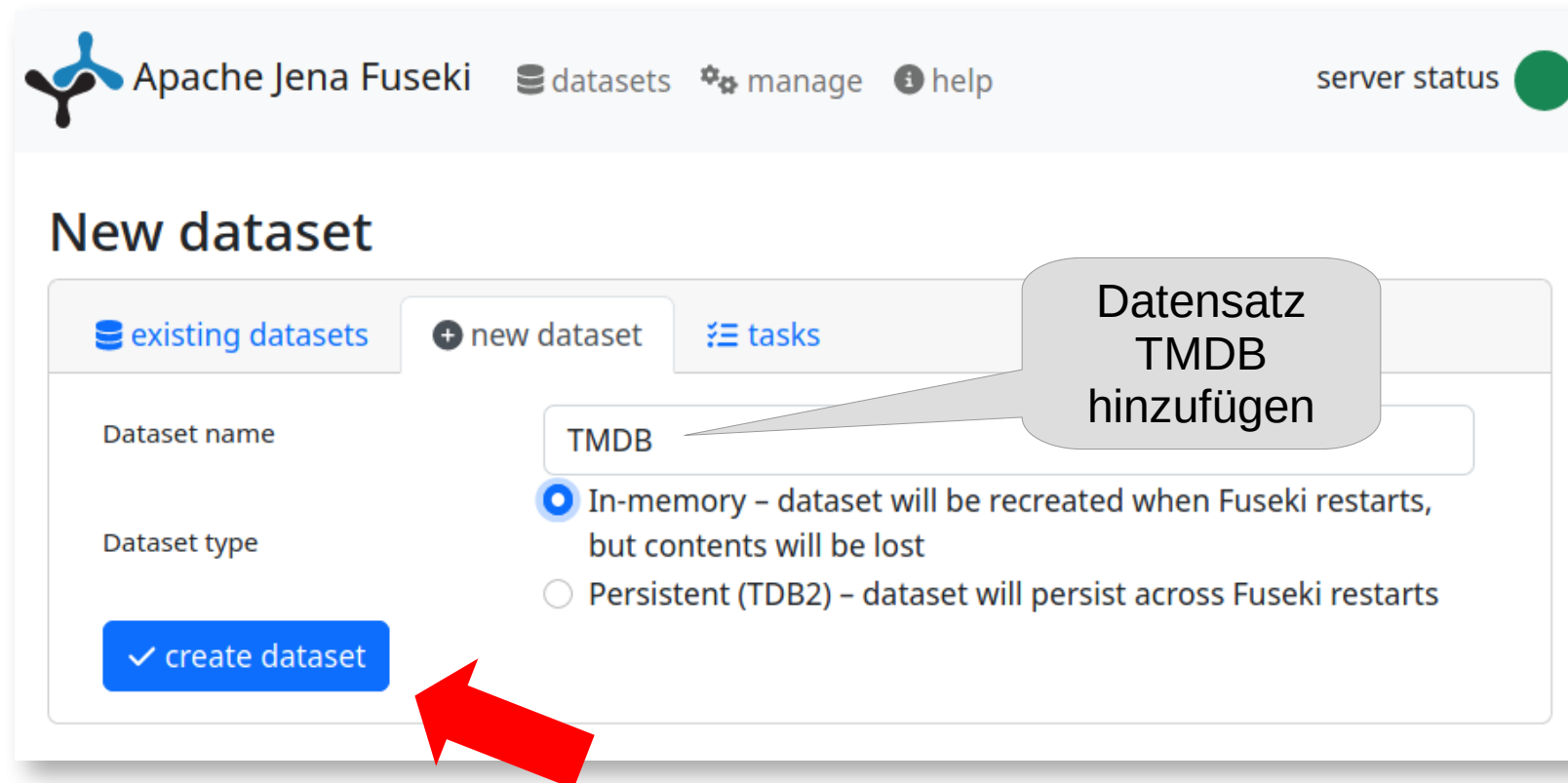
Filename	Description
----------	-------------

