Last time: Graphs

- Graphs can represent general knowledge or facts
- Semantic networks are labelled, typically directed, graphs; defined labels are associated with specific meaning (semantics). How to reason using these labels needs to be defined via dedicated procedures.
  - Graphs can be used to represent logic statements.
- Reasoning over graphs:
  - *Logic*
  - Spreading activation
  - Graph measures.

Perceiving, thinking, acting – what did we set out to learn in this lecture?

Perceive

Data and natural language as input to artificial intelligence-enabled systems, or artificial intelligence analytics methods

Think

Different knowledge representations and reasoning mechanisms, and the concept of machine learning.

Act

Rather indirectly covered – in the sense that system/algorithm output informs human users or is consumed by other systems.

Thinking – what have we learned so far?

Knowledge Representation

Express and differentiate between:

- Facts (assertional axioms - logic, graphs)
- General knowledge (rules; terminological axioms – predicate logic,graphs)

Reasoning

Logic, graph-based procedures: Infer new knowledge, check for the correctness of specific statements, can find related entities (spreading activation, shortest path), and characterize entities with different graph measures.

Learning

(no learning yet)

Problems with knowledge representations from symbolic AI

Symbolic AI – relies on expressing knowledge in terms of human-readable, high-level symbolic expressions and concepts – in our lecture: rules, logic, and semantic networks

- **Suitability to real-world problems:** It is difficult to describe complex entities in logic-based, symbolic KR formalisms (graph representations help to some degree
- **Modelling effort:** Symbolic knowledge representations are typically manually engineered (rules, object-oriented KR, knowledge graphs)
  - Vocabulary and formal KR needs to be modelled, is difficult to derive automatically (loosening requirements on formality helps)
  - **Suitable for metadata** and common characteristics (attributes, properties, …)
  - Suitable if formal description is sufficiently important – e.g., formal verification of important systems.
  - Data graphs are typically automatically created, trade-offs in content quality typically are accepted
- **Reasoning:** New inferences, validation, consistency checking. Scalability and decidability can be an issue.

WISSEN
TECHNIK
LEIDENSCHAFT

Vectors as Knowledge Representation

Viktoria Pammer-Schindler

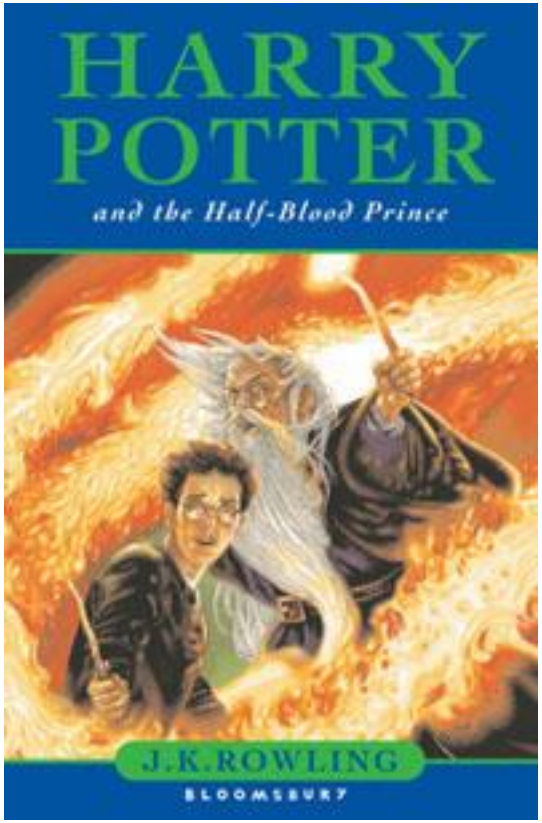
Introduction to Data Science and Artificial Intelligence

www.staff.tugraz.at/viktoria.pammer-schindler

Learning Goals

- Understand and explain vectors as representations of complex items
  - Give example symbolic descriptions of complex items
  - Give example vector representations of complex items
- Understand and explain vector operations as way to reason over vector representations
- Understand and be able to compute cosine similarity as vector-based similarity measure
- *Understand the relevance of similarity measures in computational problems (this, plus next lectures)*

