Semantic Web

7. Ontology Matching

PD Dr. Matthias Thimm

thimm@uni-koblenz.de

Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau

WeST People and Knowledge Networks

Ontology matching

Observation: People design ontologies and do not take existing ontologies into account.

Problem: There exist multiple ontologies with different vocabularies specifying (parts of) the same domain (and there always will be).

Solution: Approaches to *Ontology matching* (or *Ontology alignment*) aim at comparing two ontologies and finding relationships between their concepts to promote re-usability.

Outline

Outline

The Ontology Alignment Problem 1/4

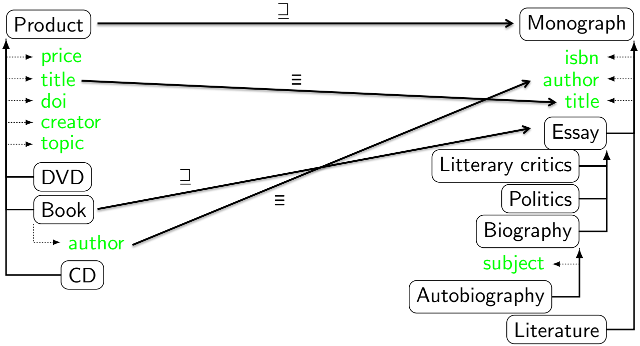
Problem definition (informal):

Given two ontologies (as e. g. description logic knowledge bases, or relational tables schemas, RDF data sets) find a set of relationships (e. g. equivalence, subsumption, disjointness) that hold between the entities of each ontology.

The Ontology Alignment Problem 2/4

The Ontology Alignment Problem 3/4

The Ontology Alignment Problem 4/4



Formalization: Abstract Ontology

Ontology alignment is not only applicable to description logic knowledge bases but also to RDF data sets, XML schemas, etc.. Therefore we use the following abstract definition of an ontology:

Definition

An *ontology* O is a tuple $O = (C, R, I, A)$ where

1. C is a set of concepts,
2. R is a set of relationships,
3. I is a set of instances, and
4. A is a set of axioms (terminological and assertional)

Formalization: Correspondences, Alignment

Let $O_1 = (C_1, R_1, I_1, A_1)$, $O_2 = (C_2, R_2, I_2, A_2)$ be two ontologies.

Definition

A *correspondence* M between O_1 and O_2 is a tuple $M = (e, e', R, n)$ with

1. e and e' are entities of O_1 and O_2:
   1.1 $e \in C_1$ and $e' \in C_2$, or
   1.2 $e \in R_1$ and $e' \in R_2$, or
   1.3 $e \in I_1$ and $e' \in I_2$.
2. $R \in \{\equiv, \sqsubseteq, \perp\}$ (equivalence, subsumption, disjointness)
3. $n \in [0, 1]$ is a *confidence degree*

An *alignment* A between O_1 and O_2 is a set of correspondences between O_1 and O_2.

Example 1/3

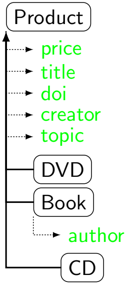
$O_1 = (C_1, R_1, I_1, A_1)$ is given via

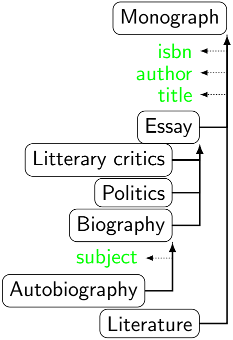
$$\begin{aligned} C_1 &= \{\text{Product}, \text{DVD}, \text{CD}\} \\ R_1 &= \{\text{price}, \text{title}, \text{doi}, \text{creator}, \text{topic}, \text{author}\} \\ I_1 &= \emptyset \\ A_1 &= \{\mathit{Product} \sqsubseteq \exists \mathit{price}.\top, \mathit{DVD} \sqsubseteq \mathit{Product}, \ldots\} \end{aligned}$$

$O_2 = (C_2, R_2, I_2, A_2)$ is given via

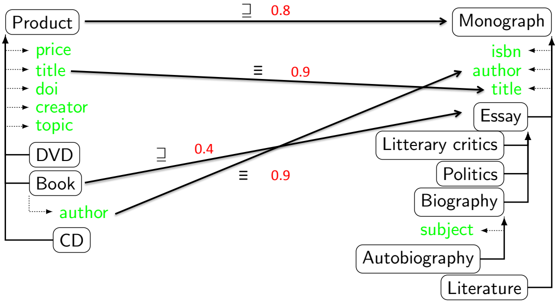
$$\begin{aligned} C_2 &= \{\text{Monograph}, \text{Essay}, \text{Literary critics}, \text{Politics}, \text{Biography}, \text{Autobiography}, \text{Literature}\} \\ R_2 &= \{\text{isbn}, \text{author}, \text{title}, \text{subject}\} \\ I_2 &= \emptyset \\ A_2 &= \{\mathit{Monograph} \sqsubseteq \exists \mathit{isbn}.\top, \mathit{Essay} \sqsubseteq \mathit{Monograph}, \ldots\} \end{aligned}$$

