# Mining Meaning From Wikipedia

PD Dr. Günter Neumann

LT-lab, DFKI, Saarbrücken

# Outline

1. Seed Elements for Relation Extraction
2. Ontology Building from WikiPedia - Introduction

# How to exploit Wikipedia for Relation Extraction ?

- Note that each article represents a particular concept that serves as a clearly recognizable principal entity for relation extraction from that article.
- Its description contains links that point to other articles/entities.
- It only remains to identify the possible relations between these pairs of entities.
- Using/learning syntactic and lexical patterns as textual anchors.

# How to get seed elements?

- Using WordNet as a source of positive seeds
- Automatically identified in Wikipedia articles
- Automatically extracted from infoboxes

# WordNet for identifying seeds, cf. Ruiz-Casado et al., 2007

- Idee: Given two co-occurring semantically related WordNet nouns in a Wikipedia article, the intervening text is used to find relations that are absent from WordNet.
- Use WordNet synsets as a sense basis:
  - For each entry, retrieve synsets
  - If multiple senses exist, then disambiguate entry by means of entry's definition & each sense's WordNet gloss
  - Select WordNet sense with maximum similarity according to vector space model
- Result:
  - ~83,89% accurcacy on Simple Wikipedia test set
  - 1/3 of Simple Wikipedia are polysemious (1/3 are unique, 1/3 are not covered by WordNet)

# Pattern extraction for this approach

- Determine the two arguments in the definition:
  - 1st argument is clear: the entry e itself
  - 2nd argument: a linked term f, s.t. f&t have a common relation in WordNet
- Pattern extraction:
  - Substitute e with ENTRY, f with TARGET
  - Follow approach like in Snowball
- Result:
  - 1200 new relations for WordNet with 61-69% precision

# Wikipedia articles as seed basis, cf. Ruiz-Calados et al., 2006

- Starting point:
  - a list of interesting relationships, e.g., Person's birth year, Person's death year, Person-birth place, Actor-film, Writer-book, Football player-team, Country-chief of state, Country-capital

- Automatic crawling of Wikipedia articles, that contain these relations:
  - Prime Minister, that contains hyperlinks to Prime Ministers from many countries.
  - Lists of authors, that contain hyperlinks to several lists of writers according to various organising criteria.
  - Lists of actors, that contain hyperlinks to several lists of actors.
  - List of football (soccer) players, containing hyperlinks to many entries about players.
  - List of national capitals, containing the names of national capitals from countries in the world.
- Pattern extraction process like the previous one

