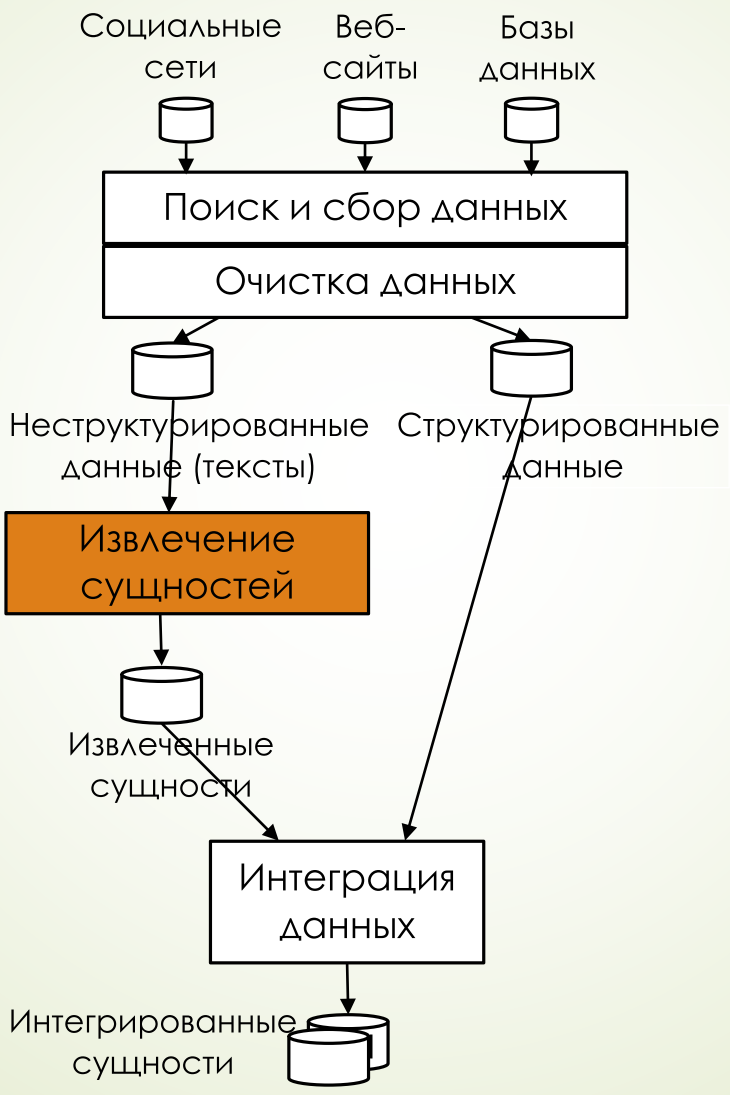
Извлечение и интеграция информации из больших данных

Представление знаний

Брюхов Д.О., Вовченко А.Е., Калиниченко Л.А., Ковалев Д.Ю., Скворцов Н.А.

Институт Проблем Информатики ФИЦ ИУ РАН

Процесс интеграции информации



Процесс интеграции информации

Def: Information Extraction

- Information Extraction (IE) is the process of deriving structured factual information from digital text documents.

Elvis is a singer.

Persons	Occupation
Elvis	singer

Def: Structured Information

- Structured information is any data that expresses relationships between entities.
- Think of it as a database table.

Persons	Occupation
Elvis	singer

Def: Entity

- An entity (also: instance, resource) is any particular object of the world or imagination, be it abstract or concrete.

Def: Class

- A class (also: concept) is a set of similar entities.

- Persons
- Cars
- Universities
- Movies
- Actors
- …

Def: Instance of a class

- An entity is an instance of a class(also: belongs to a class, has the type), if the entity is an element of that class.

Def: Multiple inheritance

- Multiple inheritance is the phenomenon that an entity belongs to multiple classes.

Def: Subclass/Superclass

- A class is a subclass of another class, if it is a subset of that class.
- A class is a superclass of another class, if it is a superset of that class.