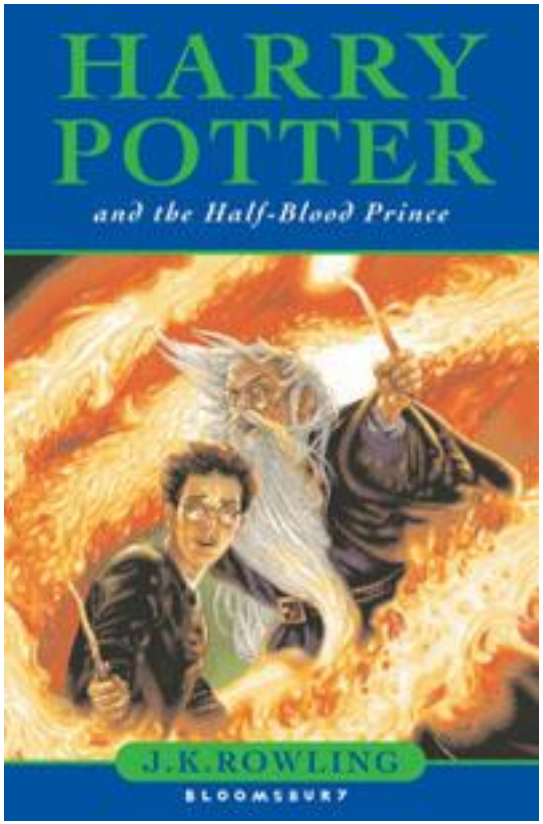
Representation



Goal: Computationally represent books, e.g., Harry Potter and the Half-Blood Prince.

Definition: A **representation** Y conforms in a systematic manner to X , preserving pre-selected characteristics of X.

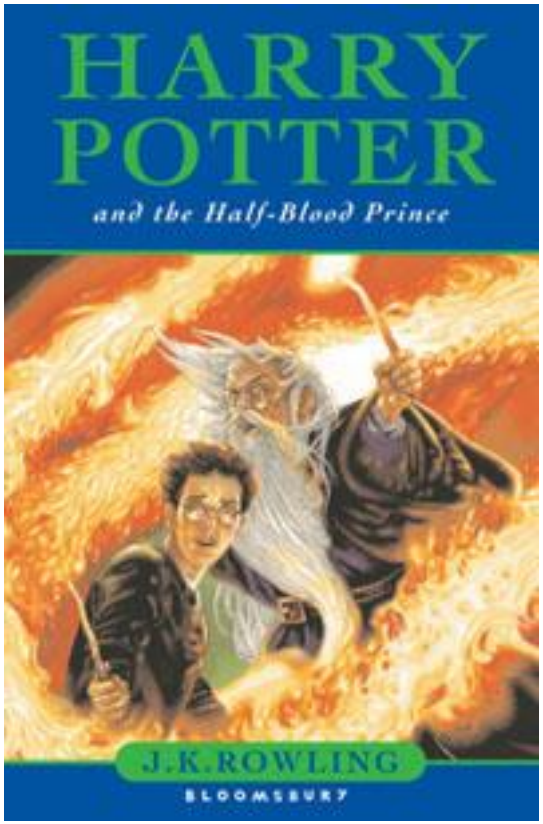
A representation always loses something, it is an approximation.



Represent this book in predicate logic:

- Book(Harry-potter-hbp).
- hasCoverColor(Harry-potter-hbp, green)
- isGenre(Harry-potter-hbp, fantasy-novel)
- isGenre(Harry-potter-hbp, young-adult-novel)
- ...

- Fictional-Character(Harry)
- Fictional-Character(Hermione)
- Fictional-Character(Ron).
- appearsIn(Harry,Harry-potter-hbp)
- …
- is-friend-of(Harry,Hermione)
- is-friend-of(Harry,Ron)
- ...