# Result for this approach

- Depending in specific relation

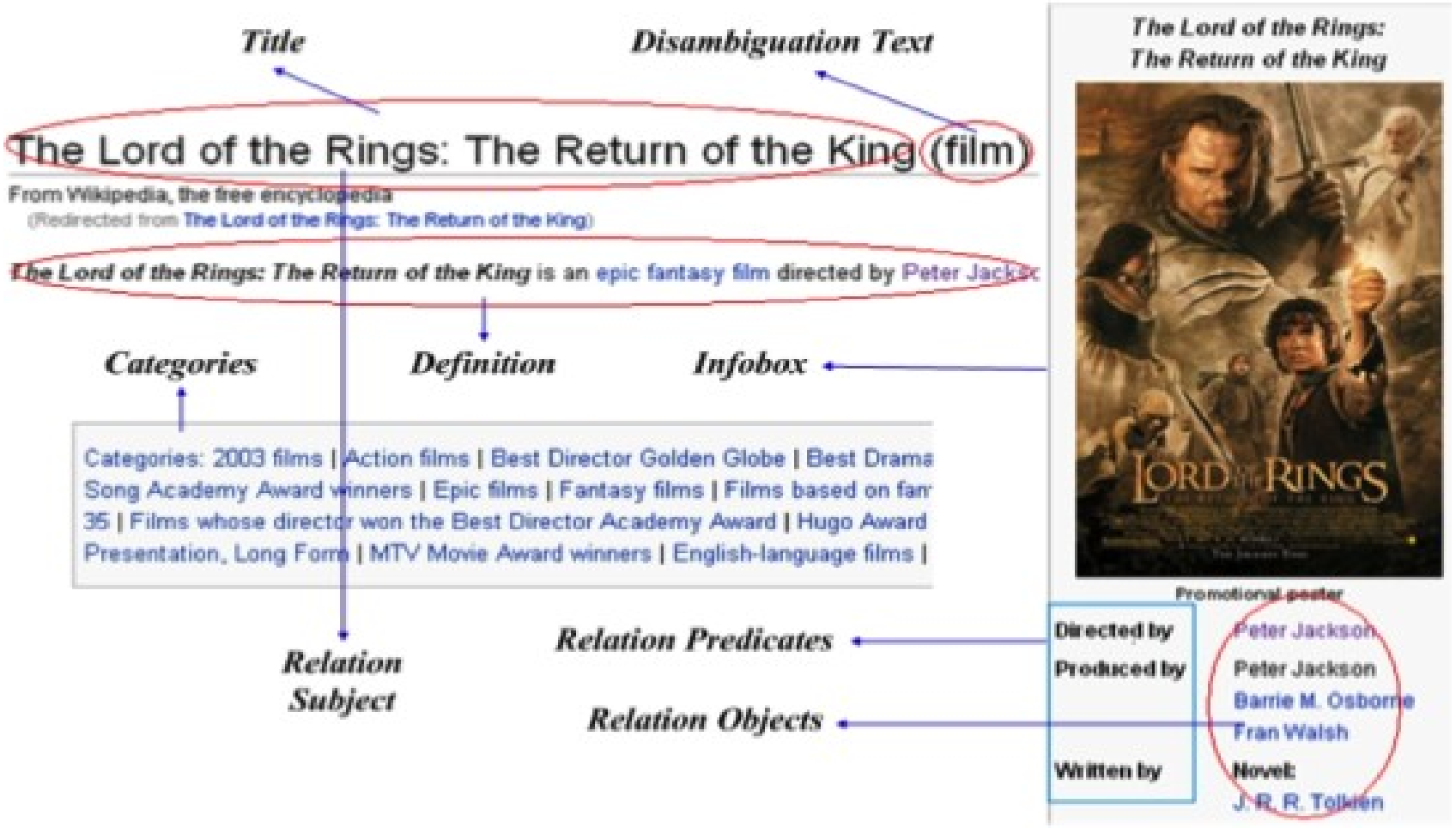
| Relation | No. of patterns | No. of results | Precision |
|---|---|---|---|
| Birth-year | 16 | 15746 | 74.14% |
| Death-year | 8 | 5660 | 90.20% |
| Birth-place | 3 | 154 | 27.27% |
| Actor-film | 11 | 4 | 50.00% |
| Country-Chief of state | 109 | 272 | 50.00% |
| Writer-book | 176 | 179 | 37.29% |
| Country-capital | 150 | 825 | 11.45% |
| Player-team | 173 | 315 | 7.75% |

**Table 2.** Number of patterns obtained for each relationship, number of results extracted by each pattern set, and precision.

# Exploring dependency parsing, cf. Nguyen et al. 2007

- Approach:
  - Select Wikipedia article
  - Perform anaphora resolution for article title (determines 1st argument)
  - Tag all link elements as 2nd argument
  - For all sentences containing 1st & 2nd arguments, do dependency parsing and generalized to match similar sentence (subtrees are substituted by variables)
- Result:
  - F-measuer of 38% on 45 Wikipedia test articles

# Seeds from Infoboxes, cf. Wang et al. 2007



# Automatically extracted from infoboxes

- Select relation seed instances randomly selected from infoboxes of Wikipedia.
  - Example
    - hasDirector(film, director)
    - <Titanic, James_Cameron>
    - <King_Kong (2005), Peter_Jackson>
  - The seeds are used to query the wikipedia corpus.
- From returned text snippets, patterns are created
  - From 'Titanic was a romantic film directed by James Cameron' and 'King Kong (2005) is an American movie directed by Peter Jackson'
  - Create pattern: '* (is|was) (a|an) * (film|movie) directed by *'.

