Welcome to INFO216: Knowledge Graphs

Spring 2024

Andreas L Opdahl
<Andreas.Opdahl@uib.no>

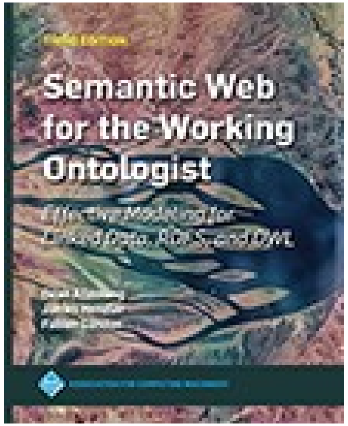
Session 7: Enterprise Knowledge Graphs

- Themes:
  - Open Knowledge Graphs *(← S05-S06)*
    - Linked Open Data resources / datasets
    - Wikidata, DBpedia, GeoNames, GDELT, WordNet, BabelNet, ConceptNet...
  - Enterprise Knowledge Graphs (EKGs)
    - Google's Knowledge Graph
    - Amazon's Product Graph
    - *Bosch' Line Information System (LIS)*
    - *the News Hunter platform*

Maybe later...

Readings

- Sources (suggested):
  - Blumauer & Nagy (2020):
    Knowledge Graph Cookbook – Recipes that Work:
    parts 2 and 4
- Resources in the wiki <http://wiki.uib.no/info216>:
  - *Introducing the Knowledge Graph: Things not Strings*, Amit Singhal, Google (2012)
  - *A reintroduction to our Knowledge Graph and knowledge panels*, Danny Sullivan, Google (2020)
  - *How Amazon's Product Graph is helping customers find products more easily*, Arun Krishnan, Amazon (2018)



Is anyone really using Knowledge Graphs?

Is anyone really using this?

Yes!

Tencent 腾讯

UniProt

USGS

Google

Bing

Baidu 百度

Alibaba.com

PubMed

facebook

ANTONI VAN LEEUWENHOEK FOUNDATION

MAASTRO

DEUTSCHE NATIONAL BIBLIOTHEK

The New York Times

BBC

europeana

GETTY

NXP

REUTERS

KUNGL. BIBLIOTEKET

National Library of Sweden

RENAULT

EPA United States Environmental Protection Agency

IOS Press

European Commission

Walmart

SIEMENS

BEST BUY

Deloitte.

ELSEVIER

SPRINGER NATURE

amazon.com

accenture

Is anyone really using this?

Yes!

- But...
  - not quite as in the semantic web vision
  - not quite as in the LOD vision either
- Knowledge graphs are (additionally) becoming:
  - company internal
  - based on other technologies
    - such as general graph databases
  - not always linked to the LOD cloud

Is anyone really using this?

Yes!

- But...
  - not quite as in the semantic web vision
  - not quite as in the LOD vision either
- Knowledge graphs are (additionally) becoming:
  - company internal
  - based on other technologies
    - such as general graph databases
  - not always linked to the LOD cloud

Many of these ideas are widely adopted too, such as:

- microdata / schema.org
- RDF / SPARQL / … for semantic data exchange
- graph representations in general

Is anyone really using this?

Yes!

- But...
  - not quite as in the semantic web vision
  - not quite as in the LOD vision either
- Knowledge graphs are (additionally) becoming:
  - company internal
  - based on other technologies
    - such as general graph databases
  - not always linked to the LOD cloud

Similar ideas,
adapted to new uses
and business contexts,
using a combination of
standard and other
technologies

Google's Knowledge Graph

Google's Knowledge Graph

- Google Knowledge Graph (from 2012)
  - "Things, not Strings"
  - seeded from Freebase
  - facts from Wikipedia, Wikidata, CIA World Factbook
    - a growing number of other sources
  - used internally for many purposes
  - visible in Google Search results (Knowledge Panels)
  - question answering in Google Assistant / Home
  - semantic API (https://developers.google.com/knowledge-graph)
    - "returns only individual matching entities, rather than graphs of interconnected entities"

Caution: *The public documentation is limited, so this is compiled based on presentations, technical notes, forums etc.*

Google's Knowledge Graph

- Coverage:
  - claimed *(but be cautious)*
    - 18 billion facts (18G, norsk: 18 milliarder)
      about 570 million entities
    - 70 billion facts claimed in (2016)
    - *500 billion facts about five billion entities (2020)*
      - ...more than 3 times the size of the LOD cloud
  - from English to multiple languages
- Critiques:
  - source attribution, incl. Wikipedia / Wikidata
  - zero-click searches (around 25% of desktop searches)

***Caution:** The public documentation is limited, so this is compiled based on presentations, technical notes, forums etc.*

