

# Knowledge Graph Construction

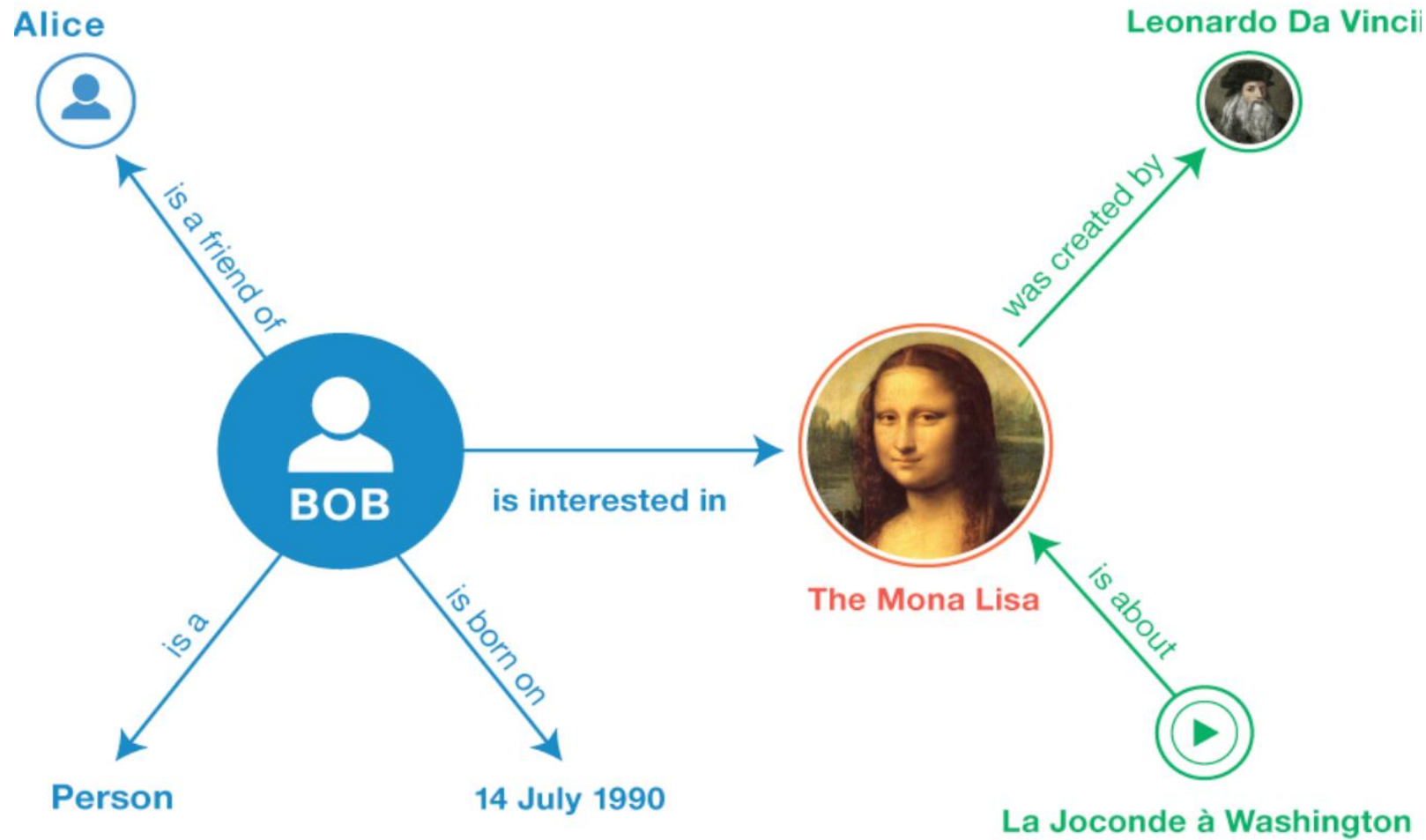
Slide courtesy: Jay Pujara, Luke Zettlemoyer

# Lecture Outline

- Automatic Knowledge Graph Construction from unstructured data
  - NLP techniques for KG construction

Recap

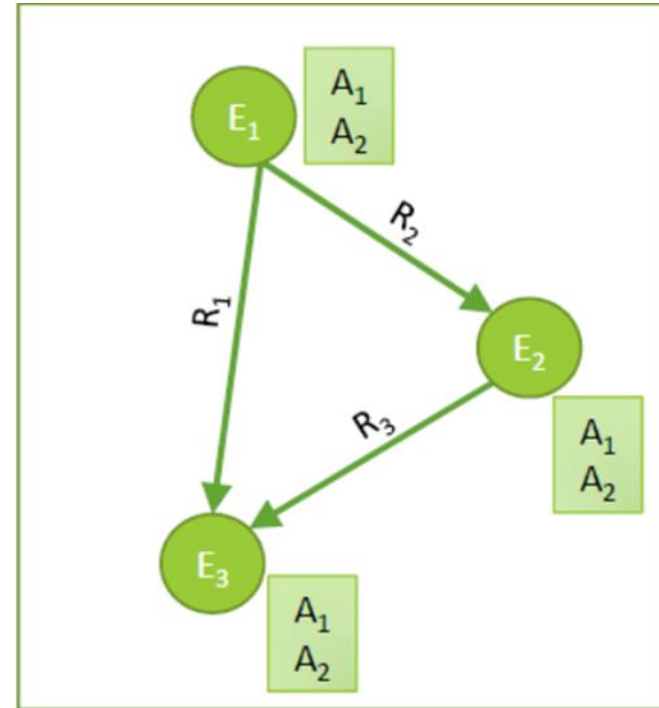
# Structure of Knowledge Graph



Credit: Peter Haase, Getting Started with Knowledge Graphs

# Key Elements

- Entities
- Relations between entities
- Attributes/properties



# Extracting Candidates: Entities and Relationships

# What is NLP?



Unstructured  
Ambiguous  
Lots and lots of it!

Humans can read them, but  
... very slowly  
... can't remember all  
... can't answer questions

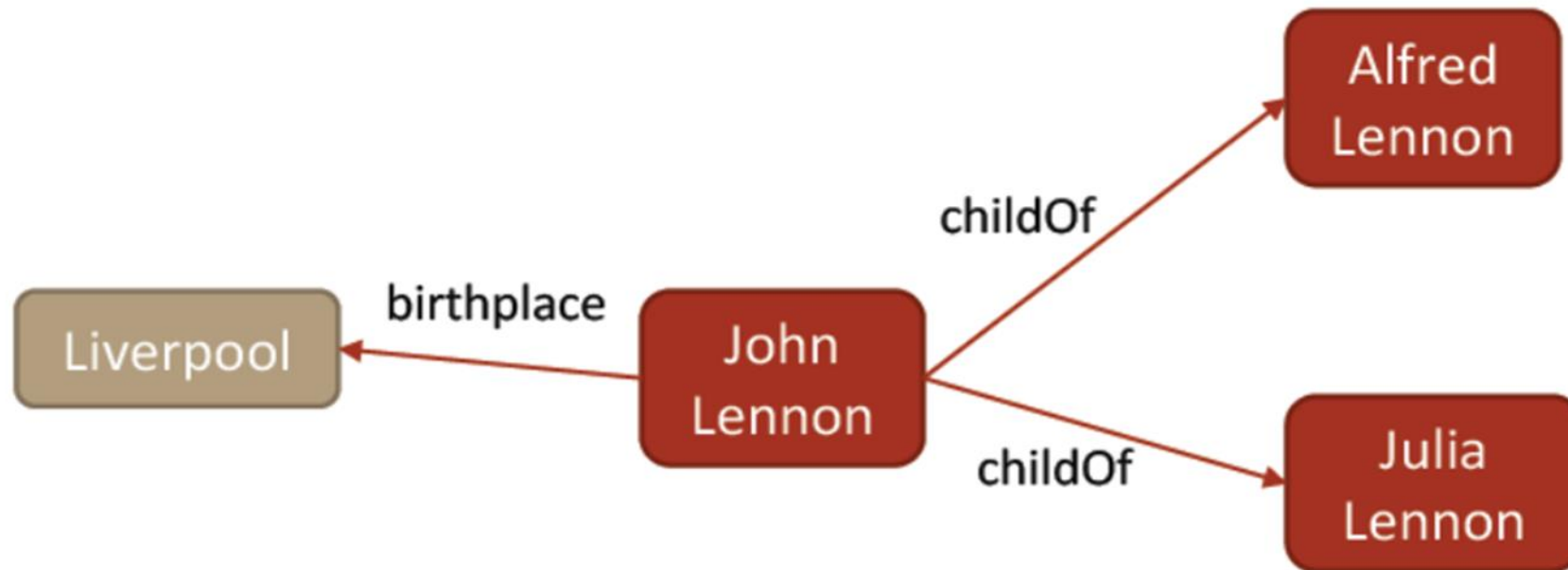


Structured  
Precise, Actionable  
Specific to the task

Can be used for downstream  
applications, such as creating  
Knowledge Graphs!