# Semanticfy extracted pattern

- Determine types for the pattern variables using information from Wikipedia articles
  - Y must be a director or at least a person
  - Type labels are automatically determined from words commonly co-occuring in Wikipedia articles describing these entities
    - Artist → singer, musician, guitarist, rapper
  - Advantage is, that much more selectional restriction patterns can be inferred
- Result: for 100 relations nearly 100 % accuracy

# System

Self-supervised learning -> autonomous
Form training dataset based on infoboxes
Extract semantic relations from Wikipedia articles

## From infoboxes to a training set

| Clearfield County, Pennsylvania | |
|---|---|
| **Statistics** | |
| **Founded** | March 26, 1804 |
| **Seat** | Clearfield |
| **Area** | |
| - **Total** | 2,988 km² (1,154 mi²) |
| - **Land** | sq mi ( km²) |
| - **Water** | 17 km² (6 mi²), 0.56% |
| **Population** | |
| - **(2000)** | 83,382 |
| - **Density** | 28/km² |

Clearfield County was created in 1804 from parts of Huntingdon and Lycoming Counties but was administered as part of Centre County until 1812.

Its county seat is Clearfield.

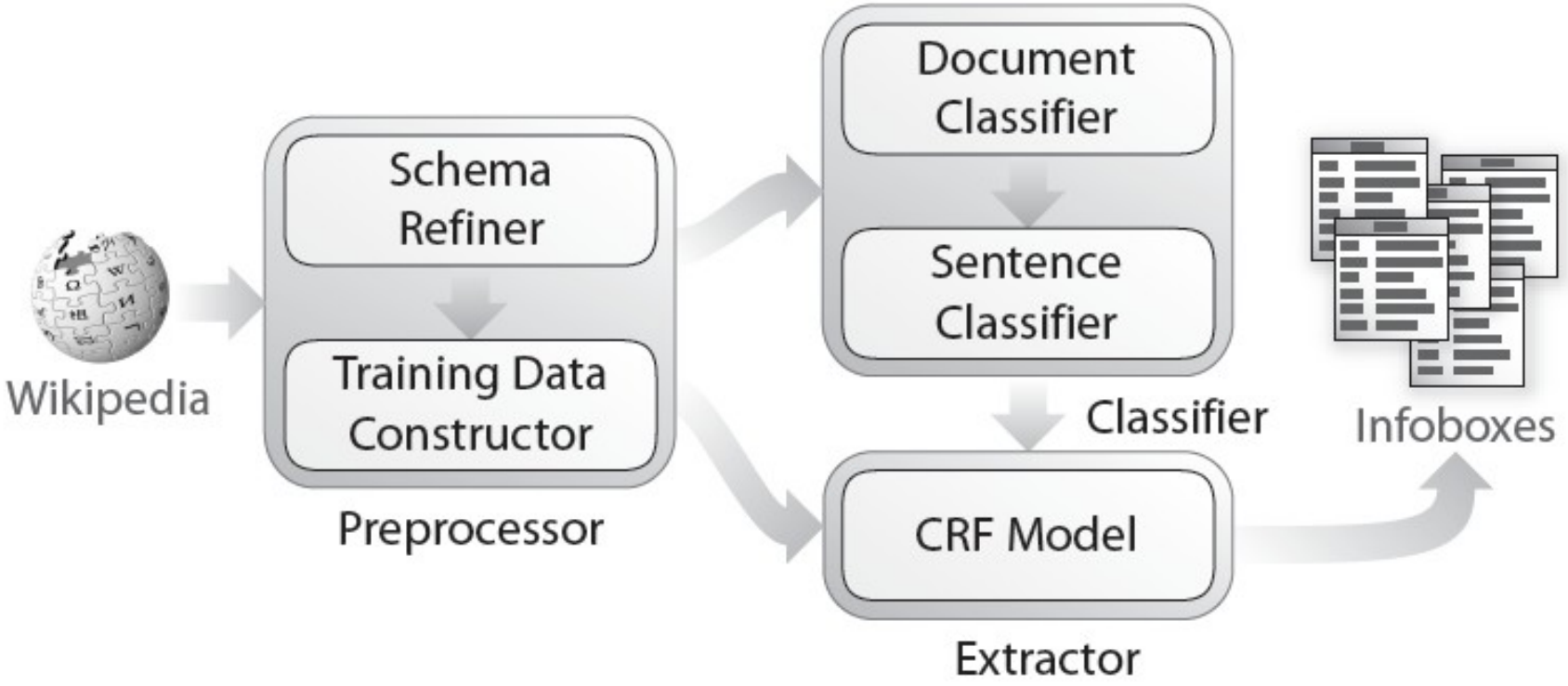
2,972 km² (1,147 mi²) of it is land and 17 km² (7 mi²) of it (0.56%) is water.

