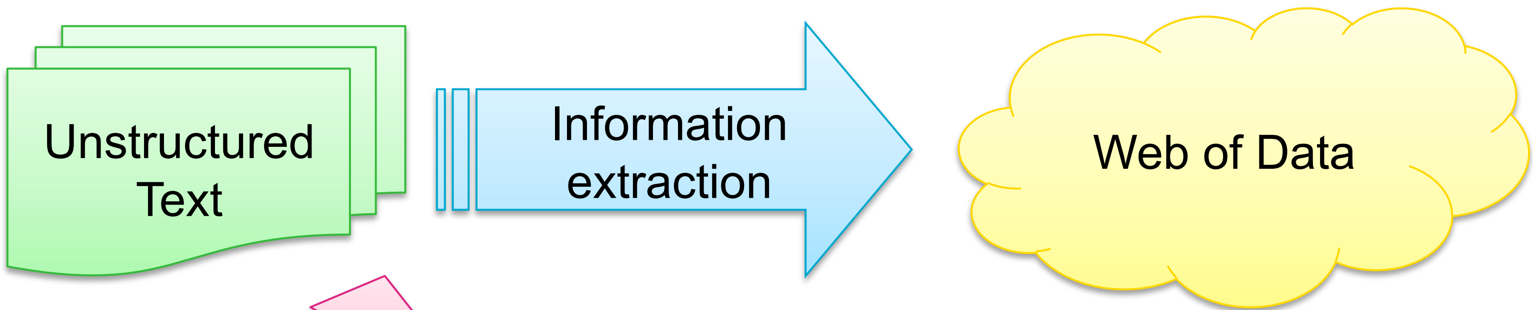
UNIVERSITEIT TWENTE.

HANDLING UNCERTAINTY IN INFORMATION EXTRACTION

Maurice van Keulen and Mena Badieh Habib

URSW
23 Oct 2011

INFORMATION EXTRACTION



Inherently imperfect process

Word "Paris"

source: GeoNames

- First name? City? …
- City => over 60 cities "Paris"
- Toponyms: 46% >2 refs

"We humans happily deal with doubt and misinterpretation every day;
Why shouldn't computers?"

Goal: Technology to support the development of domain specific information extractors

SHERLOCK HOLMES-STYLE INFORMATION EXTRACTION

"when you have eliminated the impossible, whatever remains, however improbable, must be the truth"

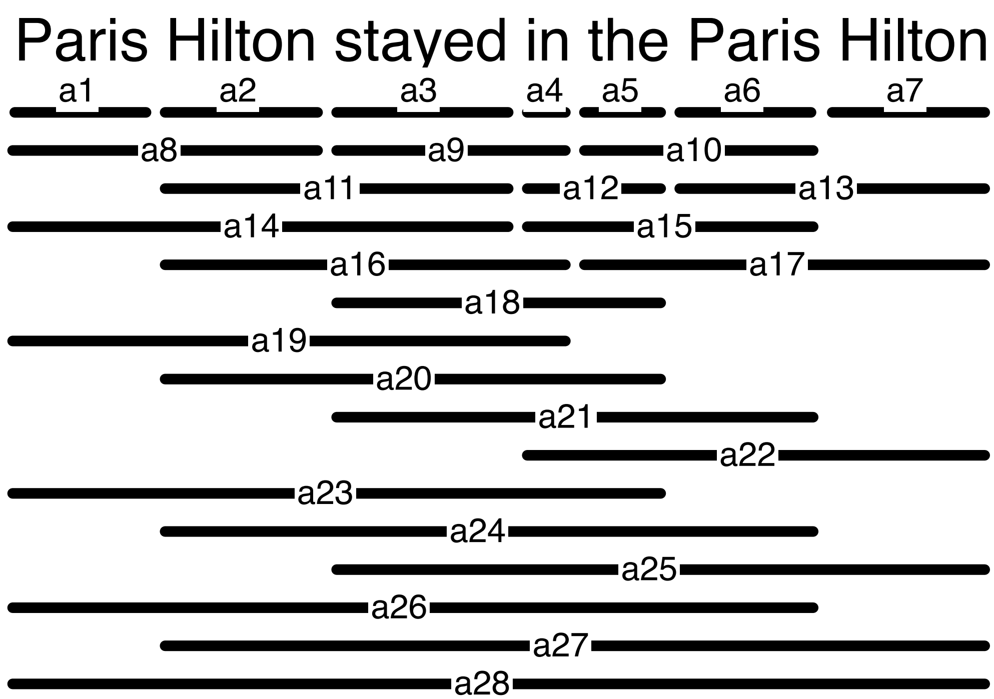
Information extraction is about gathering enough evidence to decide upon a certain combination of annotations among many possible ones
Evidence comes from ML + developer (generic) + end user (instances)