# Example: Knowledge Extraction for KG

**John Lennon was born in Liverpool, to Julia and Alfred Lennon**



# Main Components

**John Lennon** was born in **Liverpool**, to **Julia** and **Alfred Lennon**

Entities	Relationship	From entity	To entity	Entity	Property name	Value
John Lennon	ChildOf	John Lennon	Julia	John Lennon	birthplace	Liverpool
Liverpool	ChildOf	John Lennon	Alfred Lennon			
Julia	Alfred Lennon					
Alfred Lennon						

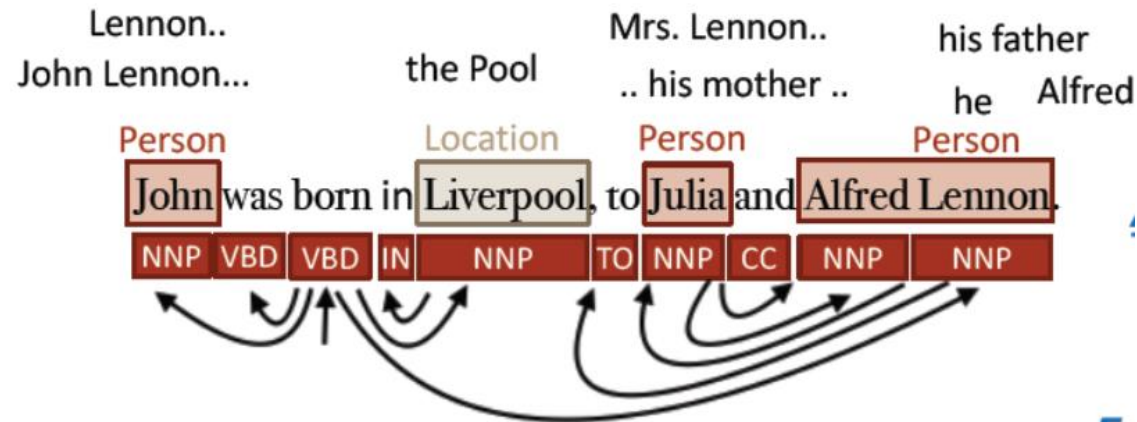
# General Pipeline

Natural language text

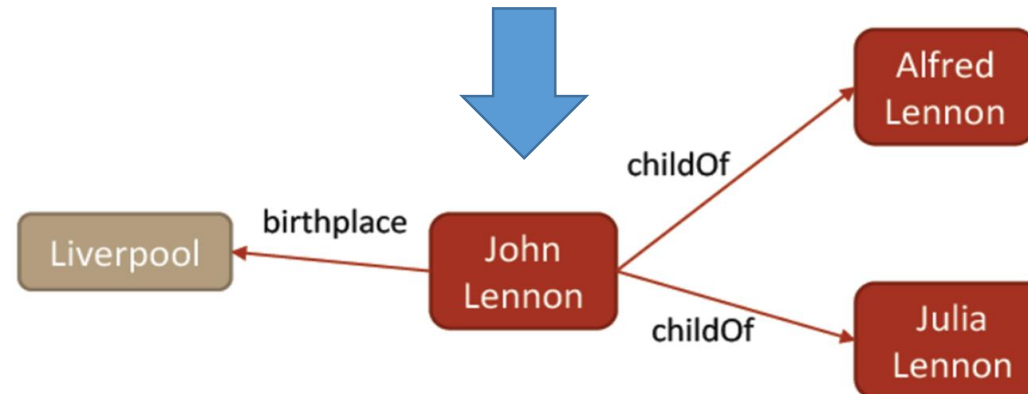
John was born in Liverpool, to Julia and Alfred Lennon



Annotated text



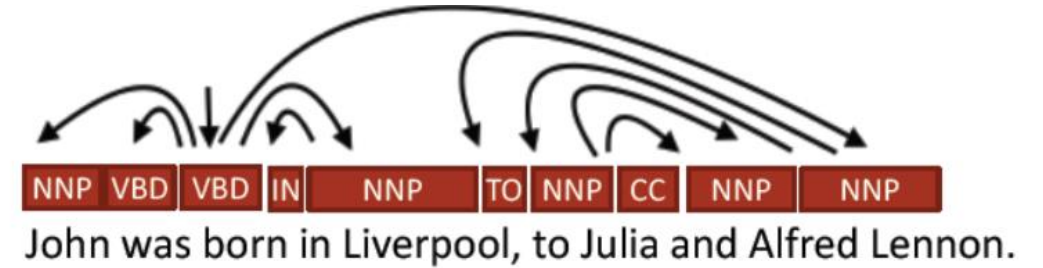
Extracted Graph



# Tools and Techniques

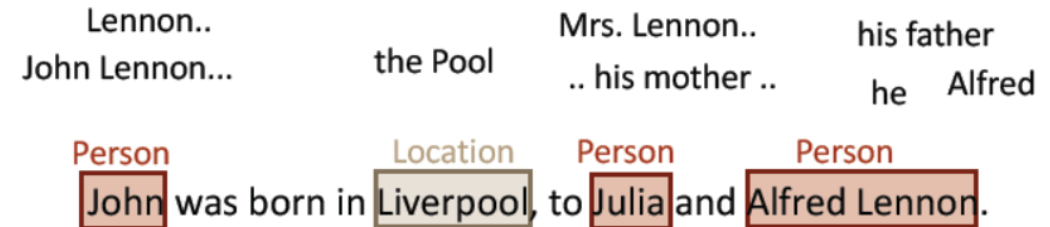
NLP  
Sentence level

Dependency Parsing,  
Part of speech tagging,  
Named entity recognition...



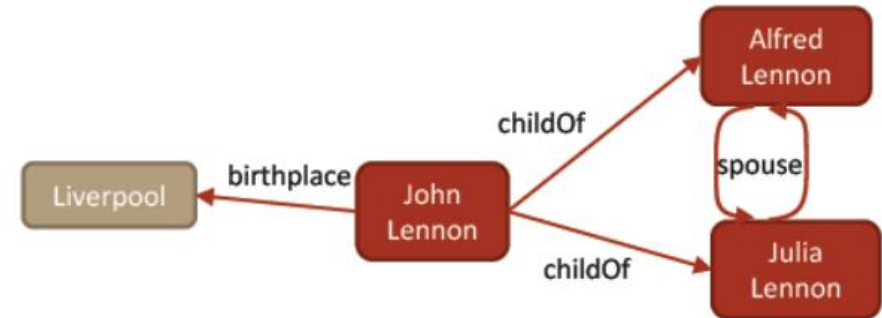
NLP  
Document/passage  
level

Coreference Resolution...



Information  
Extraction

Entity resolution,  
Entity linking,  
Relation extraction...



# Parts of Speech Tagging

- How does it help in KG?
  - Usually Entities are noun
  - Verbs help to determine relationship/properties of entities
- What is Parts of Speech Tagging?

**John was born in Liverpool, to Julia and Alfred Lennon.**

NNP	VBD	VBD	IN	NNP	TO	NNP	CC	NNP	NNP
-----	-----	-----	----	-----	----	-----	----	-----	-----

Usually statistical sequence models are used on a large manually labelled data to learn tagging rules.

# Identifying named entities

- Noun/noun phrases that represent some name (person, location, organization etc)
- Names are always entities in KG
- Also captures entity types
- Example:

Person                      Location                      Person                      Person  
John was born in Liverpool, to Julia and Alfred Lennon.