Google's Knowledge Vault Project

- Attempt to extend the Knowledge Graph
  - covered resources not from open semantic datasets
  - facts extracted from the whole web
    - NLP of text documents
    - HTML trees and tables
    - human annotated pages (e.g., schema.org)
  - reported size
    - 1.6 billion facts
    - 271 million “confident” ones (90%)
  - *not put in production (never achieved 99% confidence target)*

Caution: *The public documentation is limited, so this is compiled based on presentations, technical notes, forums etc.*

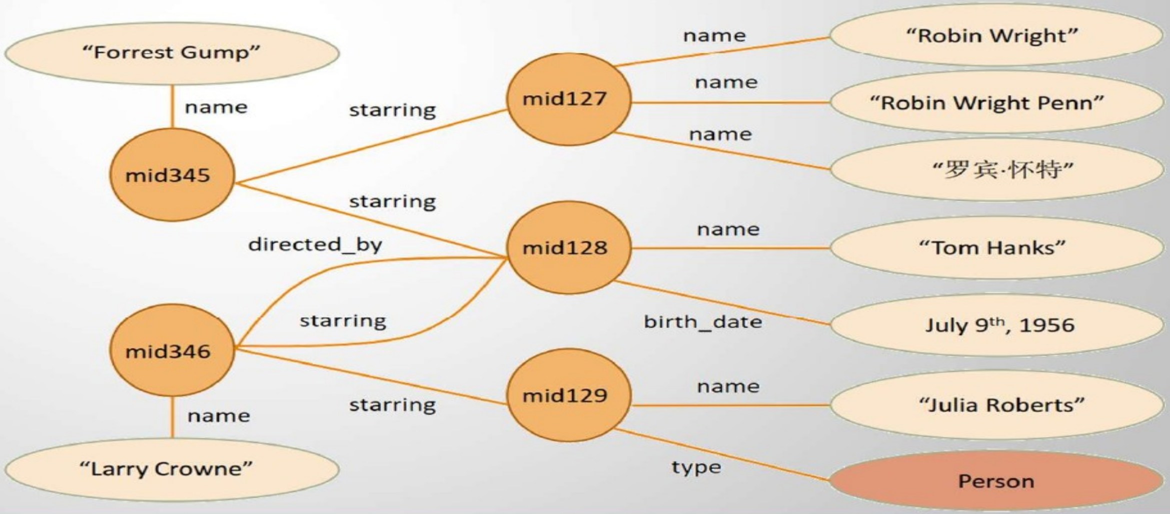
Amazon's Knowledge Graph

Amazon’s ambition (← S01)

- Let shoppers find the best products that fit their needs
  - allow greater variation in search terms
  - allow complex queries
- Ambition: *to structure all of the world’s information as it relates to everything available on Amazon*
- Describe every product on Amazon
  - both products and non-products
  - both concrete and abstract concepts
  - link related entities, both internal and external
- Enhanced customer experience
  - visit Amazon to see what’s new or interesting
  - discover ways to simplify and enrich their lives