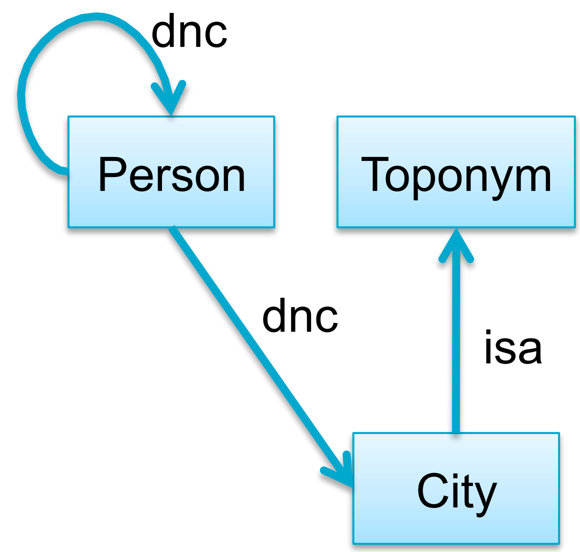
- **Annotations are uncertain**
  Maintain alternatives + probabilities throughout process (incl. result)
- **Unconventional starting point**
  Not "no annotations", but "no knowledge, hence anything is possible"
- Developer **interactively** defines information extractor until **"good enough"**
  Iterations: Add knowledge, apply to sample texts, evaluate result
- **Scalability** for storage, querying, manipulation of annotations
  From my own field (databases): Probabilistic databases?

SHERLOCK HOLMES-STYLE INFORMATION EXTRACTION

EXAMPLE: NAMED ENTITY RECOGNITION (NER)

> "when you have eliminated the impossible, whatever remains, however improbable, must be the truth"





inter-actively defined

SHERLOCK HOLMES-STYLE INFORMATION EXTRACTION

EXAMPLE: NAMED ENTITY RECOGNITION (NER)

Paris Hilton stayed in the Paris Hilton

annotations

	b	e	type	...	wsd
a_1^1	1	1	Person	...	$\{x_1^1 = 1\}$
a_1^2	1	1	City	...	$\{x_1^2 = 1\}$
a_8^1	1	2	Person	...	$\{x_8^1 = 1\}$
	...	...	...	...	

w

x		
x_1^1		
x_1^1		
x_1^2	0	
x_1^2	1	0.
x_8^1	0	0.2
x_8^1	1	0.8
...	...	...

- |A|=O(klt) *lin r?!*
  - k: leng
  - l: ma ed
  - t: nu
- 6 (I saw one with 5)
- 0 * 20 * 6
- ,000 possible annotations

Although conceptual/theoretical, it doesn't seem to be a severe challenge for a probabilistic database

The problem is not in the amount of alternative annotations!

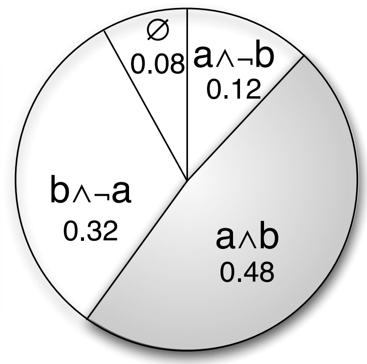
ADDING KNOWLEDGE = CONDITIONING

Paris Hilton stayed in the Paris Hilton

annotations

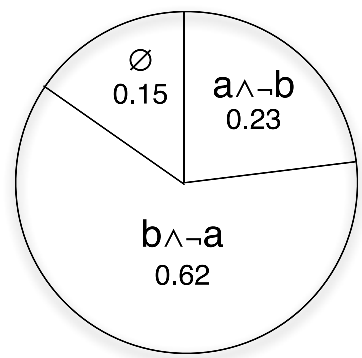
	b	e	type	...	wsd
a_1^1	1	1	Person	...	$\{x_1^1 = 1\}$
a_1^2	1	1	City	...	$\{x_1^2 = 1\}$
a_8^1	1	2	Person	...	$\{x_8^1 = 1\}$
				...	