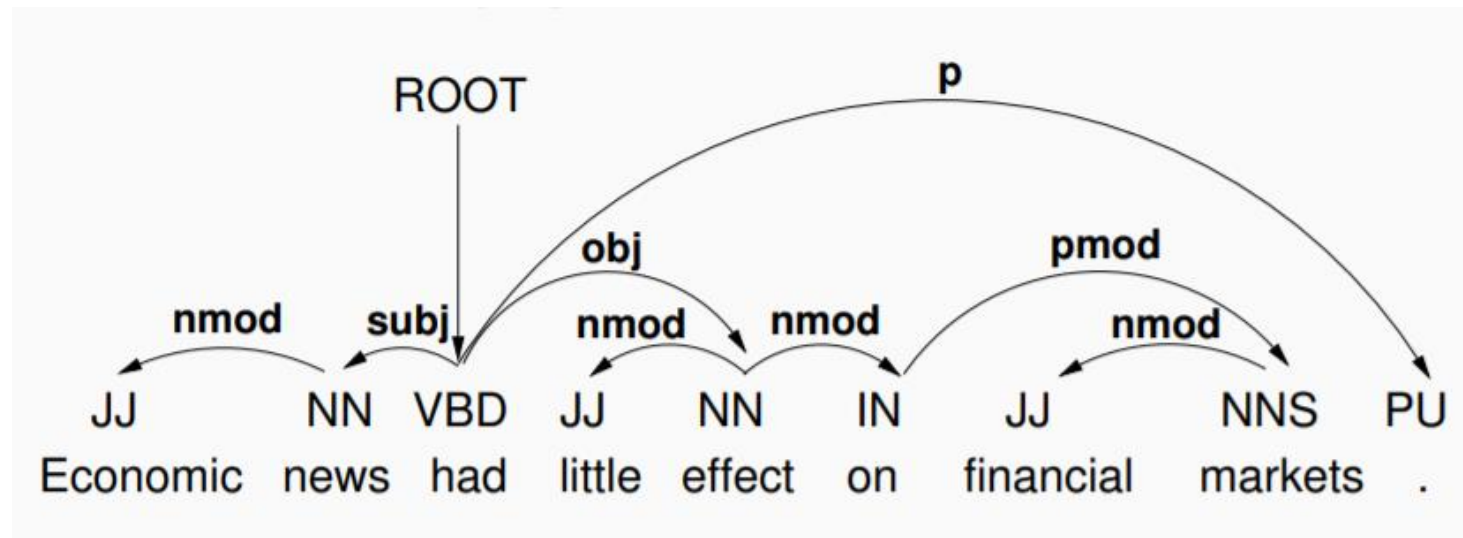
# Dependency Parsing

- So far
  - Word/phrase level tagging, but no information about the relationship between words/phrases
- We need to know how the words/phrases are related
- Dependency parsing:
  - Analyses the grammatical structure of a sentence,
  - Establishes relationships between "head" words and words which modify those heads.

# Dependency Parsing

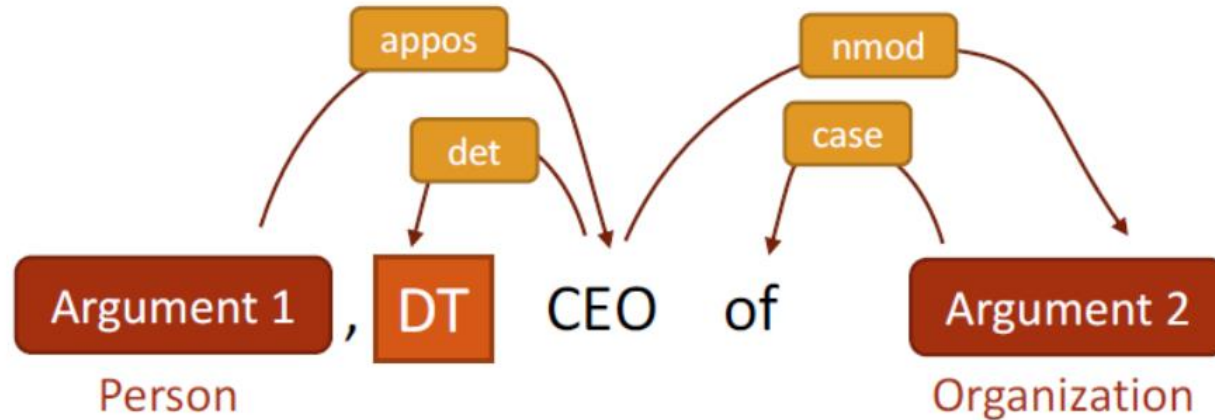
- dependency relations are binary and asymmetric.
- A relation consists of:
  - A head (H)
  - A dependent (D)
  - A label denoting the relation between H and D

- Example:

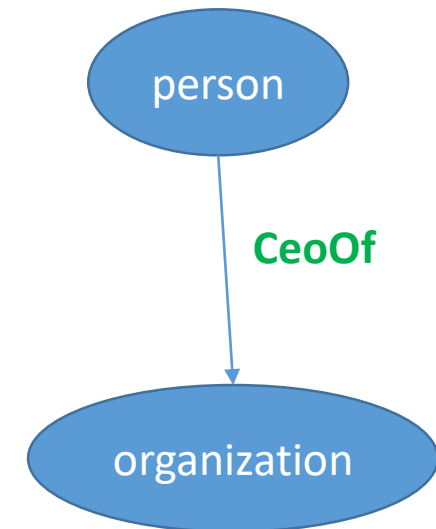


# Role of Dependency Parsing in KG

Combine tokens, dependency paths, and entity types to define rules.



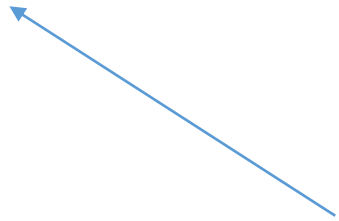
Bill Gates, the CEO of Microsoft, said ...  
Mr. Jobs, the brilliant and charming CEO of Apple Inc., said ...  
... announced by Steve Jobs, the CEO of Apple.  
... announced by Bill Gates, the director and CEO of Microsoft.  
... mused Bill, a former CEO of Microsoft.  
*and many other possible instantiations...*



# Within document Co-reference resolution

John was born in Liverpool, to Julia and Alfred Lennon.

He was a famous musician.



It consolidates the information extracted from the text.

# Problems with Entity Names

## Entities with Same Name

Same type of entities share names

Kevin Smith, John Smith,  
Springfield, ...

Things named after each other

Clinton, Washington, Paris,  
Amazon, Princeton, Kingston, ...

Partial Reference

First names of people, Location  
instead of team name, Nick names

## Different Names for Entities

Nick Names

Bam Bam, Drumpf, ...

Typos/Misspellings

Baarak, Barak, Barrack, ...

Inconsistent References

MSFT, APPL, GOOG...

# Entity Resolution

- Identifying and grouping different manifestations of the same real-world object

...during the late 60's and early 70's, **Kevin Smith** worked with several local...

...the term hip-hop is attributed to **Lovebug Starski**. What does it actually mean...



Like Back in 2008, the Lions drafted **Kevin Smith**, even though Smith was badly...

... backfield in the wake of **Kevin Smith**'s knee injury, and the addition of Haynesworth...



The filmmaker **Kevin Smith** returns to the role of Silent Bob...

Nothing could be more irrelevant to **Kevin Smith**'s audacious ''Dogma'' than ticking off...