Intuition: Relations

- A relation is like a table.
- Relation "born":

Person	City	Year
Mironov	Moscow	1941
Ekaterina II	Stettin	1729

Def: Tuple

- An n-tuple is an ordered list of n elements.
  - < a; b; c >
  - < this; is; a; different; tuple >
  - < a; different; tuple; is; this >

Def: Cartesian Product

- The cartesian product of n sets is the set of all n-tuples where the i-th element is an element of the i-th set.
  - $C_1 \times C_2 \times \ldots \times C_n := \{\langle c_1, c_2, \ldots, c_n \rangle : c_1 \in C_1, \ldots, c_n \in C_n\}$

- Person × City × Year :=
  - {< Mironov, Moscow, 1941>,
  - < Mironov, Moscow, 1729>,
  - < Mironov, Stettin, 1941>,
  - < Mironov, Stettin, 1729 >,
  - < Ekaterina II, Moscow, 1941>,
  - < Ekaterina II, Moscow, 1729>,
  - < Ekaterina II, Stettin, 1941>,
  - < Ekaterina II, Stettin, 1729>,
  - …}

Relation

- A relation is a subset of a cartesian product
- Born :=
  - {< Mironov, Moscow, 1941>,
  - < Ekaterina II, Stettin, 1729>,
  - …}

Def: Relation

- A relation (also: predicate) over classes is a subset of their cartesian product.
- $R \subseteq C_1 \times C_2 \times \ldots \times C_n$
- born $\subseteq$ person $\times$ city $\times$ year

  born :=

  {< Mironov, Moscow, 1941>,

  < Ekaterina II, Stettin, 1729>,

  ...}

Def: Domain

- The classes of a relation are called the domains of the relation.
- $R \subseteq C_1 \times C_2 \times \ldots \times C_n$
- born $\subseteq$ person $\times$ city $\times$ year

Def: Arity

- The arity of a relation is the number of its domains.
- $R \subseteq C_1 \times C_2 \times \ldots \times C_n$

- born $\subseteq$ person $\times$ city $\times$ year
  - The arity of "born" is 3

Syntax

- We write

  $r(x_1, \ldots, x_n)$

- to mean

  $< x_1, \ldots, x_n > \in r$

Def: Tuples and Arguments

- An element of a relation is called tuple (also: fact, statement, record).
- Its components are called arguments.

Arguments

$\underbrace{\text{born (Mironov, Moscow, 1941)}}_{\text{Tuples}}$

Tuples and Domains

- The i-th element of a tuple of a relation is an instance of the i-th domain of the relation.

- born $\subseteq$ person $\times$ city $\times$ year
- born (Mironov, Moscow, 1941)

$\in$ person

Relations and Tables

Domain Domain Domain

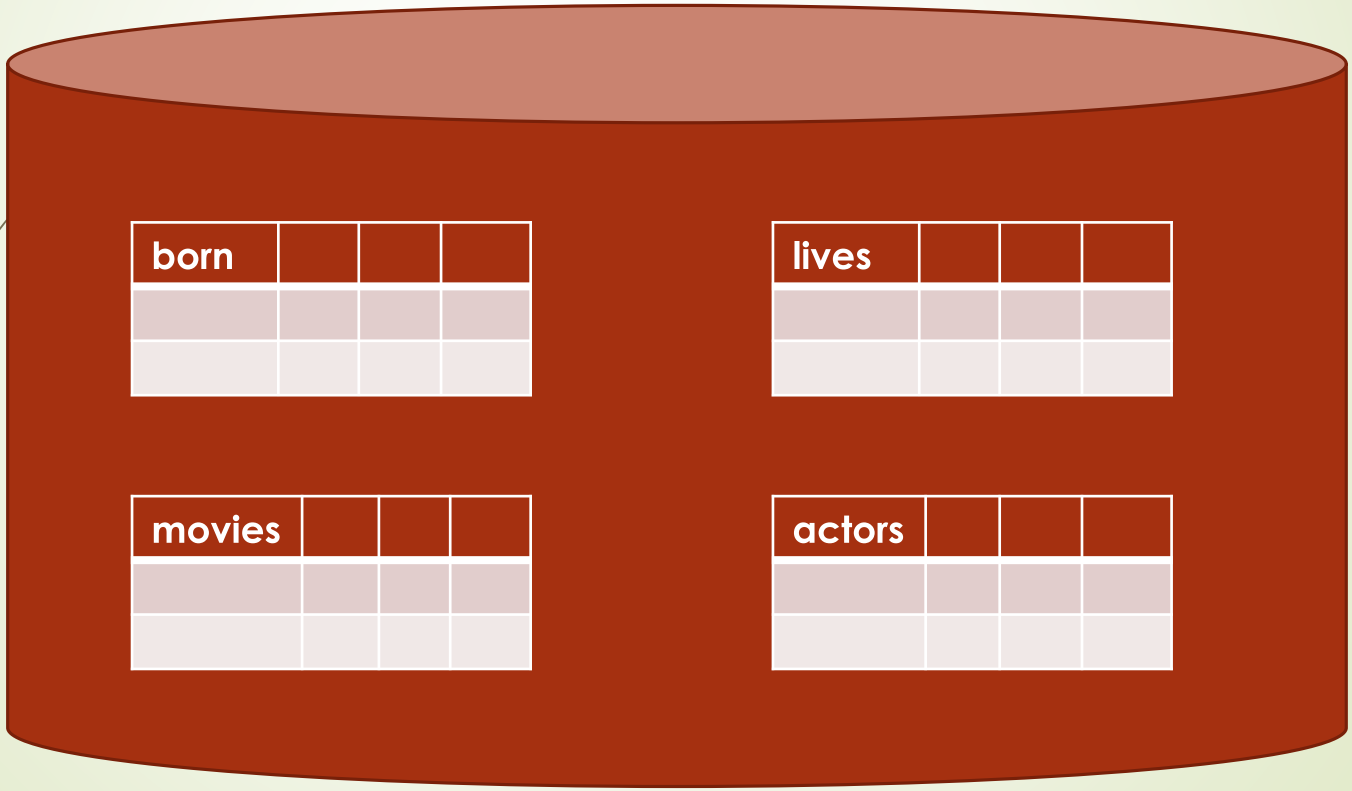
Person	City	Year
Mironov	Moscow	1941
Ekaterina II	Stettin	1729