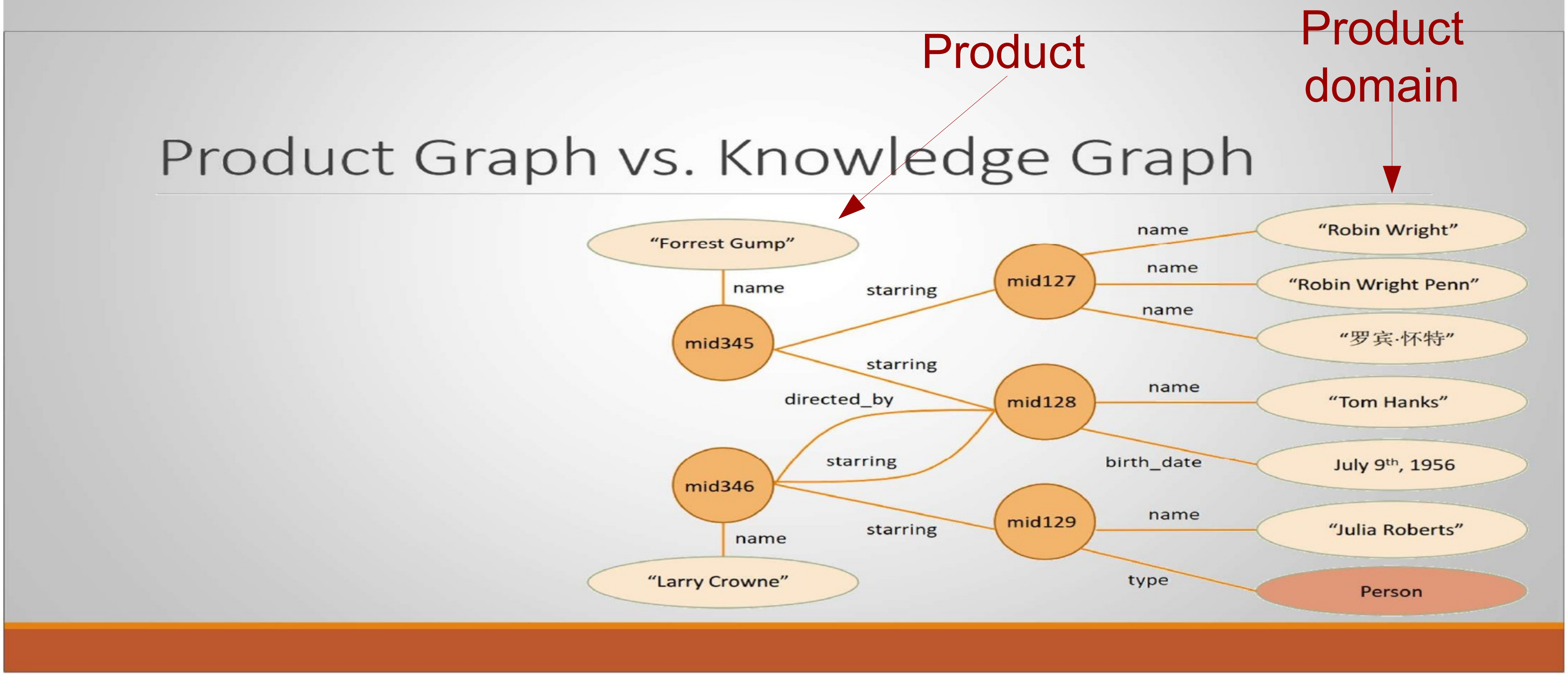
Amazon

Product Graph vs. Knowledge Graph



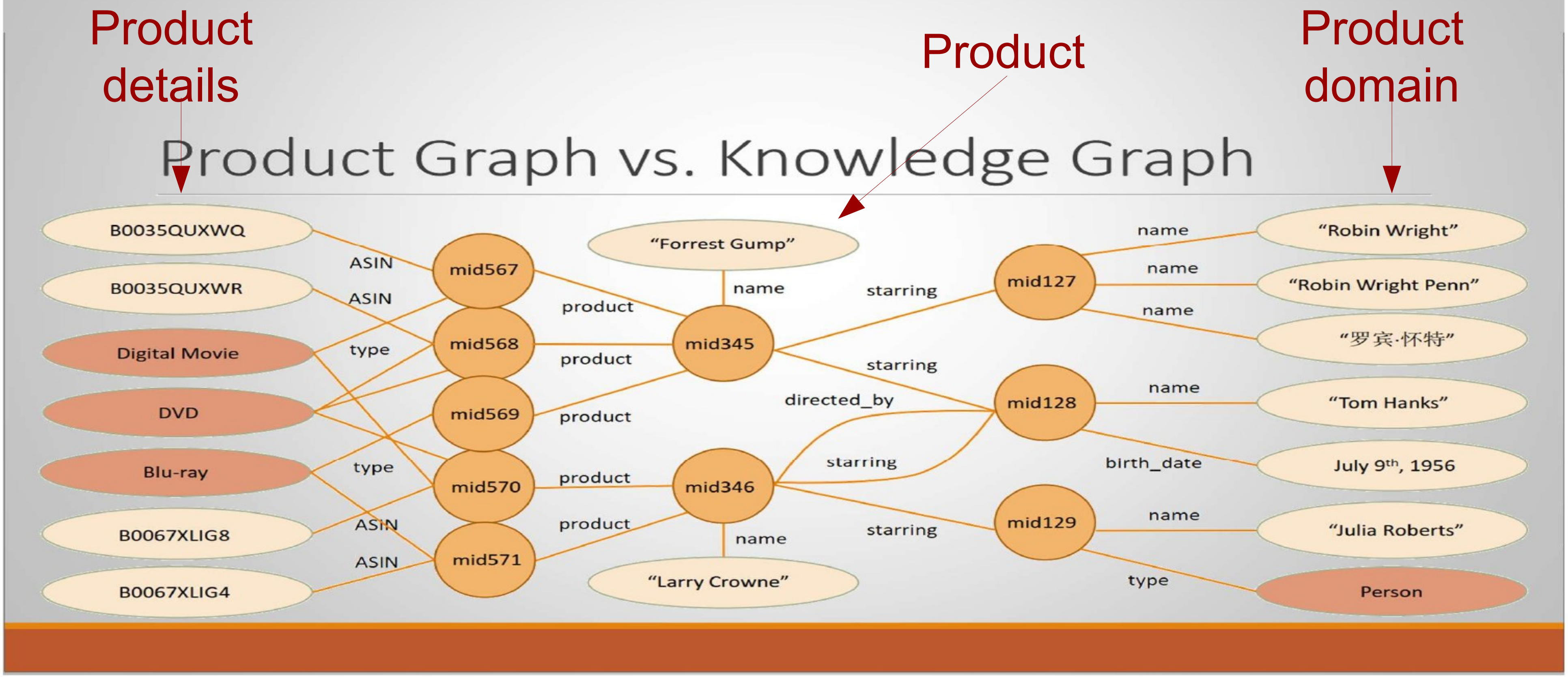
Frank van Harmelen (2018): Keynote at CAiSE'18