Example 2/3


Product
price
title
doi
creator
topic
DVD
Book
author
CD


Monograph
isbn
author
title
Essay
Litterary critics
Politics
Biography
subject
Autobiography
Literature

Example 3/3



Alignment $A = \{M_1, M_2, M_3, M_4\}$

$$M_1 = (o2 : Monograph, o1 : Product, \sqsubseteq, 0.8)$$
$$M_2 = (o1 : title, o2 : title, \equiv, 0.9)$$
$$M_3 = (o2 : Essay, o1 : Book, \sqsubseteq, 0.4)$$
$$M_4 = (o1 : author, o2 : author, \equiv, 0.9)$$

Outline

Dimensions for Matching

- Input dimensions
  - Underlying models (XML, OWL, DLs)
  - Components: concepts, relations, instances, axioms
  - Schema-level vs. instance-level
- Process dimensions (what information is used and how?)
  - String comparisons, language aspects
  - Graph structure
  - Approximate vs. Exact
- Output dimensions
  - Cardinality (one-to-one mappings or one-to-many)
  - Equivalence or also other relations (subsumption)
  - Graded vs. absolute confidence

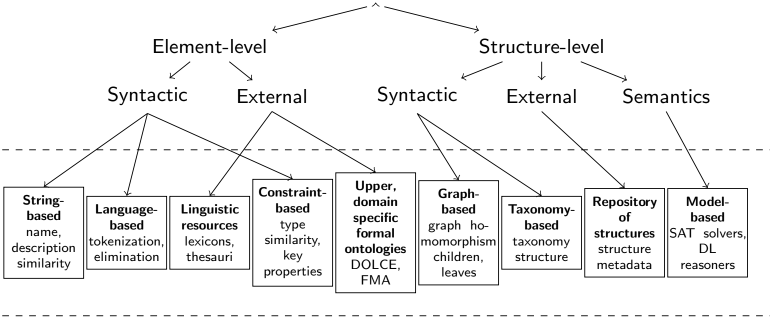
Three layers of classification

- The upper layer
  - Granularity of match (local or global)
  - Interpretation of the input information (syntax, external information)
- The lower layer (type of information used)
  - terminological information (string/language comparison)
  - structural information (graph similarity)
  - semantic information (logic-based)
- The middle layer: basic approaches

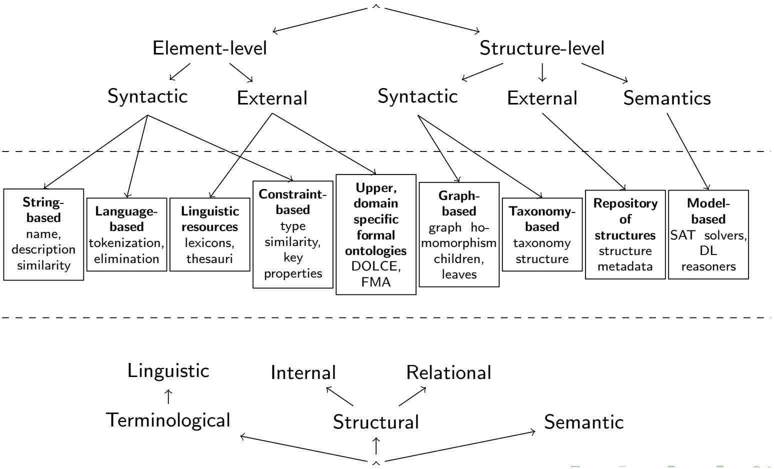
Classification of Ontology Alignment Approaches

- Element-level
  - Syntactic
  - External
- Structure-level
  - Syntactic
  - External
  - Semantics