As of 2005, the population density was 28.2/km².

# Some more details concerning creation of training data

- For each article with an infobox mentioning one or more target attributes, Kylin segments the document using the OpenNLP library.
- Then, for each target attribute, Kylin tries to find a unique, corresponding sentence in the article.
- The resulting labelled sentences form positive training examples for each attribute.
- Other sentences form the negative training examples.

# Kylin Architecture

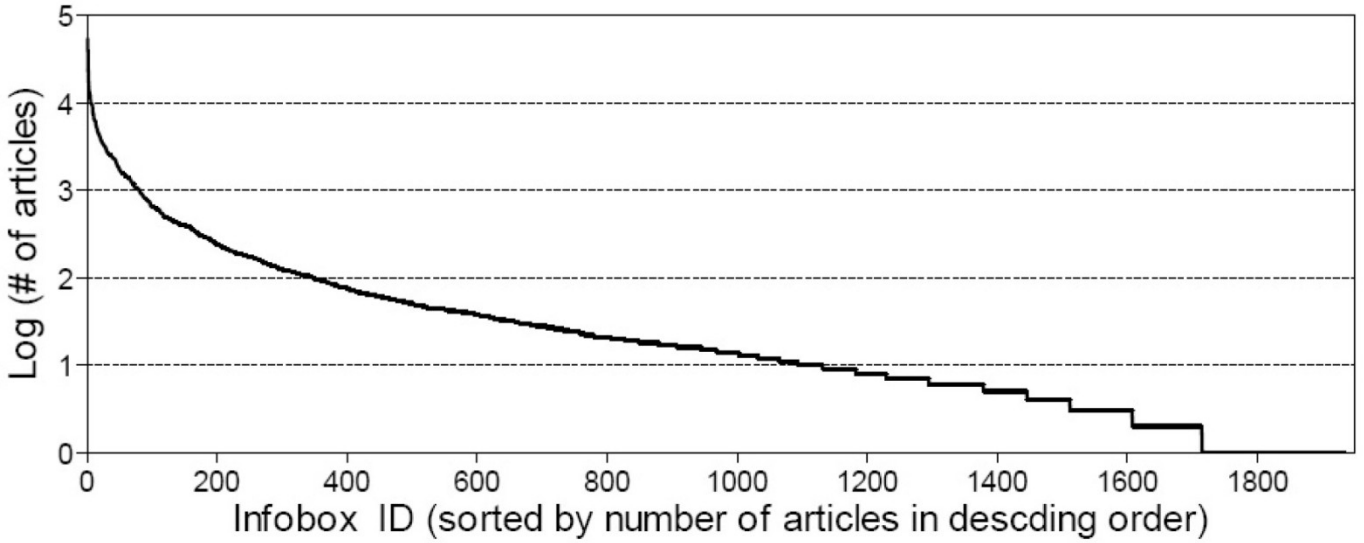


# Preliminary Evaluation

- Kylin Performed Well on Popular Classes (e.g., articles belonging to Lists or Categories):

  Precision: mid 70% ~ high 90%

  Recall: low 50% ~ mid 90%
- ... Floundered on Sparse Classes – Little Training Data




*82% < 100 instances; 40% <10 instances*

# Semantic Relations in Structured Parts of Wikipedia

- Goal is to build alternative knowledge bases for manually defined KB's such as WordNet and Cyc
- Approaches
  - Label existing links between categories and articles
  - Extract relations from infoboxes