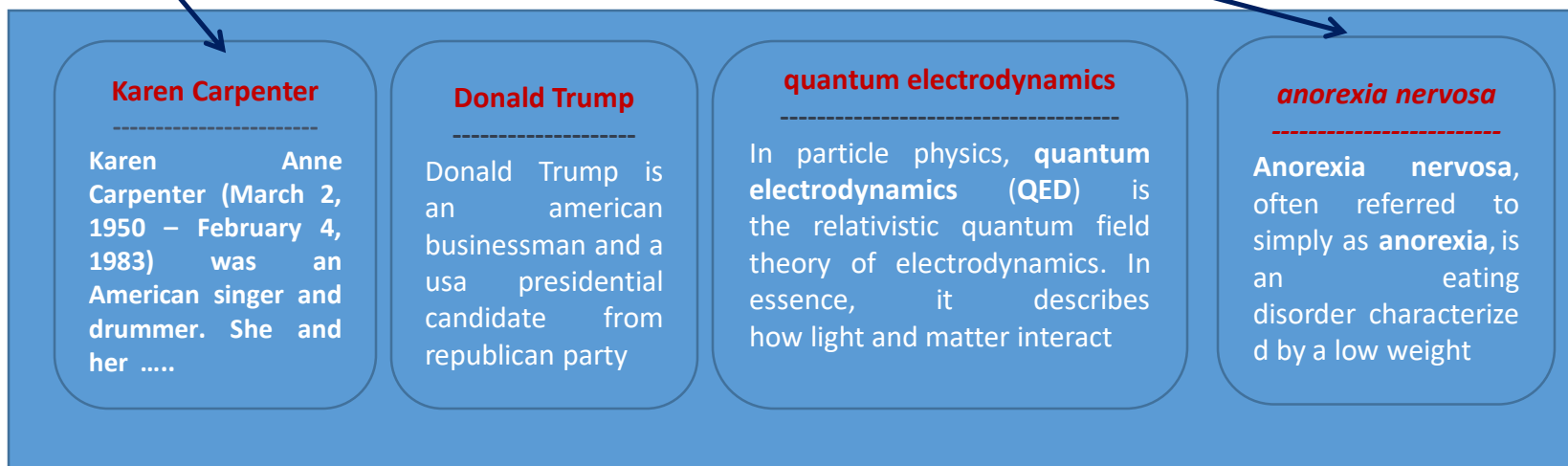
... The Physiological Basis of Politics," by **Kevin Smith**, Douglas Oxley, Matthew Hibbing...



# Entity linking

- Entity Linking (EL) problem
  - the task of **linking name mentions** in text with their **referent entities** in a **knowledge base**

***Karen Carpenter**, to appear more attractive, continued to diet even after losing 20 lbs, until her death at the age of 32. She died of cardiac arrest due to **anorexia nervosa**.*



# Entity linking: overall approach

***Karen Carpenter, to appear more attractive ... age of 32. She died due to anorexia nervosa.***

- Main steps

1. Entity extraction
2. String match with the encyclopedia entity catalog
3. Entity disambiguation

Michael Jordan is a former American basketball player who led the Chicago Bulls to six NBA championships.

Article [Talk](#) [Read](#) [Edit](#) [View](#)

## Michael Jordan (footballer)

From Wikipedia, the free encyclopedia

**Michael William Jordan** (born 7 April 1966) is an English football goalkeeper born in Cheshunt, Hertfordshire. He made seven appearances in the Football League for Chesterfield, having started his career as a trainee at Arsenal.

Contents

- 1 Career
- 1.1 Club career
- 1.2 International career
- 2 Honours
- 3 References
- 4 External links

### Career

[ edit ]

#### Club career

[ edit ]

Jordan signed for Arsenal as a scholar in 2002, turning professional on 1 November 2004 after making impressive performances for the youth team. However, he never played for the Arsenal first team; the closest he came was appearing on the bench for a League Cup match on 9 November 2004 against Everton, a match Arsenal won 3–1.<sup>[3]</sup>

After trials at Doncaster Rovers,<sup>[4]</sup> and Bournemouth,<sup>[5]</sup> Jordan signed for Yeovil Town on 9 March 2006 on a month's loan a goalkeeping cover during an injury crisis. The loan was later extended to the end of the season,<sup>[6]</sup> at which time Yeovil declined to make the deal permanent and the player returned to Arsenal.<sup>[7]</sup> On 30 June 2006, Arsenal released Jordan.<sup>[8]</sup>

Article [Talk](#) [Read](#) [View source](#) [View his](#)

## Michael Jordan

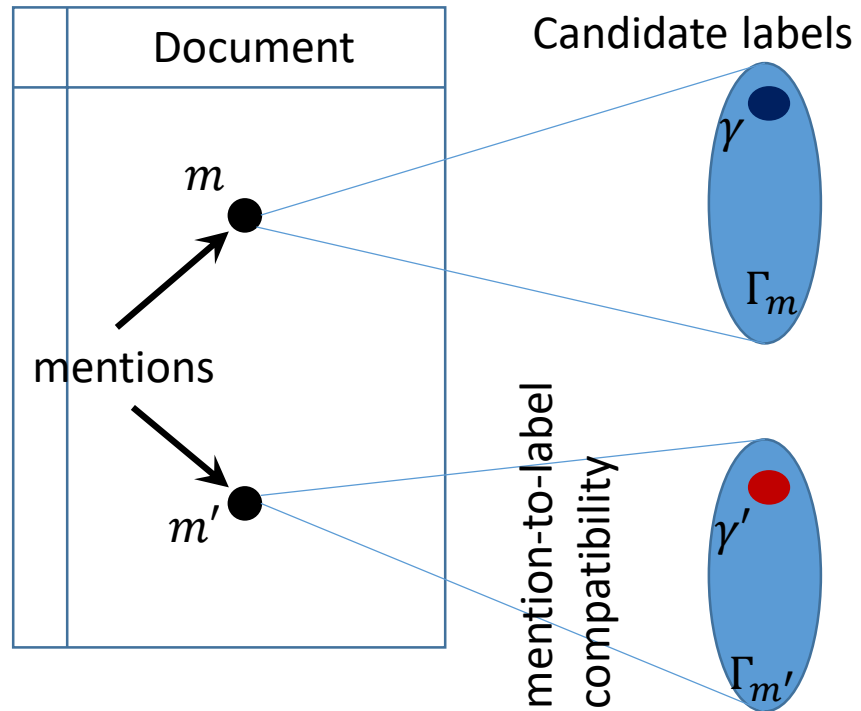
From Wikipedia, the free encyclopedia

*For other people named Michael Jordan, see Michael Jordan (disambiguation).*

**Michael Jeffrey Jordan** (born February 17, 1963), also known by his initials, **MJ**<sup>[3]</sup> is an American former professional basketball player. He is also a businessman, and principal owner and chairman of the Charlotte Hornets. Jordan played 15 seasons in the National Basketball Association (NBA) for the Chicago Bulls and Washington Wizards. His biography on the NBA website states: "By acclamation, Michael Jordan is the greatest basketball player of all time."<sup>[4]</sup> Jordan was one of the most effectively marketed athletes of his generation and was considered instrumental in popularizing the NBA around the world in the 1980s and 1990s.<sup>[5]</sup>

Jordan played three seasons for coach Dean Smith at the University of North Carolina. He was a member of the Tar Heels' national championship team in 1982. Jordan joined the NBA's Chicago Bulls in 1984 as the third overall draft pick. He quickly emerged as a league star, entertaining crowds with his prolific scoring. His leaping ability, illustrated by performing slam dunks from the free throw line in slam dunk contests, earned him the nicknames "Air Jordan" and "His Ainess". He also gained a reputation for being one of the best defensive players in basketball.<sup>[6]</sup> In 1991, he won his first NBA championship with the Bulls, and followed that achievement with titles in 1992 and 1993, securing a "three-peat". Although Jordan abruptly retired from basketball before the beginning of the 1993–94 NBA season to pursue a career in baseball, he returned to the Bulls in March 1995 and led them to three additional championships in 1996, 1997, and 1998, as well as an NBA-record 72 regular-season wins in the 1995–96 NBA season. Jordan retired for a second time in January 1999, but returned for two more NBA seasons from 2001 to 2003 as a member of the Wizards.

# Local Approach to Linking



- *Local* approaches disambiguate each mention in a document separately
- Utilizes clues such as the textual similarity between the document and each candidate disambiguation's Wikipedia page

# Issue with Local Approach

**Michael Jeffrey Jordan** (born February 17, 1963), also known by his initials, MJ, is an American former professional basketball player. **Jordan** joined the NBA's Chicago Bulls in 1984. **Michael Jordan** fuelled the success of Nike's Air Jordan sneakers. He also starred in the 1996 feature film Space Jam as himself.