Amazon



Frank van Harmelen (2018): Keynote at CAiSE'18

Amazon



Frank van Harmelen (2018): Keynote at CAiSE'18

Ratings & reviews

Amazon

Delivery services

Customers

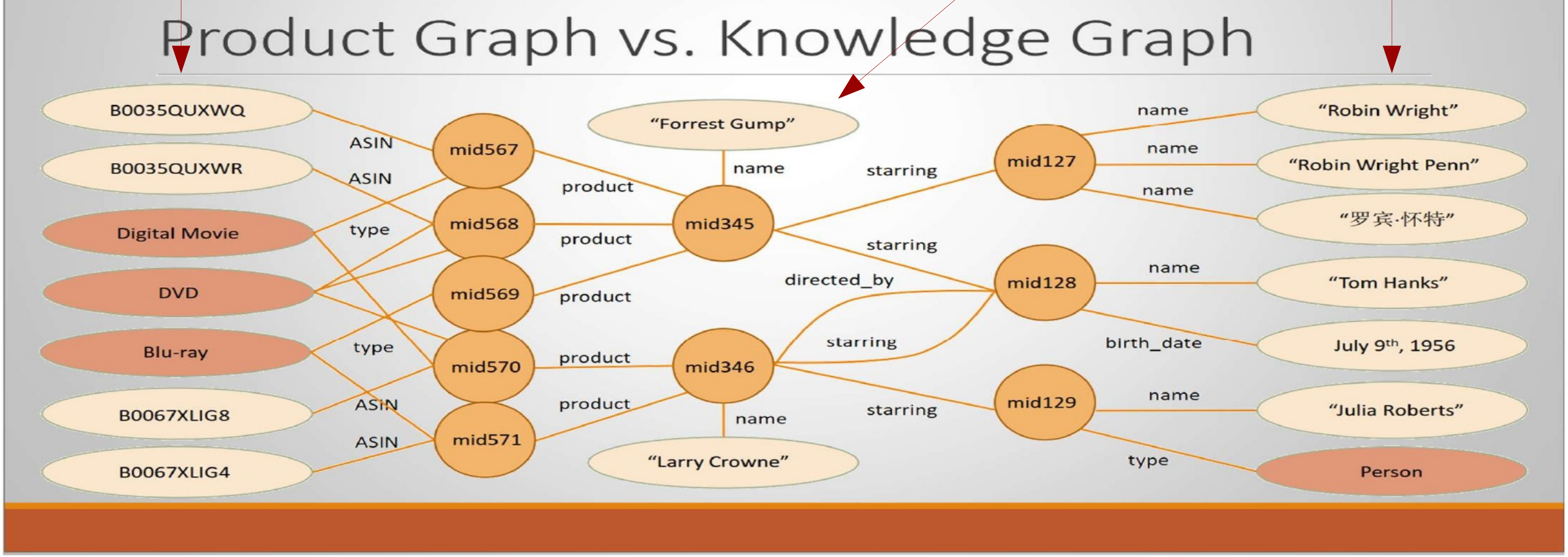
Product details

Suppliers

Support

Product

Product domain

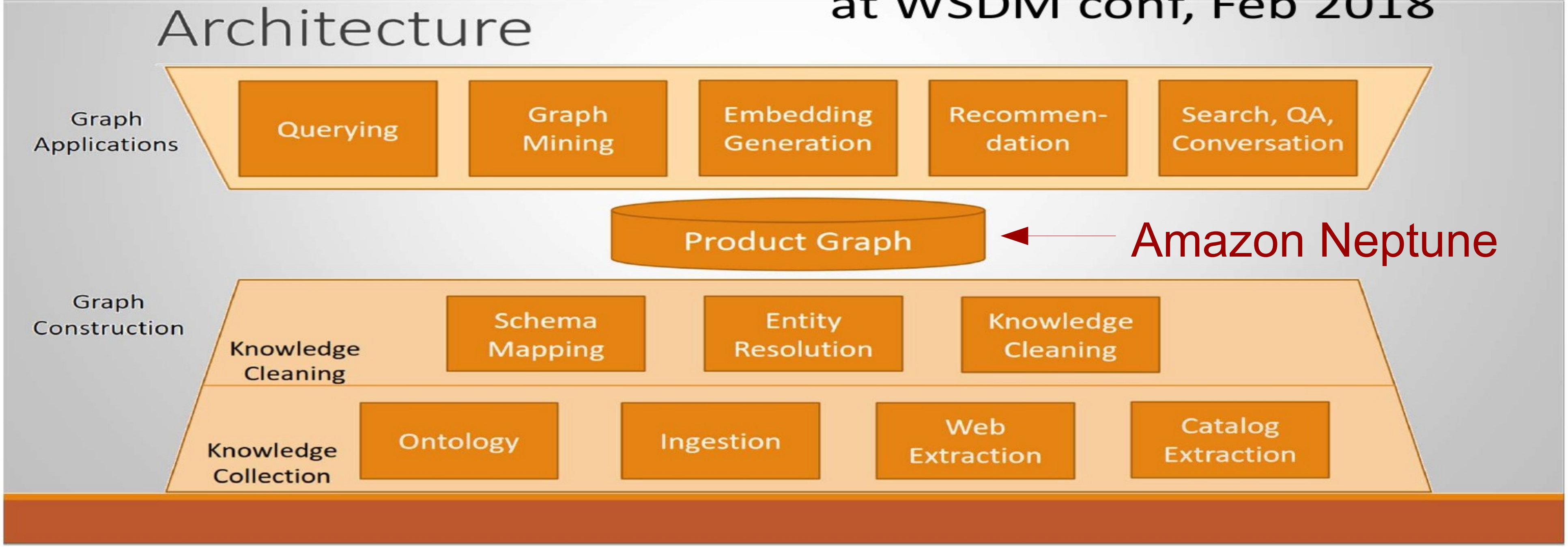


Frank van Harmelen (2018): Keynote at CAiSE'18

Amazon

"We aim at building an authoritative knowledge graph for all products in the world"

Xin Luna Dong, Amazon, at WSDM conf, Feb 2018



Challenges

- Ingest product-related information from Amazon’s detail pages and from the Internet at large