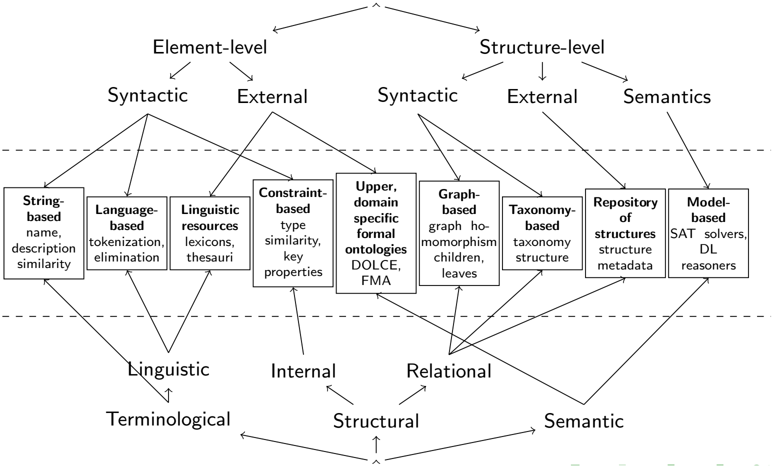
Classification of Ontology Alignment Approaches



Classification of Ontology Alignment Approaches



Classification of Ontology Alignment Approaches



String-based techniques: Prefix and Suffix

Let s, s' be the names of two entities from O_1 and O_2 (concepts, relations, or instances).

- Prefix-based comparison
  - s and s' are similar if s is a prefix of s'
  - Confidence degree can be obtained by comparing the length of s and s'
  - Concepts `net` and `network` are similar with confidence $3/7$
  - Concepts `hot` and `hotel` are similar with confidence $3/5$
- Suffix-based comparison
  - s and s' are similar if s is a suffix of s'
  - Confidence degree can be obtained by comparing the length of s and s'
  - Concepts `ID` and `PID` are similar with confidence $2/3$
  - Concepts `word` and `sword` are similar with confidence $4/5$

String-based techniques: Levenshtein distance

If $s = a_1 \dots a_n$ is a string define $\hat{s} = a_1 \dots a_{n-1}$. If $s' = b_1 \dots b_m$ is another string define $1_{s \sim s'} = 0$ if $a_n = b_m$, otherwise $1_{s \sim s'} = 1$. $|s|$ is the number of characters in s.

Let s, s' be the names of two entities from O_1 and O_2 (concepts, relations, or instances).

Definition

The *Levenshtein distance* (or *edit distance*) $lev(s, s')$ is defined via

$$lev(s,s') = \begin{cases} \max\{|s|, |s'|\} & \text{if } \min\{|s|, |s'|\} = 0 \\ \min\{lev(\hat{s}, s') + 1, lev(s, \hat{s}') + 1, lev(\hat{s}, \hat{s}') + 1_{s \sim s'}\} & \text{otherwise} \end{cases}$$

Examples:
$lev(\texttt{Product}, \texttt{Produkt}) = 1$
$lev(\texttt{Son}, \texttt{Sun}) = 1$
$lev(\texttt{House}, \texttt{Building}) = 8$

Language-based techniques

- Tokenization
  - Parse (concept) names into tokens by recognizing punctuation, cases
  - `BigBrownHorse` becomes {`big`, `brown`, `horse`}
  - Compare set of tokens rather than names: `BigBrownHorse` is equivalent to `big_brown_horse`
- Lemmatization
  - Bring names into normal forms (singular, plural)
  - `RunningHorses` becomes {`run`,`horse`}
- Elimination
  - Remove stop words from entity names
  - `a`, `the`, `by`, ...

Techniques based on Linguistic Resources

Let s, s' be the names of two concepts from O_1 and O_2.

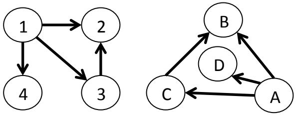
Use e. g. WordNet to compare meanings of words.

- if s is a hyponym or meronym of s' then $s \sqsubseteq s'$
  - $\texttt{Brand} \sqsubseteq \texttt{Name}$
- if s is a hypernym or holonym of s' then $s \sqsupseteq s'$
  - $\texttt{Europe} \sqsupseteq \texttt{Greece}$
- if s and s' are synonyms then $s \equiv s'$
  - $\texttt{Quantity} \equiv \texttt{Amount}$
- if s and s' are antonyms $s \perp s'$
  - $\texttt{Microprocessor} \perp \texttt{PC Board}$

Basic idea: Graph isomorphism

Definition

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic*, $G_1 \equiv G_2$ if there is a bijection $\sigma : V_1 \rightarrow V_2$ with $(v, w) \in E_1$ if and only if $(\sigma(v), \sigma(w)) \in E_2$ for all $v, w \in V$.