Fuseki Download
zum Installieren

Quelle: <https://jena.apache.org/documentation/fuseki2/>

Fuseki Server und Web Client starten und den Datensatz TMDB hinzufügen

- Server läuft auf <http://localhost:3030>



The screenshot shows the Apache Jena Fuseki web interface. The top navigation bar includes the Fuseki logo, the text 'Apache Jena Fuseki', and links for 'datasets', 'manage', and 'help'. On the right, there is a 'server status' indicator with a green circle. The main content area is titled 'New dataset' and contains three tabs: 'existing datasets', 'new dataset' (selected), and 'tasks'. Below the tabs, there are two input fields: 'Dataset name' with the text 'TMDB' and 'Dataset type' with two radio button options. The first option, 'In-memory - dataset will be recreated when Fuseki restarts, but contents will be lost', is selected. The second option is 'Persistent (TDB2) - dataset will persist across Fuseki restarts'. At the bottom left of the form is a blue button labeled '✓ create dataset'. A red arrow points to this button. A grey callout box with a pointer to the 'TMDB' text in the 'Dataset name' field contains the text 'Datensatz TMDB hinzufügen'.

Apache Jena Fuseki datasets manage help server status

New dataset

existing datasets + new dataset tasks

Dataset name TMDB

Dataset type

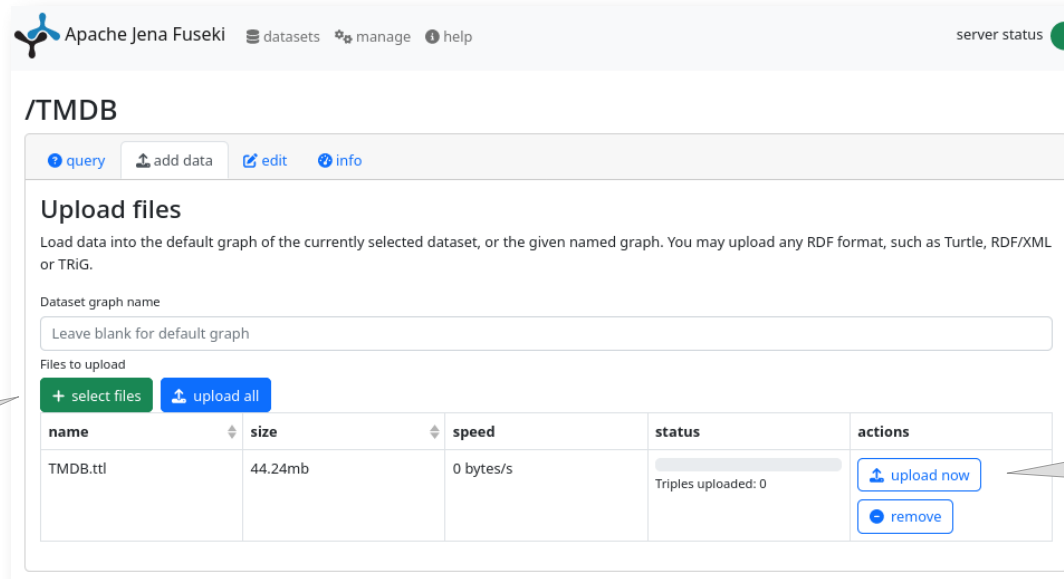
☒ In-memory - dataset will be recreated when Fuseki restarts, but contents will be lost