  - product information is largely unstructured
  - trustworthiness of sources
- Machine learning techniques for
  - knowledge extraction, linkage and cleaning
  - distantly supervised learning (distant supervision)
    - use existing structured data to generate weak training data
    - train model on text data
  - open information extraction
  - graph mining techniques to identify interesting hidden patterns (buying product-X → buying product-Y)

Amazon AutoKnow

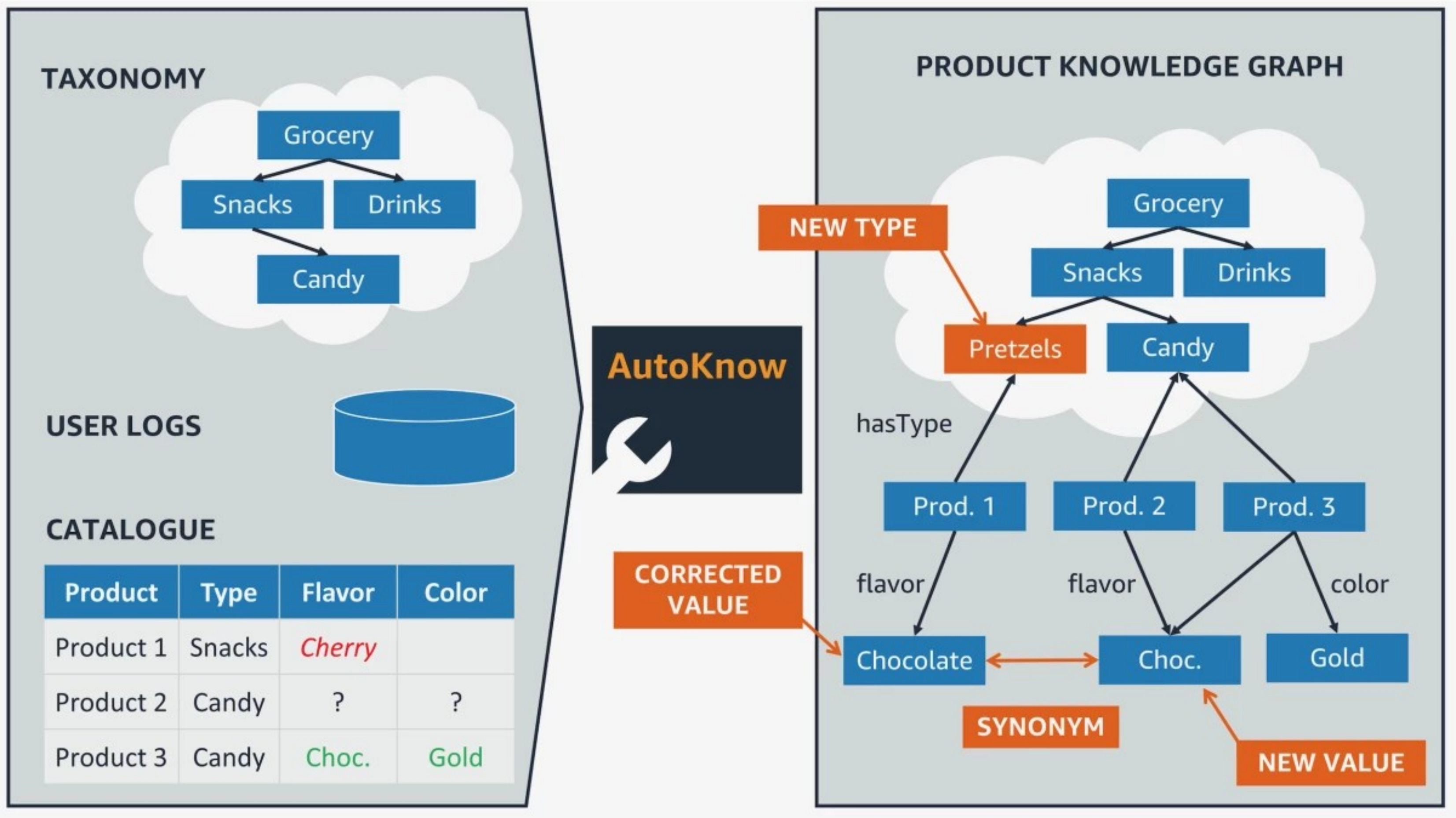
How to build a Product KG?