## Michael Jordan (disambiguation)

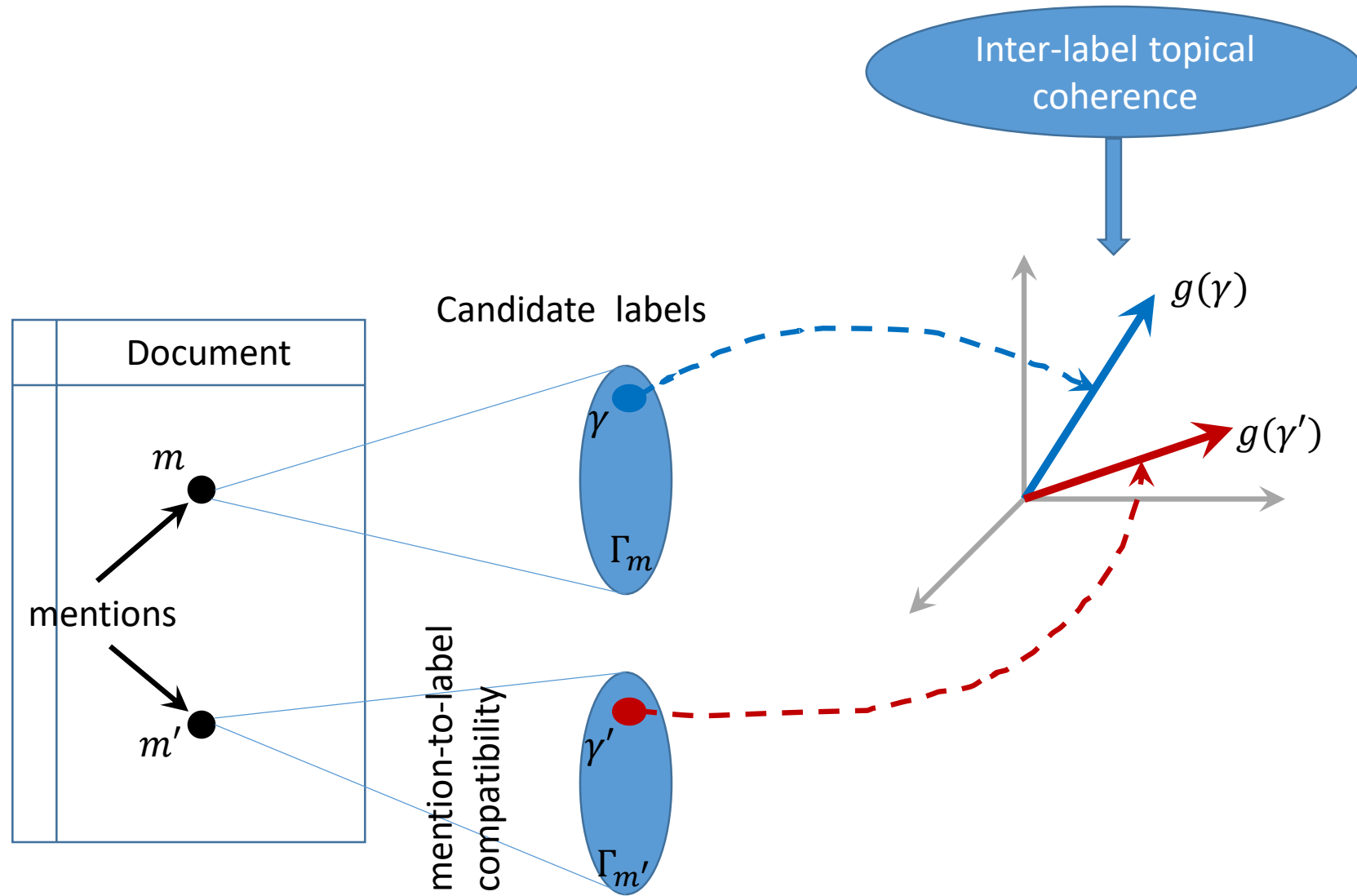
From Wikipedia, the free encyclopedia

**Michael Jordan** (born 1963) is an American basketball player.

**Michael Jordan** may also refer to:

- [Michael Jordan \(mycologist\)](#), English mycologist
- [Michael Jordan \(footballer\)](#) (born 1986), English goalkeeper (Arsenal, Chesterfield, Lewes)
- [Michael Jordan \(insolvency baron\)](#) (born 1931), English businessman
- [Mike Jordan](#) (born 1958), English racing driver
- [Mike Jordan \(baseball\)](#) (1863–1940), baseball player
- [Michael Jordan \(Irish politician\)](#), Irish Farmers' Party TD from Wexford, 1927–1932
- [Michael B. Jordan](#) (born 1987), American actor
- [Michael I. Jordan](#) (born 1957), American researcher in machine learning and artificial intelligence
- [Michael H. Jordan](#) (1936–2010), American executive for CBS, PepsiCo, Westinghouse
- [Michael-Hakim Jordan](#) (born 1977), American professional basketball player
- [Michal Jordan](#) (born 1990), Czech ice hockey player
- "Michael Jordan", a song by Kendrick Lamar featuring Schoolboy Q on the album [Overly Dedicated](#)