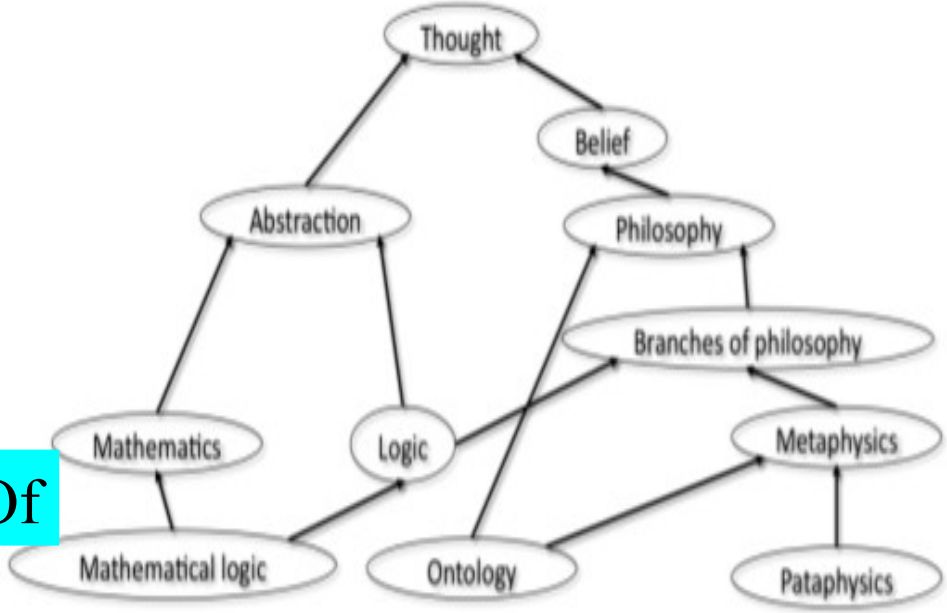



Figure 13. Fragment of Wikipedia's category structure [Ponzetto, 2007].

# Three prominent approaches

- Yago (MPI, Saarbrücken)
- DBPedia (Uni. Leipzig, FU Berlin)
- EMLR (EML Research, Heidelberg)

# Yago – Yet Another Great Ontology

- Wikipedia's leaf categories are mapped onto the WordNet taxonomy of synsets
  - the articles belonging to those categories are added to the taxonomy as new elements.
- Mapping:
  - Extract lexical head from category name, e.g., Category:American people in Japan
  - Apply WSD in case of polysemous senses
- 143,000 isA relations
  - e.g., isA(American people in Japan, person/human)

# Yago – Extracting further relations

- Programmed heuristics
  - NE parser for identifying given and family names
    - 440,000 relations of form familyNameOf(Albert Einstein, "Einstein")
  - „parsing" category names
    - bornInYear, establishedIn, locatedIn
    - e.g., based on suffix analysis of category name
  - Gives 370,000 non-hierarchical, non-synomyous relations (91%-99% accuracy)
- Further relations from other Wikipedia structures
  - 2M synonymy relations from redirect links
  - 40M context relations from cross-links between articles
  - 2M type relations (categories considered as class, articles as entities)

Next slides about Yago are from Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum, MPI, SB

# The Truth about Elvis

Elvis is alive!

This slide and the next about Yago are from Suchanel, 2009.

# The Truth about Elvis

Elvis is alive!

He works as an astronaut in

NASA's special security program

# Usual solution

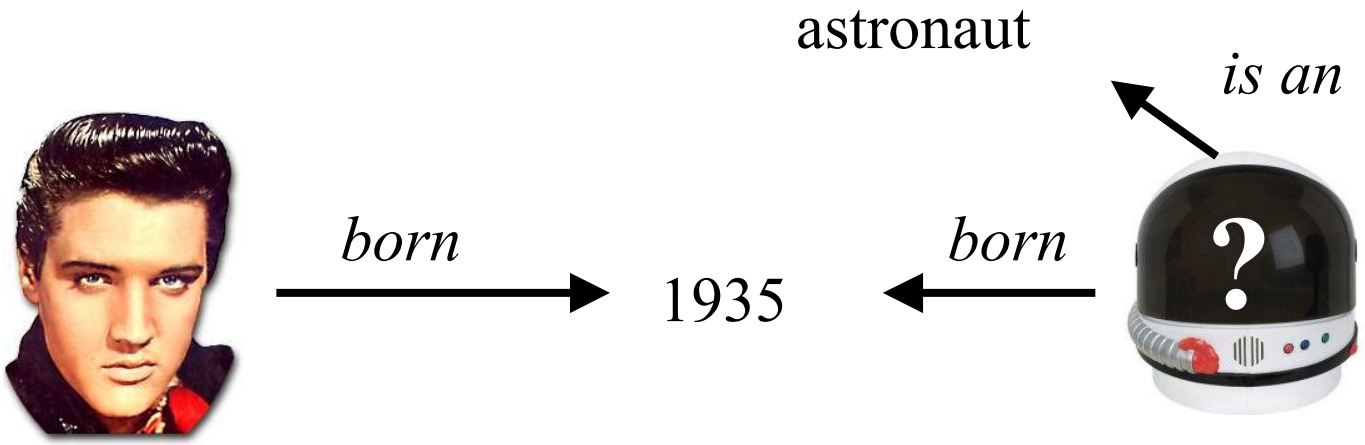

*Which NASA astronaut was born when Elvis was born?*

Yields only rubbish.