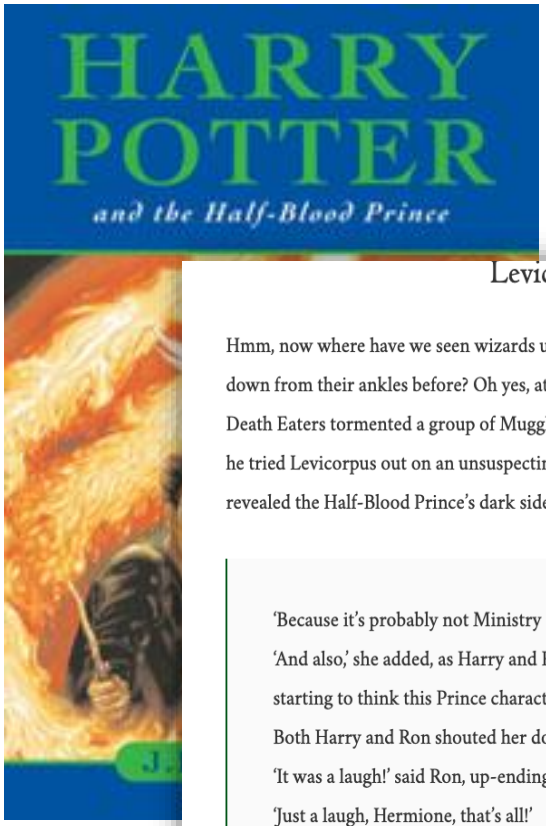
How suitable is predicate logic as a knowledge representation formalism to represent books?

\+ Suitable for expressing meta-data, and describing characteristics of the book that are similar across books – facts and knowledge ABOUT the book

- Imagine expressing the complete story of Harry Potter in predicate logic!
- Yahoo! initially categorised web sites in a web directory – required editors

~ Metadata could be extracted from a digital version of the book with some heuristics, if reasonable metadata schema exists a priori (-> typical research/engineering goal in NLP „fact extraction“, „slot filling“, „named entity recognition“)

Idea: Vectors as numeric, non-symbolic representation of complex entities



Levicorpus

Hmm, now where have we seen wizards using a spell that suspends people upside-down from their ankles before? Oh yes, at the Quidditch World Cup when those Death Eaters tormented a group of Muggles. As Hermione pointed out to Harry after he tried Levicorpus out on an unsuspecting (and soundly asleep) Ron, this spell revealed the Half-Blood Prince's dark side. If only they knew...

'Because it's probably not Ministry of Magic-approved,' said Hermione.
'And also,' she added, as Harry and Ron rolled their eyes, 'because I'm starting to think this Prince character was a bit dodgy.'
Both Harry and Ron shouted her down at once.
'It was a laugh!' said Ron, up-ending a ketchup bottle over his sausages.
'Just a laugh, Hermione, that's all!'
'Dangling people upside-down by the ankle?' said Hermione. 'Who puts their time and energy into making up spells like that?'
Harry Potter and the Half-Blood Prince

$$\begin{pmatrix} 17 \\ 0{,}5 \\ 3 \\ 4{,}2 \\ 0{,}8 \\ 2{,}33 \\ \dots \end{pmatrix}$$

From items to vectors, and how to reason using vectors

1. Choose KR formalism: Vectors
2. Choose how to represent items into vectors (~feature engineering).
   - Choose relevant characteristics of the item!

   KR perspective: An entity has a set of characteristics; in machine learning these correspond approximately to „features“; and in statistics the term „variables“ is more typical.
   - Choose easily computable characteristics of the item!

   Engineering perspective
   - … which are representable as real numbers.

   Reasoning perspective (want to use vector mathematics)
3. Choose operations on the represented items
   - Vector mathematics has a broad range of operations – what do they mean/which are useful

Choosing operations on vectors for reasoning - What questions do we want to ask?

?

For a given entity:

- **What are similar entities?**
- To which groups does an entity belong? (**classification**)
- What will we be able to observe about this entity in the future? (prediction)

$$\begin{pmatrix} 17 \\ 0{,}5 \\ 3 \\ 4{,}2 \\ 0{,}8 \\ 3{,}8 \\ \dots \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0{,}5 \\ 3 \\ 4{,}2 \\ ,8 \\ 2{,}33 \\ \dots \end{pmatrix} \quad \begin{pmatrix} 17 \\ 0{,}7 \\ 4 \\ 4{,}2 \\ 0{,}8 \\ 2{,}33 \\ \dots \end{pmatrix} \quad \begin{pmatrix} 17 \\ 0{,}5 \\ 3 \\ 4{,}2 \\ 0{,}8 \\ 2{,}33 \\ \dots \end{pmatrix}$$

Over a set of entities:

- What are meaningful sub-groups? (clustering)
- Is there a **correlation** between different entity characteristics?
- Does one characteristic cause another one? (attention – typically needs specific study set-up to assert!)
- Is there a more compact representation – which variables carry most information? (factor analysis)

Why is similarity interesting?