a and b independent
P(a)=0.6
P(b)=0.8



Person --- dnc --- City

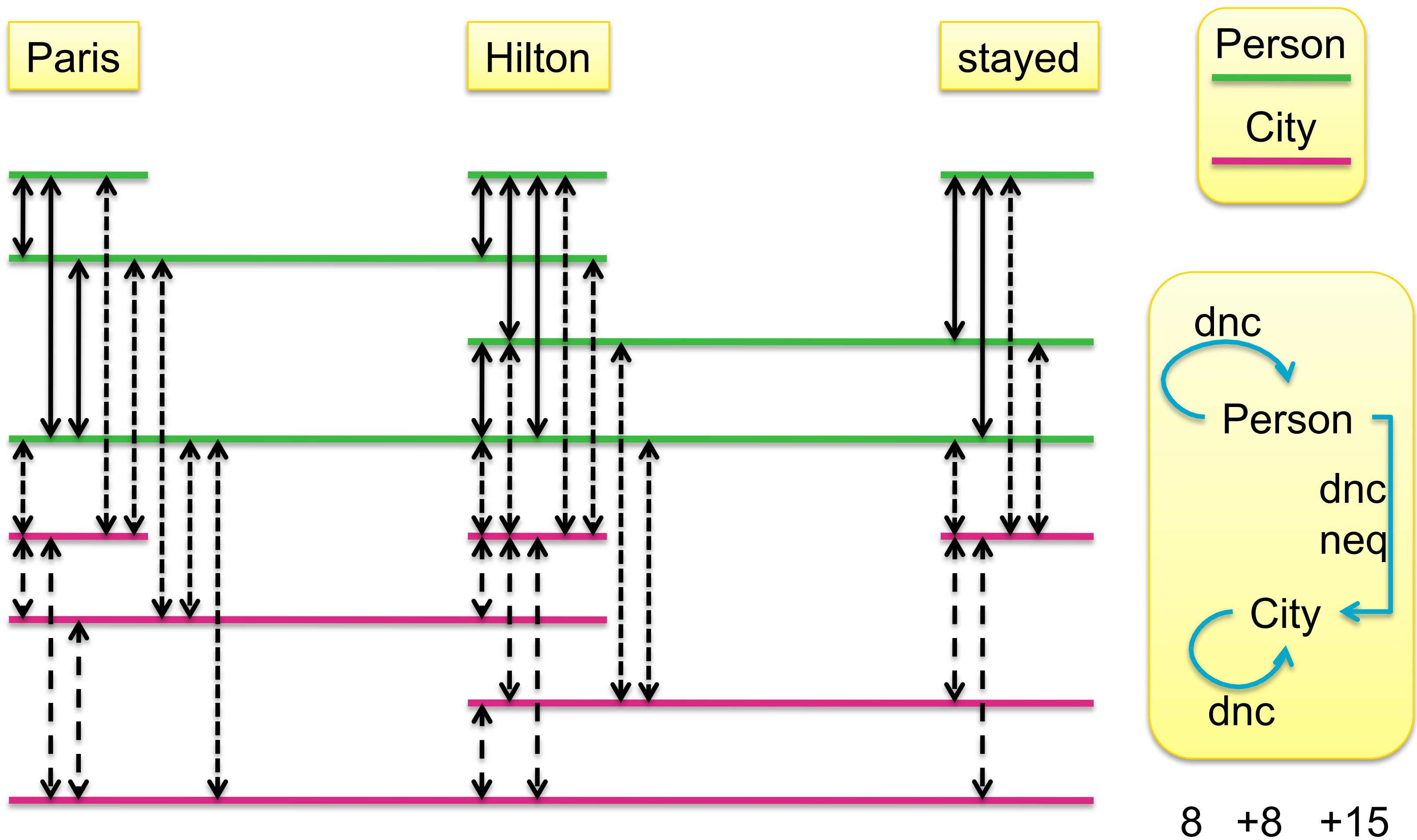
a and b mutually exclusive
(a∧b is not possible)



x_1^2 (“Paris” is a City) [a]
x_8^1 (“Paris Hilton” is a Person) [b]
become mutually exclusive

ADDING KNOWLEDGE CREATES DEPENDENCIES

NUMBER OF DEPS MAGNITUDES IN SIZE SMALLER THAN POSSIBLE COMBINATIONS



PROBLEM AND SOLUTION DIRECTIONS

I'm looking for a scalable approach to reason and redistribute probability mass considering all these dependencies to find the remaining possible interpretations and their probabilities

- Feasibility approach hinges on efficient representation and conditioning of probabilistic dependencies
- Solution directions (in my own field):
  - Koch etal VLDB 2008 (Conditioning in MayBMS)
  - Getoor etal VLDB 2008 (Shared correlations)
- This is not about only learning a joint probability distribution. Here I'd like to estimate a joint probability distribution based on initial independent observations and then batch-by-batch add constraints/dependencies and recalculate
- Techniques out there that fit this problem?

Questions / Suggestions?