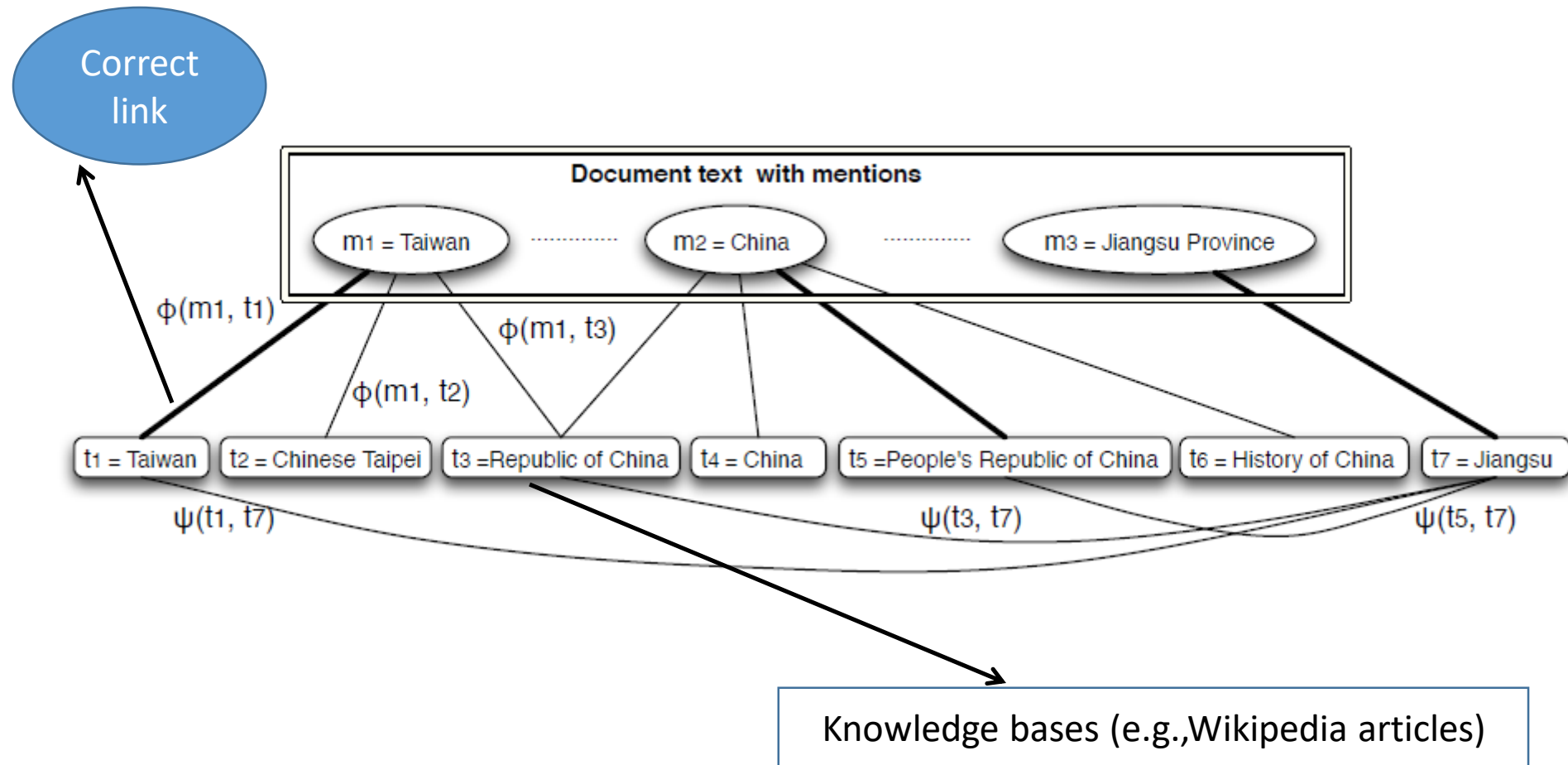
# Collective Entity Linking



# Problem Definition

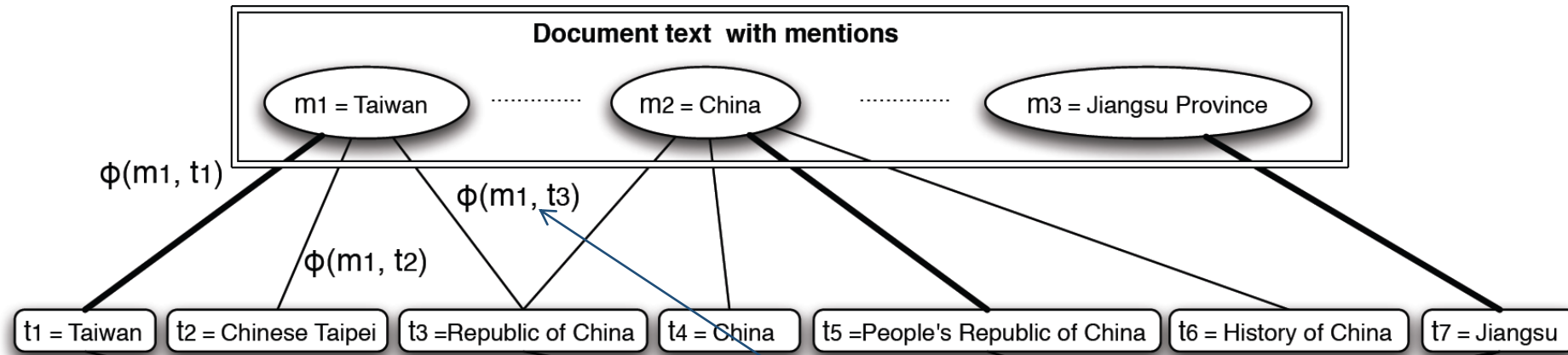
- Input
  - a document  $d$
  - Set of mentions  $M = \{m_1, m_2, \dots, m_N\}$
  - Set of Wikipedia titles  $W = \{t_1, t_2, \dots, t_{|W|}\}$
- Output
  - A mapping  $\Gamma: M \mapsto W$

# Problem Definition



many-to-one mapping

# Local approach

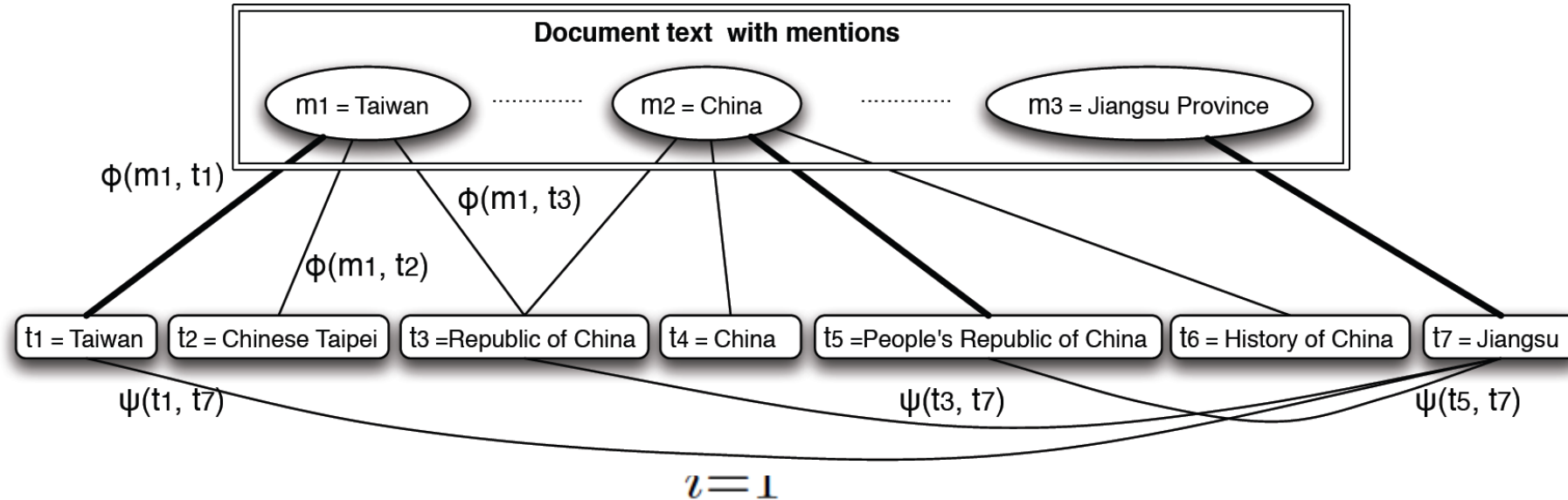


- $\Gamma$  is a solution to the problem
  - A set of pairs  $(m, t)$
- $m$ : a mention in the document
- $t$ : the matched Wikipedia Title

Local score of matching  
the mention to the title

$$\Gamma_{\text{local}}^* = \arg \max_{\Gamma} \sum_{i=1}^N \phi(m_i, t_i)$$

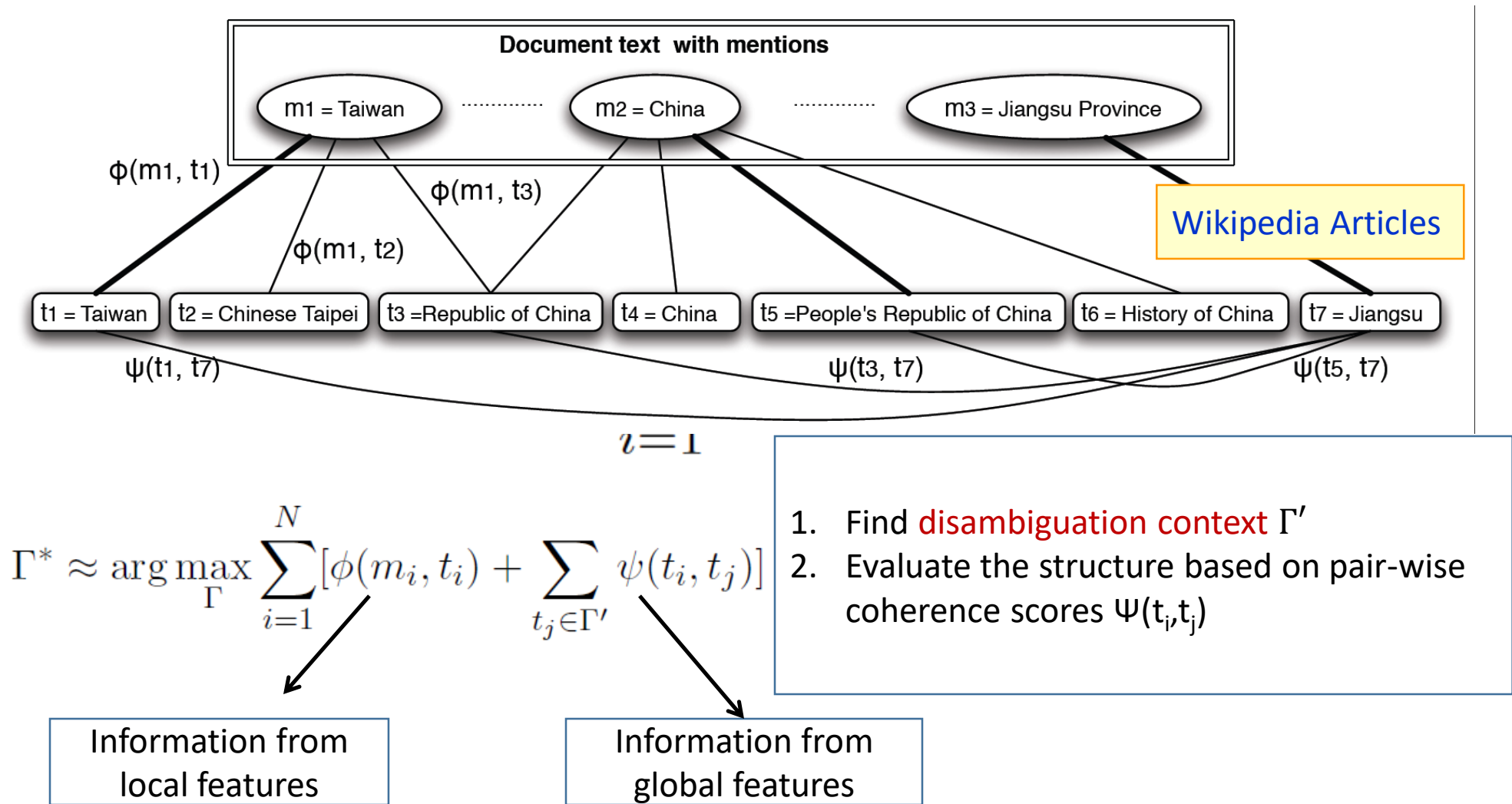
# Local + Global: using the Wikipedia structure