☐ Persistent (TDB2) - dataset will persist across Fuseki restarts

✓ create dataset

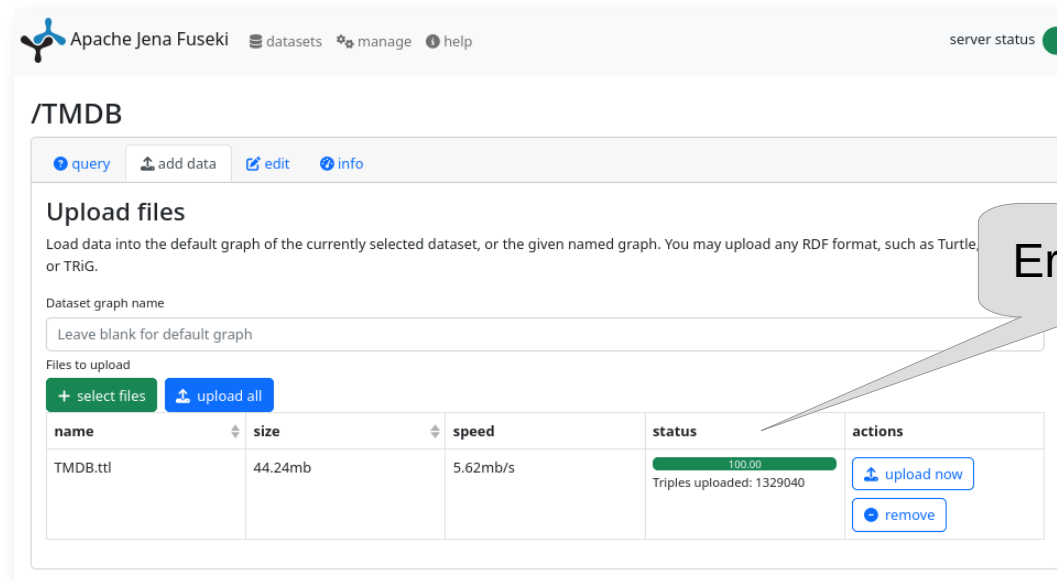
Datensatz TMDB hinzufügen

Wissensgraph TMDb.ttl laden



Zuerst die Datei TMDb.ttl auswählen

Und anschließend hochladen



Ergebnis

SPARQL Queries schreiben und ausführen

```

1 PREFIX : <https://www.themoviedb.org/kaggle-export/>
2
3 SELECT ?m ?t ?g ?n
4 WHERE {
5   ?m a :Movie ;
6     :title ?t ;
7     :genres ?g .
8   ?g :name ?n .
9 }

```

7511 results in 0.195 seconds

Simple view ☒ Ellipse ☐ Filter query results Page size: 50

m	t	g	n
https://www.themoviedb.org/kaggle-export/Movie_1622	Cabin in the Sky	https://www.themoviedb.org/kaggle-export/Genre_14	Fantasy
https://www.themoviedb.org/kaggle-export/Movie_1622	Cabin in the Sky	https://www.themoviedb.org/kaggle-export/Genre_10402	Music
https://www.themoviedb.org/kaggle-export/Movie_2191	The Women	https://www.themoviedb.org/kaggle-export/Genre_18	Drama
https://www.themoviedb.org/kaggle-export/Movie_2191	The Women	https://www.themoviedb.org/kaggle-export/Genre_35	Comedy
https://www.themoviedb.org/kaggle-export/Movie_235	The Lost City of Z	https://www.themoviedb.org/kaggle-export/Genre_18	Drama

SPARQL Query
ausführen

Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

rdflib: Ein Python package für RDF/SPARQL

pip install rdflib

python3 -m pip install --user rdflib

rdflib 6.3.2

✓

Latest version

pip install rdflib

Released: Mar 26, 2023

RDFLib is a Python library for working with RDF, a simple yet powerful language for representing information.