- Amazon's AutoKnow:
  - a suite of techniques for automatically augmenting product KGs with both structured data and data extracted from free-form text sources
- Tasks:
  - combining data from different sources into a product graph
  - adding new product types to the taxonomy
  - adding new values for product attributes
  - correcting errors
  - identifying synonyms
- *"With AutoKnow, we increased the number of facts in Amazon's consumables product graph (which includes the categories grocery, beauty, baby, and health) by almost 200%, identifying product types with 87.7% accuracy."*

Challenges

- Retail information is hard:
  - the number of product types tends to grow as the graph expands
  - each product type has its own set of attributes
  - attributes vary widely, e.g.,
    color and texture versus battery type and effective range
  - the types of relationships between data items are essentially unbounded
  - vital product information exists in free-form text, e.g.,
    user reviews or question-and-answer sections

AutoKnow architecture



AutoKnow architecture

- Inputs:
  - an existing product taxonomy
    - a graph structure
  - a product catalogue
    - structured information, such as labelled product names
    - unstructured product descriptions
  - user logs
    - free-form textual product-related information: customer reviews, product-related questions and answers; and product query data
- Output:
  - Amazon's product graph

AutoKnow architecture

- Five modules in two suites:
  - Ontology suite
    1) taxonomy enrichment: identify and classify new entity types
    2) relation discovery: identifies (1) attributes of products, (2) their range of possible values, and (3) their importance to customers
  - Data suite
    3) data imputation: uses the entity types and relations to determine whether free-form text associated with products contains any information missing from the graph
    4) data cleaning: sorts through existing and newly extracted data to see whether any of it was misclassified
    5) synonym finding: identifies entity types and attribute values with identical/similar meaning

Taxonomy enrichment module

- Identification of new product types:
  - ML model labels substrings of product titles in the source catalogue.
    - also labels substrings that indicate product attributes
    - for use during the relation discovery step.
  - trained on product descriptions with hand-labelled types and attributes
- Classification of product types according to their hypernyms (i.e., the broader product categories that they fall under):
  - ML classifier uses data about customer interactions, such as which products customers viewed or purchased after a single query
  - trained on product data hand-labelled according to an existing taxonomy

Relation discovery module

- Classification of product attributes by two criteria and ML classifiers:
  - whether the attribute applies to a given product
    - example: flavour (an attribute) applies to food but not to clothes
  - how important the attribute is to buyers of a particular product
    - example: brand name (an attribute) is more important to buyers of snack foods than to buyers of produce
- Input data:
  - product descriptions from providers
    (attribute frequencies per product and per product type)
  - reviews and Q&As from customers
    (attribute frequencies per product)
  - manually-labelled data that match attributes with products