$$\Gamma^* = \arg \max_{\Gamma} [\sum_{i=1}^N \Phi(m_i, t_i) + \Psi(\Gamma)]$$

A "global" term –  
evaluating how good  
the solution is

# A tractable variation



# Local Features: $\phi_i(m, t)$

- Local features

- $\text{cosine-similarity}(\text{Text}(t), \text{Text}(m))$
- $\text{cosine-similarity}(\text{Text}(t), \text{Context}(m))$
- $\text{cosine-similarity}(\text{Context}(t), \text{Text}(m))$
- $\text{cosine-similarity}(\text{Context}(t), \text{Context}(m))$

- $\text{Text}(t)$  : TF-IDF vector of Wikipedia title  $t$
- $\text{Text}(m)$ : TF-IDF vector of  $d$  containing  $m$
- $\text{Context}(t)$ : TF-IDF vector of the context within which  $t$  is hyperlinked in Wikipedia
- $\text{Context}(m)$ : TF-IDF vector of context window of  $m$

- ❖ Combining local features

- $\Phi(m, t) = \sum_{i=1}^n w_i \phi_i(m, t)$ 
  - $w_i$  = the weight of the  $i$ -th feature to be learnt

# semantic relatedness measures

## ❖ Wikipedia relatedness measures

- Normalized Google Distance

$$NGD(L_1, L_2) = \frac{\text{Log}(\text{Max}(|L_1|, |L_2|)) - \text{Log}(|L_1 \cap L_2|)}{\text{Log}(|W|) - \text{Log}(\text{Min}(|L_1|, |L_2|))}$$

- $W$  = set of Wikipedia titles

- Pointwise Mutual Information

- $PMI(L_1, L_2) = \log\left(\frac{|L_1 \cap L_2|/|W|}{\frac{|L_1|}{|W|} * \frac{|L_2|}{|W|}}\right)$

# Global Features: $\psi_i(t, t')$

$$I_{[t_i - t_j]} * PMI(InLinks(t_i), InLinks(t_j))$$

$$I_{[t_i - t_j]} * NGD(InLinks(t_i), InLinks(t_j))$$

$$I_{[t_i - t_j]} * PMI(OutLinks(t_i), OutLinks(t_j))$$

$$I_{[t_i - t_j]} * NGD(OutLinks(t_i), OutLinks(t_j))$$

$$I_{[t_i \leftrightarrow t_j]}$$

$$I_{[t_i \leftrightarrow t_j]} * PMI(InLinks(t_i), InLinks(t_j))$$

$$I_{[t_i \leftrightarrow t_j]} * NGD(InLinks(t_i), InLinks(t_j))$$

$$I_{[t_i \leftrightarrow t_j]} * PMI(OutLinks(t_i), OutLinks(t_j))$$

$$I_{[t_i \leftrightarrow t_j]} * NGD(OutLinks(t_i), OutLinks(t_j))$$

$I_{[t_i - t_j]} = 1$ , iff  $t_i$  links to  $t_j$  or vice versa

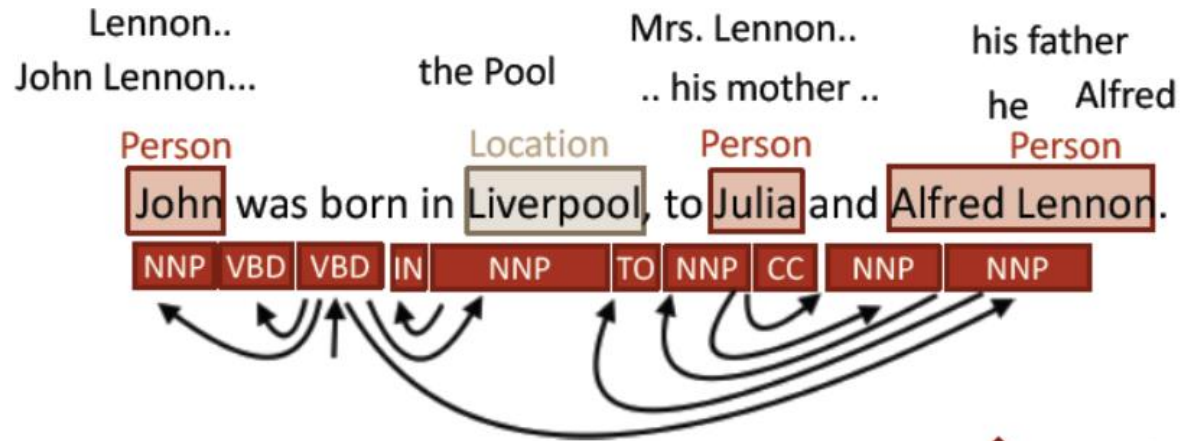
$I_{[t_i \leftrightarrow t_j]} = 1$ , iff  $t_i$  and  $t_j$  point to each other

- Combining features:

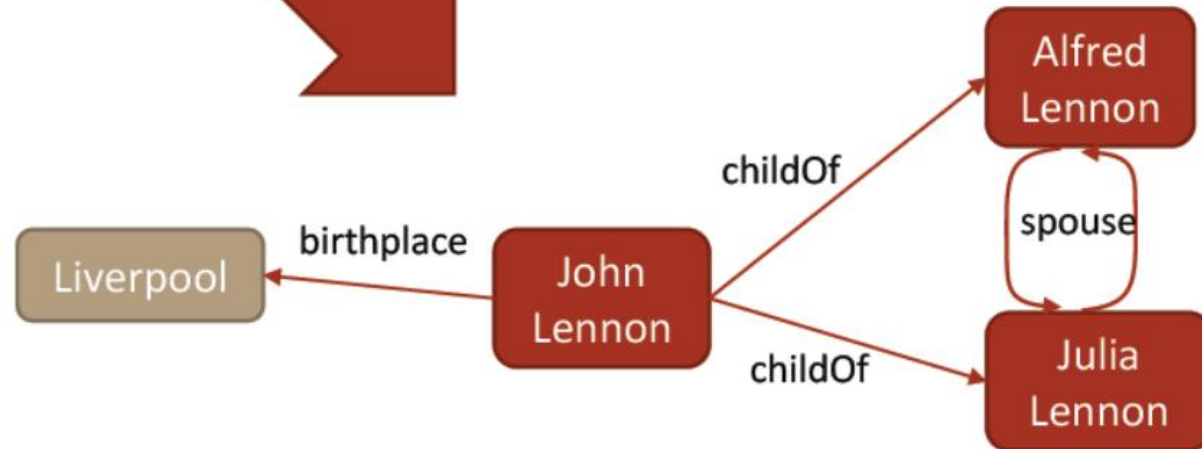
- $\psi(t, t') = \sum_{i=1}^n w_i \psi_i(t, t')$