- **Recommendation**: If you like book A, and book B is similar to A, then you will probably also like book B.
- **Information retrieval**: If your question (query) is q, then the answer to your question (a document) should be similar to q.
- **Classification**: All members of a class are similar w.r.t. to specific features
- **Clustering**: Build groups of entities that are more similar to each other than they are to members of other groups.

Vector Representations in this lecture

In this lecture, we will

- Compute similarities (today – Lecture 5)
  - Application examples: information retrieval and recommender systems (Lectures 6 and 7)
- Translate natural language into vectors (Lecture 6)
- Classify items using artificial neural networks (Lectures 8,9)

Discussion: Vectors as Knowledge Representation

- Allow us to use everything from simple to complex functions to compute single entries in the vector („features“).
- Allow us to use vector mathematics as a way to reason over knowledge
  - Complex operations on vectors available – including the whole field of statistics!
- Feature engineering is still knowledge-intensive…
- … but if easily computable features are chosen, once the features are chosen, the representation of a concrete entity is easy (typical: choose a representation that can be created automatically).

Is it a new way to structure knowledge, or is it a data structure?

A single vector represents a data “point”, an instance

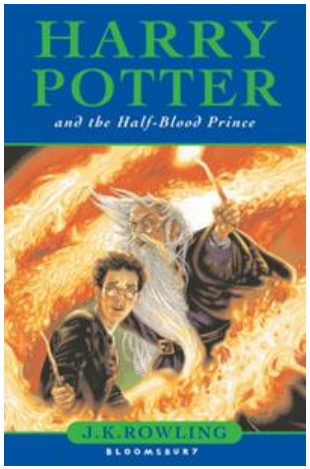
The choice of which entities are represented is a knowledge engineering choice

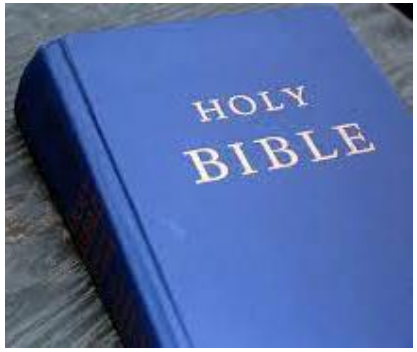
… as is the choice of how to represent every single instance (sometimes called “feature set”, the dictionary vector is an example from natural language text information retrieval)

WISSEN
TECHNIK
LEIDENSCHAFT

Similarity Measures – Cosine Similarity

How similar are two entities?