Tuple

Arity = 3

Def:Knowledge base

- A knowledge base (KB, also: database,fact collection) is a set of relations.



Def: Binary Relation

- A relation of arity 2 is called a binary relation.

- Its first domain is called "domain".
- Its second domain is called "range".

- drives $\subseteq$ person $\times$ car

domain range

Def: Pair

- A pair is a tuple with 2 elements.
- We will call them subject and object.

- drives $\subseteq$ person $\times$ car
- drives(Mironov, Volga)

Subject Object

Def: Inverse

- The inverse of a binary relation r is a relation r', such that r'(x,y) iff r(y,x).

- drives $\subseteq$ person $\times$ car
- drives(Mironov, Volga)

inverses of each other

- hasDriver $\subseteq$ car $\times$ person
- hasDriver(Volga, Mironov)

Def: Function

- A function (also: functional relation) is a binary relation that has at most 1 object for each subject.

- hasNationality
- hasSpouse

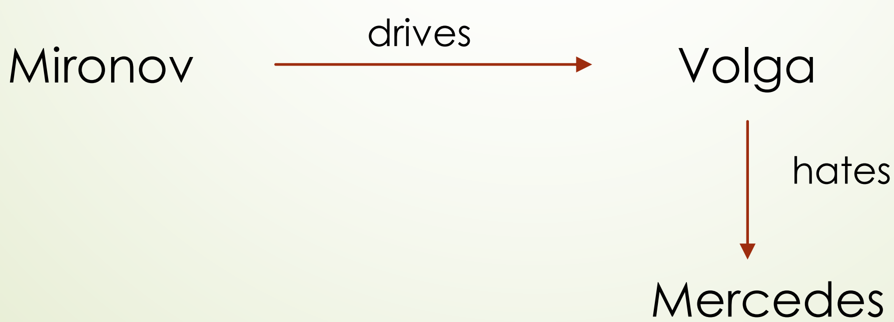
Def: Inverse Functional Relation

- An inverse functional relation is a relation whose inverse is functional
- hasCitizen
- hasSpouse

Def: Knowledge Graph

- A knowledge graph (also: Entity-Relationship graph) for a KB with binary relations is a directed labeled multi-graph that has an edge x->y with label r, iff r(x,y) is in the KB.

- Knowledge Base:
  - drives(Mironov, Volga)
  - hates(Volga, Mercedes)
- Knowledge Graph:



Def: Triple Store

- A triple store is a table that contains a KB of binary relations in the form of 3 columns: subject, relation, object.

Subject	Relation	Object
Mironov	drives	Volga
Volga	hates	Mercedes

Binary Rel. Representations

- Binary relations can be represented as
  - relation

    drives $\subseteq$ person $\times$ car

    drives(Mironov, Volga)
  - table

drives	Person	Car
	Mironov	Volga

  - graph

$\xrightarrow{\text{drives}}$

  - triple store

Subject	Relation	Object
Mironov	drives	Volga
Volga	hates	Mercedes