Data imputation module

- Identification of terms in product descriptions
  - that may fit the new product and attribute categories
  - but which are not yet represented in the KG
  - the product type is included among the inputs
- *Word embeddings* represent descriptive terms as points in a *vector space*
  - the vector space is trained to group together related terms
  - some terms are already labelled with a product type or attribute they represent
  - if many labelled terms in the same cluster share the same label, then the unlabelled terms in the same cluster have those labels too

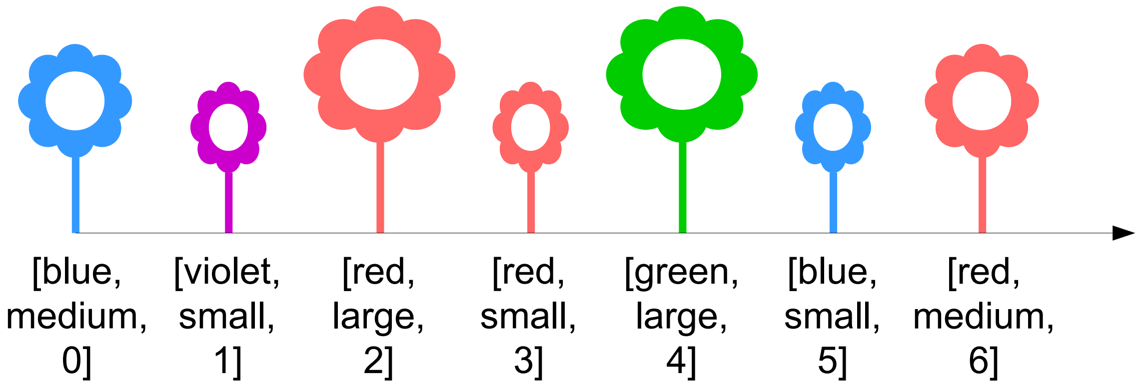
How can we represent the meaning of words?

- By designation (e.g., textual descriptions in a dictionary)
- As nodes in a network (e.g., in a knowledge graph in in WordNet)
- Formally (e.g., using an OWL ontology)
- As *vectors* in a *latent semantic space*!

How can we represent the meaning of words?

- By designation (e.g., textual descriptions in a dictionary)
- As nodes in a network (e.g., in a knowledge graph in in WordNet)
- Formally (e.g., using an OWL ontology)
- As *vectors* in a *latent semantic space*!
- Example:
  - *FlowerWorld™*
  - *“Everything is a flower!”*
  - *a flower has three attributes:*
    - *colour*
    - *size*
    - *position*

Everything in FlowerWorld™ can be uniquely described by its position along three dimensions!

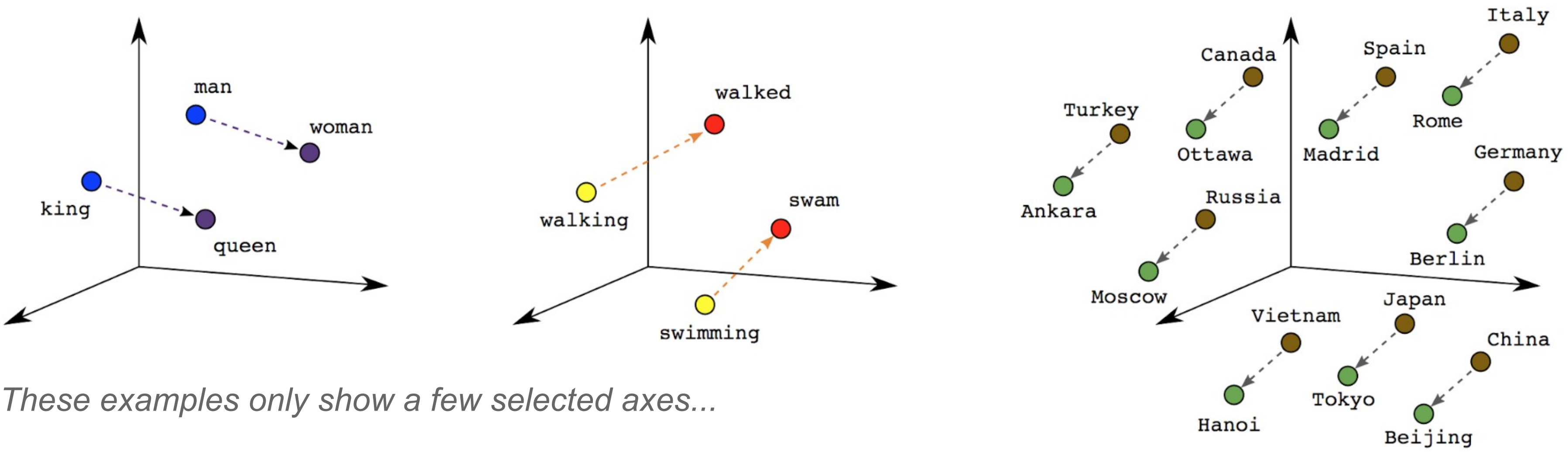


How can we represent the meaning of words?