# Information Extraction

# Information Extraction



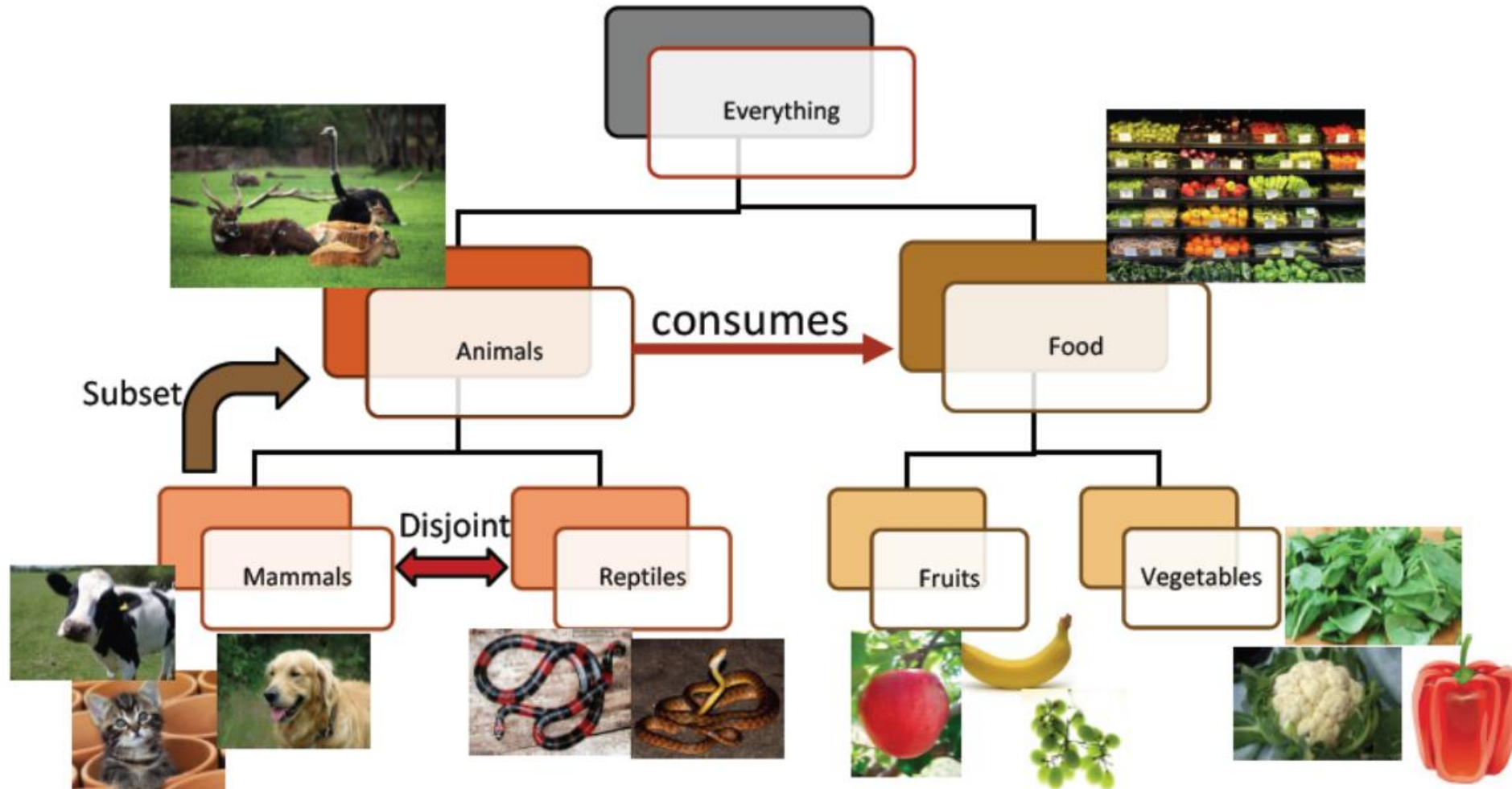
Information Extraction



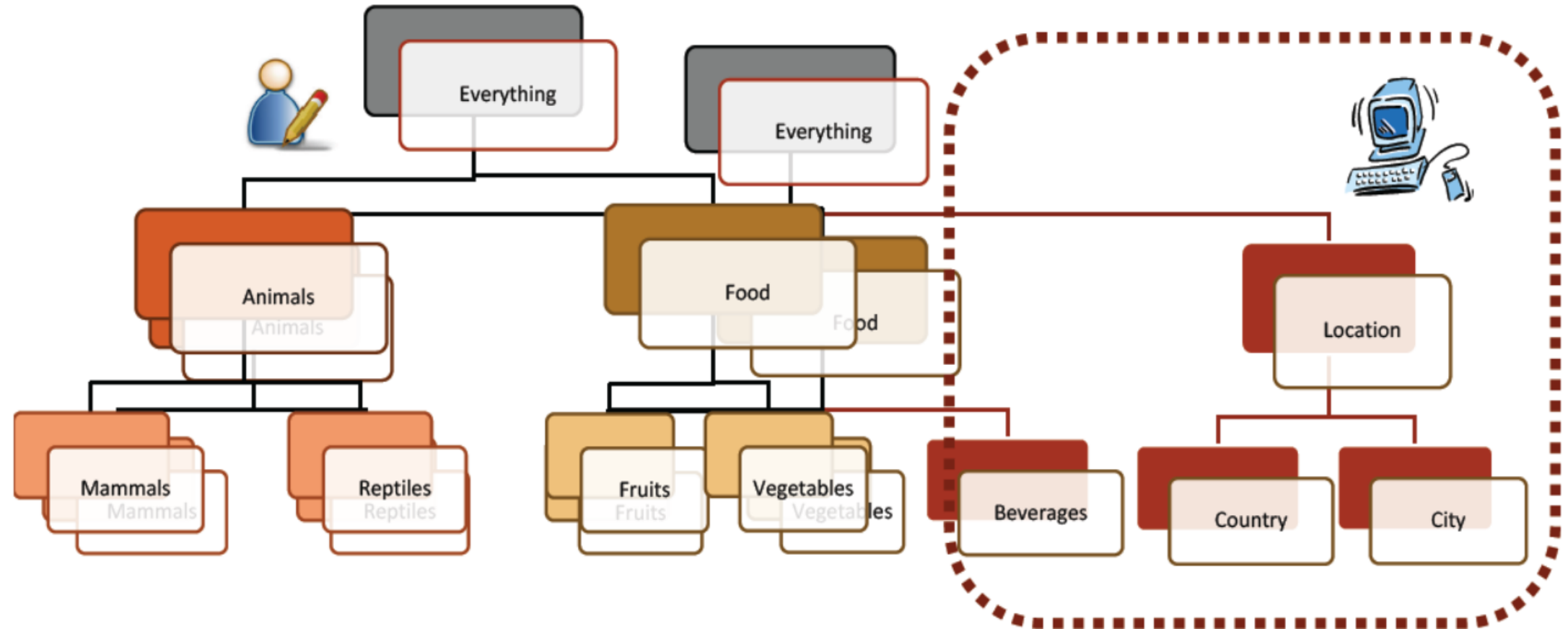
# Information Extraction

- Main problems
  - Defining domain
  - Learning extractors
  - Scoring the facts

# Defining domain: Manual Approach



# Defining Domain: Semi-manual approach



Manually defined

Semi-supervised learning  
using manual labels

# Defining Domain: Automatic

- Any noun phrase is a candidate entity
  - Dog, cat, cow, reptile, mammal, apple, greens, mixed greens
- Any verb phrase is a candidate relation
  - Eats, feasts on, grazes, consumes

# Learning extractor: Manual

<PERSON> plays in <BAND>

- Requires hand-building patterns for each relation!
  - hard to write; hard to maintain
  - there are zillions of them
  - domain-dependent

# Learning extractor: Bootstrapping

- If you don't have enough annotated text to train on ...
- But you have:
  - some seed instances of the relation
  - (or some patterns that work pretty well)
  - and lots & lots of unannotated text (e.g., the web)

# Learning extractor: Bootstrapping

- Target relation: **burial place**
- Seed tuple: **[Mark Twain, Elmira]**
- Search for sentences with both “Mark Twain” and “Elmira”
  - “Mark Twain is buried in Elmira, NY.”
    - $\rightarrow$  X is buried in Y
  - “The grave of Mark Twain is in Elmira”
    - $\rightarrow$  The grave of X is in Y
  - “Elmira is Mark Twain’s final resting place”
    - $\rightarrow$  Y is X’s final resting place
- Use those patterns to search for new tuples

# Reference

- Local and Global Algorithms for Disambiguation to Wikipedia ;  
Ratinov et al.
- Knowledge graph: <https://kgtutorial.github.io/aaai.html>