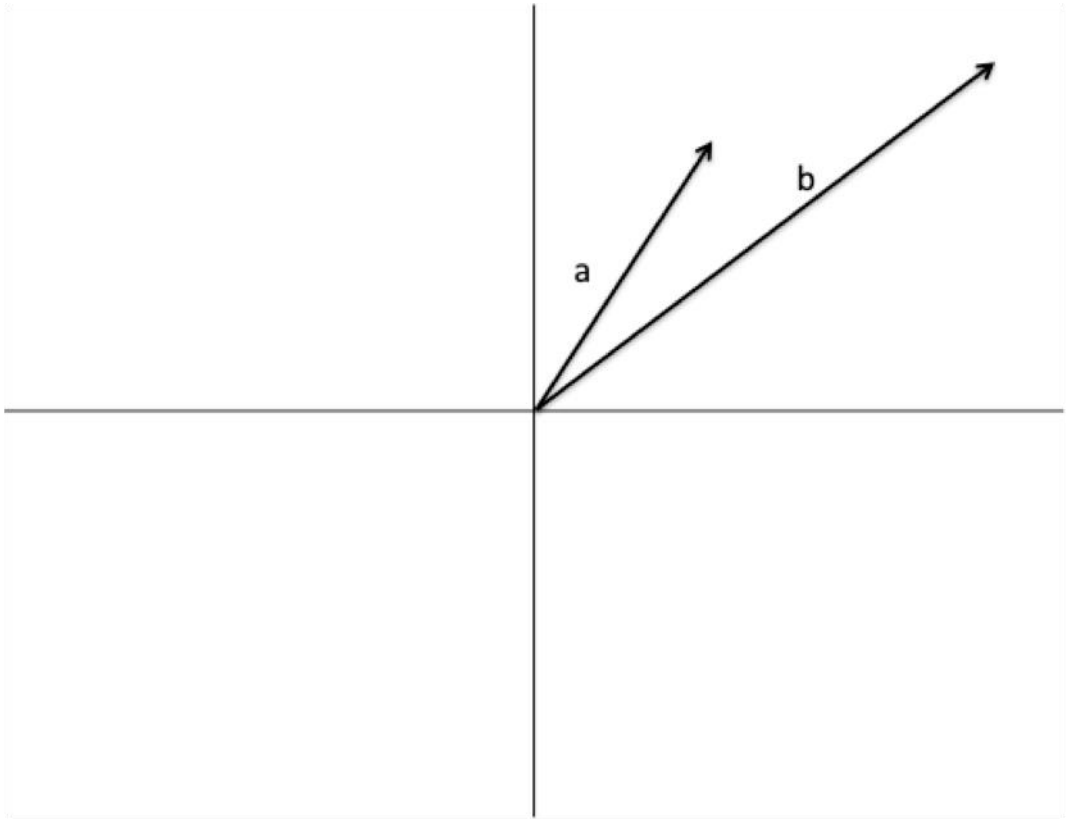




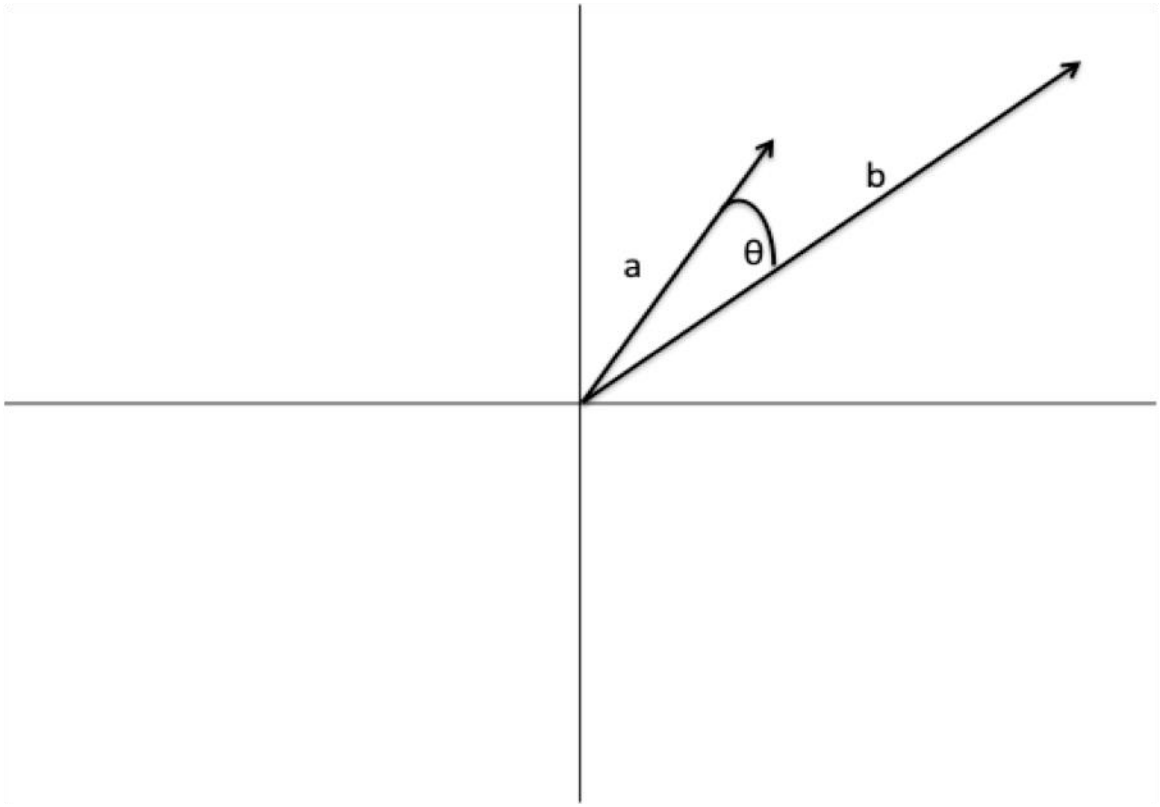
What are characteristics for comparison?

-> Choice in representation

How similar are two entities when they are represented as vectors?