- By designation (e.g., textual descriptions in a dictionary)
- As nodes in a network (e.g., in a knowledge graph in in WordNet)
- Formally (e.g., using an OWL ontology)
- As *vectors* in a *latent semantic space*!
- (Our conceptualisations of) Things in the “real world”:
  - a bit more complex...
  - not fully describable by positions along dimensions
  - but perhaps we can describe them usefully by adding more dimensions?
  - but which dimensions to add?
    - use machine learning / neural networks to analyse large text corpora!

How can we represent the meaning of words?

- By designation (e.g., textual descriptions in a dictionary)
- As nodes in a network (e.g., in a knowledge graph in in WordNet)
- Formally (e.g., using an OWL ontology)
- As *vectors* in a *latent semantic space*!



These examples only show a few selected axes...

Male-Female

Verb Tense

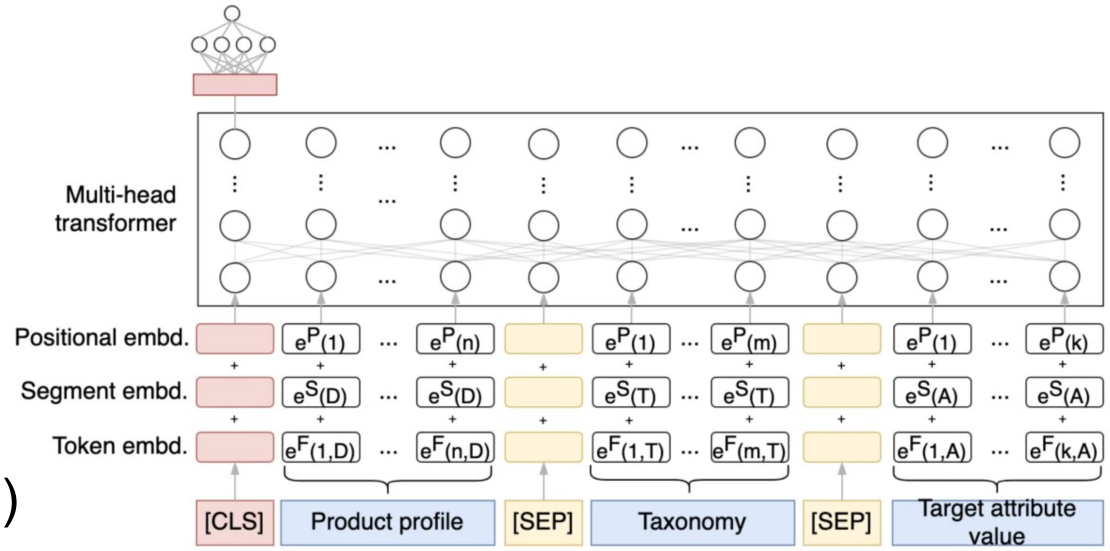
Country-Capital

How can we represent the meaning of words?

- By designation (e.g., textual descriptions in a dictionary)
- As nodes in a network (e.g., in a knowledge graph in in WordNet)
- Formally (e.g., using an OWL ontology)
- As *vectors* in a *latent semantic space*!
  - normalised values: [0.01 0.62 0.03 … 0.41]
  - important use: as inputs to deep neural networks that process NL text
  - trained, e.g., so that similar words are close to one another
  - ...so that position differences between words can be systematic
    - [Paris] – [France] + [Italy] ≈ [Rome]
  - ...so that position differences between words can represent relations
    - [J. K. Rowling] + [influenced by] ≈ [J. R. R. Tolkien]

Data cleaning module

- Detecting bad attribute values
  - using a *transformer model*
  - inputs:
    - NL product description
    - an attribute (e.g., flavour...)
    - an attribute value (e.g., vanilla...)
  - is the attribute-value pair aligned with the product?



- Trained on
  - positive examples: valid attribute-value pairs that occur across many instances of the product type (e.g., all ice cream types have flavours)
  - negative examples: generated by random replacement of values in valid attribute-value pairs

Synonym finding module

- Analysis of product and attribute sets to find mergeable KG nodes
  - customer interaction data to identify items that were viewed during the same queries
    - their product and attribute descriptions are candidate synonyms
  - a combination of techniques to filter the candidate terms
    - edit distance
    - neural network

Ongoing work

- Open questions:
  - how to handle products with multiple hypernyms
    (i.e., products that have multiple “parents” in the product hierarchy)?
  - how to clean data before it’s used to train our models?
  - how to use image data + textual data to improve model performance

Next week:
Rules (SHACL and RDFS)