Navigation

Project description

Release history

Download files

Project links

Homepage

Documentation

Repository

Statistics

GitHub statistics:

★ Stars: 1889

Project description

RDFLib

Validate

passing

docs

passing

coverage

91%

stars

1.9k

downloads/week

180k

pypi

v6.3.2

python

3.7 | 3.8 | 3.9 | 3.10 | 3.11

DOI

10.5281/zenodo.6845245

Contribute with

Gitpod

chat

on gitter

matrix.org chat

62 users

RDFLib is a pure Python package for working with [RDF](#). RDFLib contains most things you need to work with RDF, including:

- parsers and serializers for RDF/XML, N3, NTriples, N-Quads, Turtle, TriX, Trig and JSON-LD
- a Graph interface which can be backed by any one of a number of Store implementations
- store implementations for in-memory, persistent on disk (Berkeley DB) and remote SPARQL endpoints
- a SPARQL 1.1 implementation - supporting SPARQL 1.1 Queries and Update statements
- SPARQL function extension mechanisms

Quelle: <https://pypi.org/project/rdflib/>

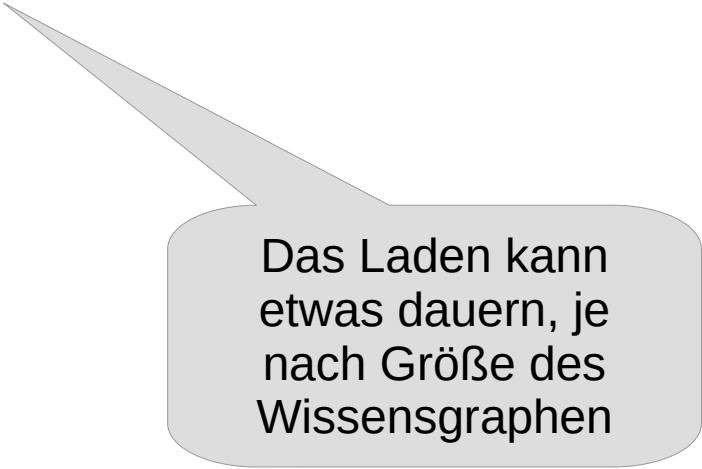
h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Einführung in die künstliche Intelligenz • Wissensrepräsentation • Prof. Dr. A. del Pino

Einen Wissensgraph mit rdflib laden

```
import rdflib  
  
g = rdflib.Graph()  
g.parse('data/TMDB.ttl', format='turtle')
```



Das Laden kann
etwas dauern, je
nach Größe des
Wissensgraphen

SPARQL Queries ausführen

```
In [15]: 1 query_result = g.query(  
2      """  
3      SELECT distinct ?gg  
4      WHERE {  
5          ?m a :Movie ;  
6          :genres ?g .  
7  
8          ?g :name ?gg .  
9      }  
10     """  
11 )  
12  
13 for row in query_result:  
14     print(row[0])
```

Ergebnis der Query
(Typ SPARQLResult)
kann iteriert werden

```
Comedy  
Drama  
Family  
Romance  
Thriller  
Action  
Animation  
Adventure  
Horror  
Documentary  
Music  
Crime  
Science Fiction  
Mystery  
Foreign  
Fantasy  
War  
Western  
History  
TV Movie
```

Aufbau

- Überblick
- Wissensgraphen
- RDF
- Der TMDb Wissensgraph
- SPARQL
- Apache Fuseki
- RDF und SPARQL in Python verwenden
- Mini-Test

Mini-Test “Wissensrepräsentation”

- Was ist ein Wissensgraph?
- Nennen Sie Beispiele für Objekte, Klassen und Beziehungen
- Was ist eine Resource in RDF?
- Aus welchen Teilen besteht ein RDF Tripel?
- Nennen Sie die Klassen des TMDb Wissensgraph, welche in diesem Kurs verwendet werden
- Was ist SPARQL?
- Wie sehen einfache Queries in SPARQL aus?
- Wie viele Zeilen und Spalten hat das Ergebnis einer Query?
- Wie drückt man in Queries Bedingungen aus?
- Wie funktioniert ein Join in SPARQL?
- Wie vermeidet man Duplikate in dem Ergebnis einer Query?