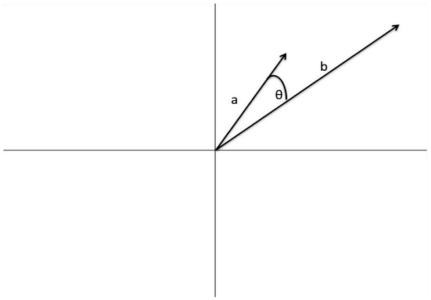


Cosine similarity – a widely used measure of similarity between two vectors



Cosine similarity – values it takes

Values are between [-1;1]



- Angle is 0° => Cos(0°)=1
  - Vectors have the same direction, and are maximally similar. Cosine doesn't measure vector equality!!!
- Angle is 90 ° => Cos(90°)=0
  - Vectors are orthogonal; they are not similar at all.
- Angle is 180° => Cos(180°)=-1
  - Vectors show into the opposite direction, they are inverse.

Cosine similarity - formula

$$sim(a, b) = \cos(\theta) = \frac{a \cdot b}{\|a\| \cdot \|b\|}$$

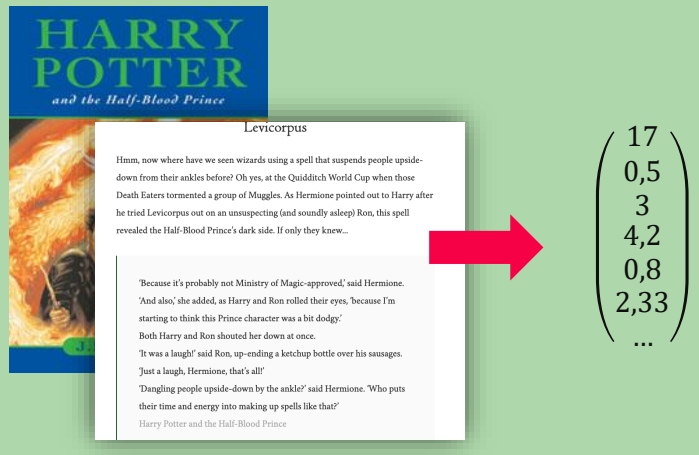
- Independent of the lengths (magnitude) – normalizes vectors.
- Measures direction

Exercise 8

Example: Online bookstore

25

What do we want to do with digital representations of book?



- We have chosen to represent books as vectors
- **Questions (leaning on Slide 12):**
  - For which use cases do I need electronic representations of books? For which of them could a vector representation be useful?
  - What could be different types of groups, into which you would like to be able to automatically group books?
  - What could be useful to predict about books?
  - What are two characteristics of books that you would expect to be correlated?

What do we know about our books, and where from?

Metadata we have from the publisher:

- Title, author, part of a series, genre(s), number of pages, price, ISBN

Publicly available data:

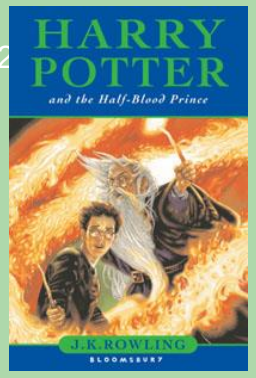
- Book reviews (text and ratings) from online sources
- Which other relevant data is publicly available?

Data that we (=bookstore) generate ourselves:

- Every time the book is bought
- Which other data could be available in-house?

Does anyone else have data that would be interesting? How could we get it?

Example Metadata Structure



Book
Title (string)
Author (string)
Part-of-series (string)
Publisher (string)
Number-of-pages (integer)
ISBN (integer)
Book-ID-internal (integer)

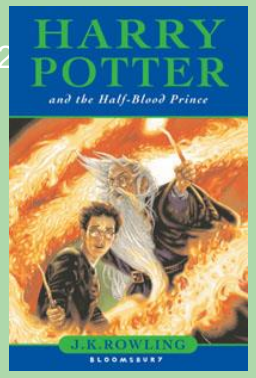
is Genre

Genre
Genre-name (string)

has Price

Price
Price-value
Price-currency

Metadata



Genre
Genre-name (string)

is Genre

Book
Title (string)
Author (string)
Part-of-series (string)
Publisher (string)
Number-of-pages (integer)
ISBN (integer)
Book-ID-internal (integer)

has Price

Price
Price-value
Price-currency

Is_a

Price-20E
Price-value: 20
Price-currency: €

Fantasy
Genre-name: Fantasy

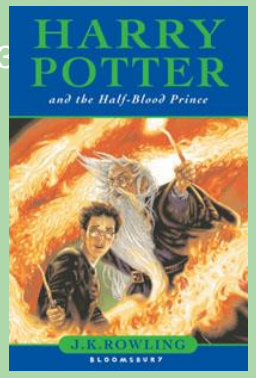
has Price

is Genre

Harry-Potter-and-the-HBP
Title: Harry Potter and the Half-Blood-Prince
Author: J. K. Rowling
Part of series: Harry Potter
Publisher: Bloomsbury
Number-of-pages: 607
ISBN:0-7475-8108-8