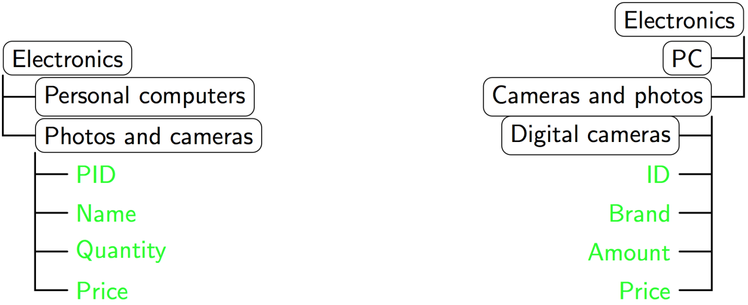
For ontologies: two entities are similar if they are matched by a graph isomorphism

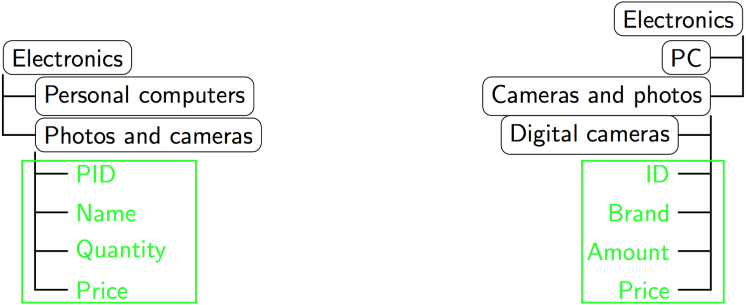
Graph-based techniques 2/3

Propagation of structural similarity:

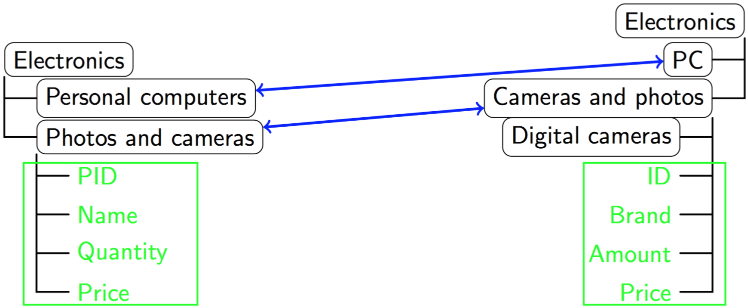
- Two non-leaf schema elements are structurally similar if their immediate children sets are similar, or
- Two non-leaf schema elements are structurally similar if their leaf sets are similar (even if their immediate children are not)

Graph-based techniques 3/3




Electronics
Personal computers
Photos and cameras
PID
Name
Quantity
Price
Electronics
PC
Cameras and photos
Digital cameras
ID
Brand
Amount
Price

Graph-based techniques 3/3



Logic-based techniques

Description logic based approach:

Ontology 1: $MicroCompany = Company \sqcap \exists_{\leq 5} hasEmployee.\top$
Ontology 2: $SME = Firm \sqcap \exists_{\leq 10} hasAssociate.\top$

Language-based matching technique discovers:

- $Company \equiv Firm$
- $hasAssociate \sqsubseteq hasEmployee$

Use DL-reasoning to obtain: $MicroCompany \sqsubseteq SME$

Wrap-up

Usually, ontology alignment tools use many of these approaches and aggregate individual confidence values.

The final alignment is selected by e. g.. . .

- . . . taking all correspondences from all matchers with a certain minimal confidence value
- . . . taking correspondences with maximum confidence that do not conflict with other correspondence (e. g. using optimization techniques for stable marriage problems)
- . . . taking preference among matchers into account (semantic matchers usually have higher priority than string-based matchers)

Outline

1. The Ontology Alignment Problem
2. Approaches to Ontology Alignment
3. Summary

Summary

- The ontology alignment problem
  - Given ontologies O_1 and O_2
  - Find *correspondences* $M = (e, e', R, n)$
- Approaches
  - String-based techniques
  - Language-based techniques
  - Techniques based on linguistic resources
  - Graph-based techniques
  - Logic-based techniques
- Systems
- Applications

Pointers to further reading

- http://ontologymatching.org
- Jerome Euzenat and Pavel Shvaiko. Ontology matching, Springer-Verlag, 978-3-540-49611-3. 2007
- Jerome Euzenat and Pavel Shvaiko. Ontology matching: state of the art and future challenges IEEE Transactions on Knowledge and Data Engineering, 2013.