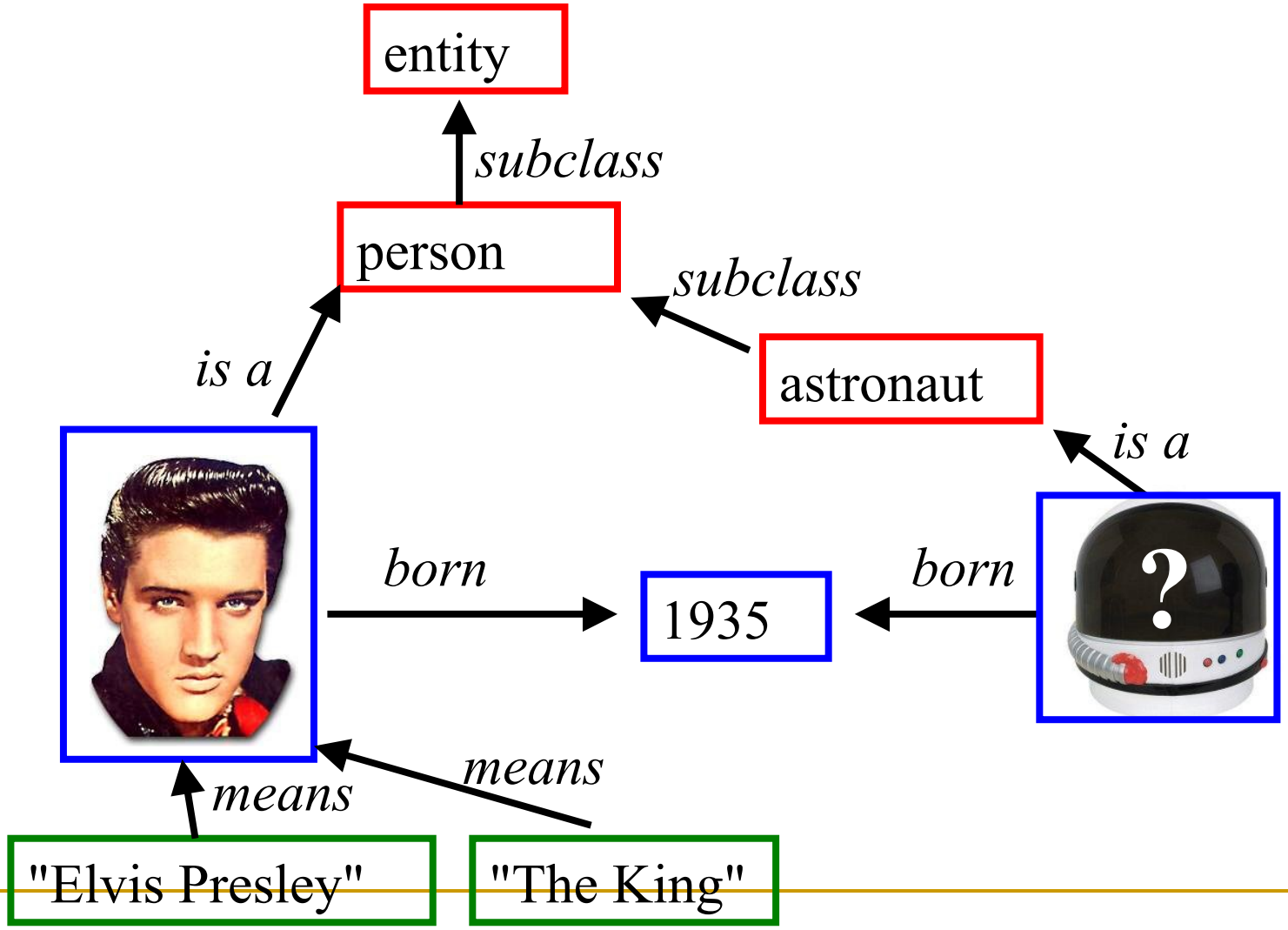
Reasons:

1. Google participates in the conspiracy
2. Google does not search knowledge, but Web sites

# Solution: An ontology



# Solution: An ontology



# Solution: An ontology