Is_a

Metadata + Data for recommendation



Metadata + Data for Inventory

Genre
Genre-name (string)

is Genre

Price
Price-value
Price-currency

Is_a

Price-20E
Price-value: 20
Price-currency: €

has Price

Book
Title (string)
Author (string)
Part-of-series (string)
Publisher (string)
Number-of-pages (integer)
ISBN (integer)
Book-ID-internal (integer)

Fantasy
Genre-name: Fantasy

has Price

is Genre

Harry-Potter-and-the-HBP
Title: Harry Potter and the Half-Blood-Prince
Author: J. K. Rowling
Part of series: Harry Potter
Publisher: Bloomsbury
Number-of-pages: 607
ISBN:0-7475-8108-8

Is_a

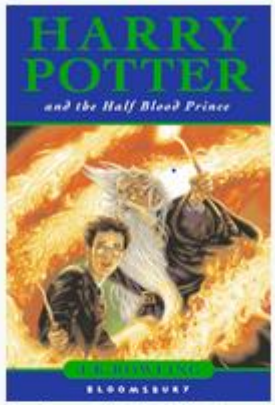
Harry-Potter-and-the-HBP-211
Book-ID-internal: 0200-211

Is_a

31

Comment: Have a look at what Wikipedia does with structured info about books

Harry Potter and the Half-Blood Prince



Cover art of the first UK edition

Author	J. K. Rowling
Illustrator	Jason Cockcroft (UK) Mary GrandPré (US)
Country	United Kingdom
Language	English
Series	*Harry Potter*
Release number	6th in series
Genre	Fantasy
Publisher	Bloomsbury (UK) (Canada 2010–present) Arthur A. Levine/ Scholastic (US) Raincoast (Canada 1998–2010)
Publication date	16 July 2005
Pages	607 (Original UK Edition) 542 (2014 UK Edition) 652 (US Edition)
ISBN	0-7475-8108-8
Dewey Decimal	823.914
Preceded by	*Harry Potter and the Order of the Phoenix*
Followed by	*Harry Potter and the Deathly Hallows*

https://en.wikipedia.org/wiki/Harry_Potter_and_the_Half-Blood_Prince

Choosing the vector representation

Metadata we have from the publisher:

- Title, author, part of a series, genre(s), number of pages, price, ISBN
- What of this is relevant for our use case, and how do we convert this information into vector elements?

Publicly available data:

- Book reviews (text and ratings) from online sources
- What of this is relevant for our use case, and how do we convert this information into vector elements?

Data that we (=bookstore) generate ourselves:

- Every time the book is bought
- What of this is relevant for our use case, and how do we convert this information into vector elements?

Exercise 9

Example Choice

Choice:

- Metadata – we use the genre. Every genre corresponds to a vector index. Everytime the book has a genre, there is a 1 in the vector, otherwise a 0.
- Publicly available data – we use the average rating from Thalia and GoodReads. Every rating-source corresponds to a vector index.
- Own data – we know every user who has bough the book. Every user corresponds to a vector index. For every user who has bought the book, there is a 1 in the vector, otherwise a 0.

Exercise: How similar are the two books „Harry Potter and the Half-Blood Prince“ and „Foundation #1“?

Dictionary-Vector	Harry-Potter-and-the-HBP	Foundation #1
$Fantasy - Genre$	1	0
$ScienceFiction - Genre$	0	1
$Thalia - AvgRating$	5	5
$GoodReads - AvgRating$	4,57	4,16
$User1$	0	1
$User2$	1	0
$User3$	1	1

Comment: for the Thalia Rating for Foundatoin I used the German book “Die Foundation-Trilogie”, as the Foundation book was not rated, and we haven’t discussed how to deal with missing values.