Ontology Matching and Data Linking
An Introduction

Konstantin Todorov

todorov@lirmm.fr

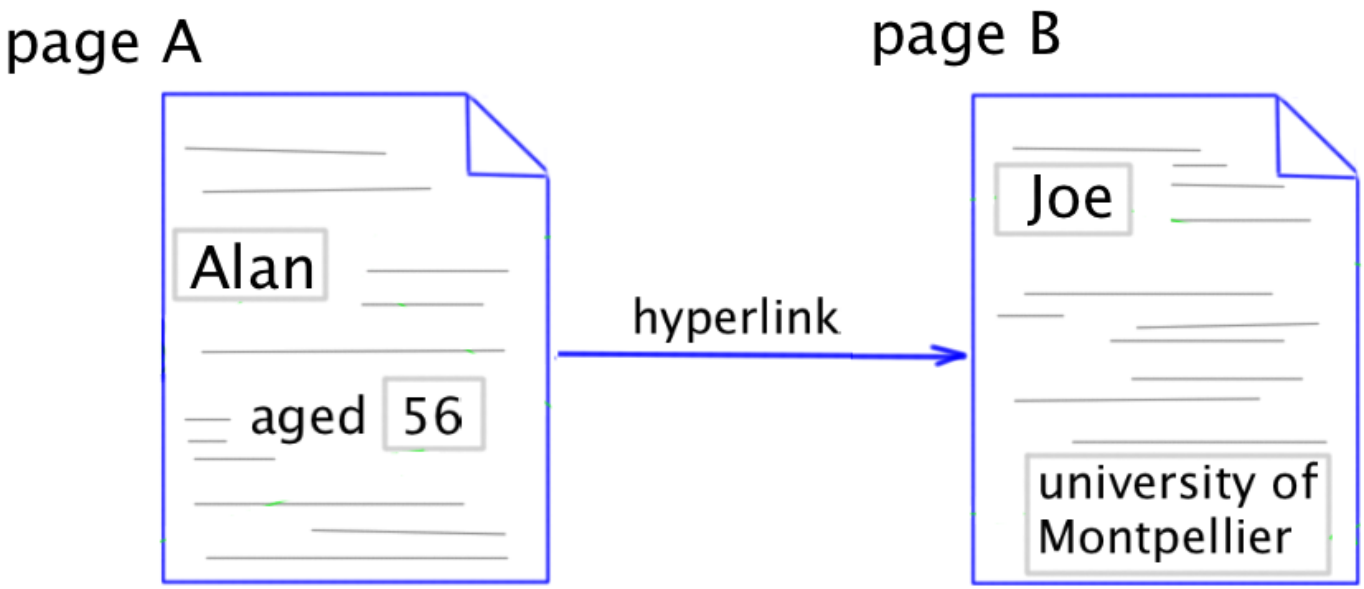
University of Montpellier

April 2018

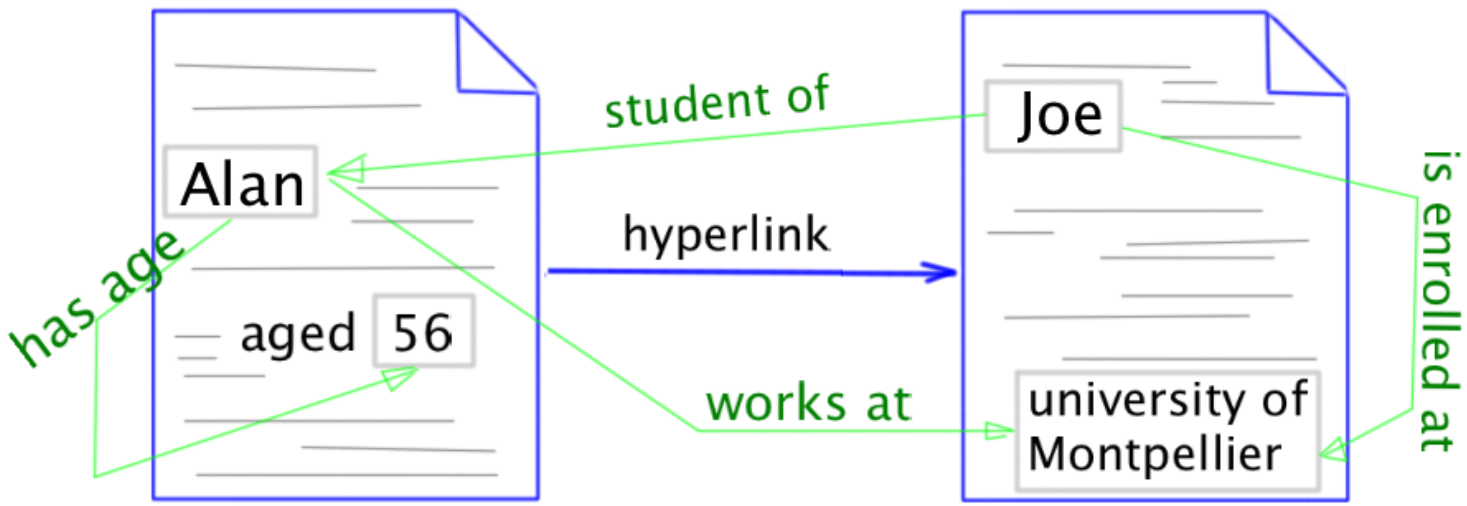
Outline

Outline

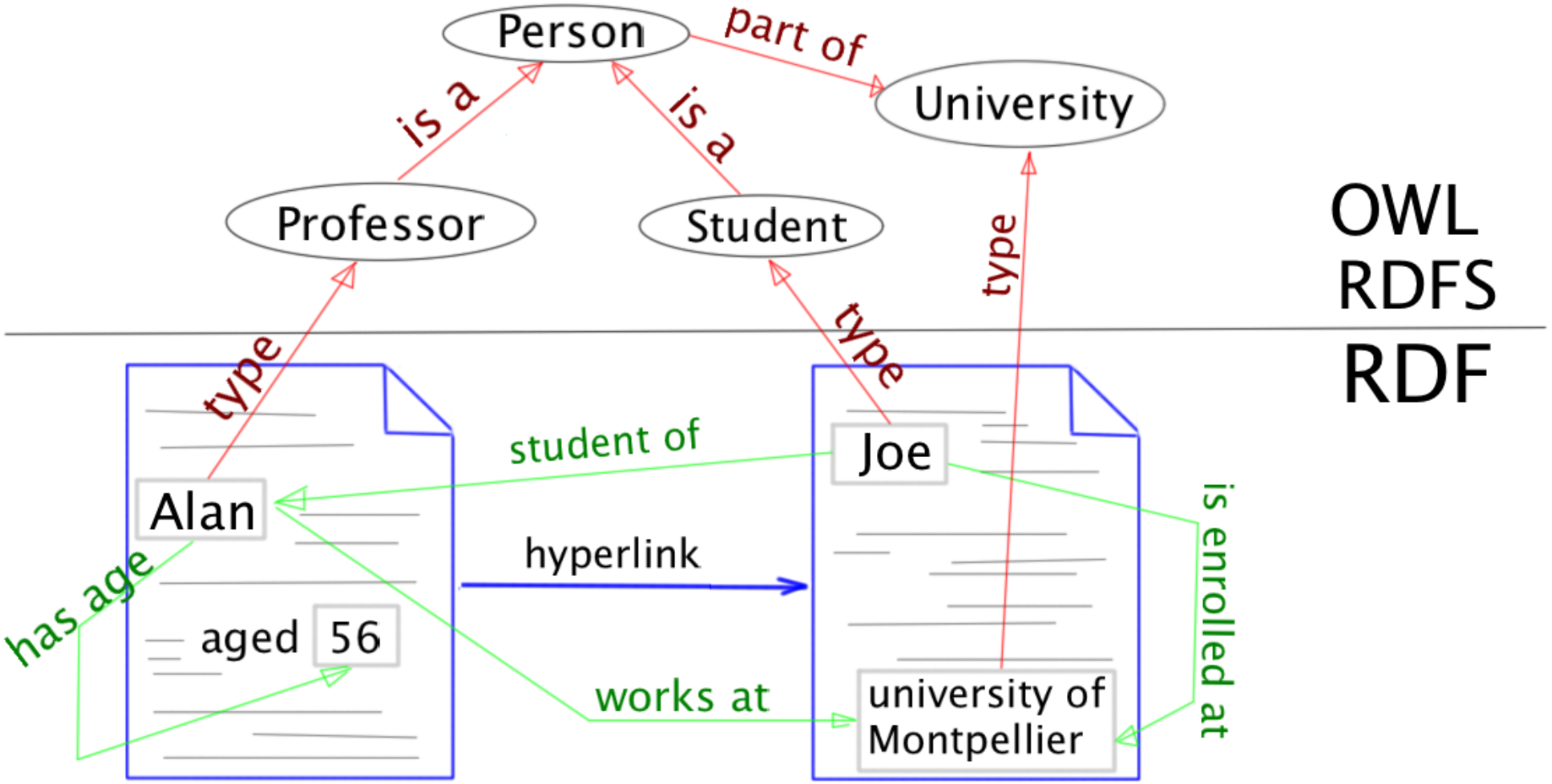
The web of documents



Linking Data



More semantics: the ontologies



Use of shared structured vocabularies, ontologies on the semantic web

- Use of terms from widely developed vocabularies
  - A set of vocabularies for describing common things like people, places or projects has emerged on the web of data.
- **Align heterogeneous vocabularies**
  - State that terms in different vocabularies are equivalent, or related: ontology matching
- **Link Data (the 4th principle of the web of data)**
  - State that resources across different datasets are equivalent, or related: data linking

A very common definition [3]:

"A **formal specification** of a **shared conceptualization** of a **domain** of interest."

- Formal specification: given in a formal language, thus executable
- Shared: regards a group of persons who agree on a given representation
- Conceptualization: it is about the concepts and how they relate to each other
- Domain: somewhere on the scale "application-driven – universally true" ("concrete – abstract")

A very common definition [3]:

"A **formal specification** of a **shared conceptualization** of a **domain** of interest."

- Formal specification: given in a formal language, thus executable
- Shared: regards a group of persons who agree on a given representation
- Conceptualization: it is about the concepts and how they relate to each other
- Domain: somewhere on the scale "application-driven – universally true" ("concrete – abstract")

A very common definition [3]:

"A **formal specification** of a **shared conceptualization** of a **domain** of interest."

- Formal specification: given in a formal language, thus executable
- Shared: regards a group of persons who agree on a given representation
- Conceptualization: it is about the concepts and how they relate to each other
- Domain: somewhere on the scale "application-driven – universally true" ("concrete – abstract")

A very common definition [3]:

"A **formal specification** of a **shared conceptualization** of a **domain** of interest."

- Formal specification: given in a formal language, thus executable
- Shared: regards a group of persons who agree on a given representation
- Conceptualization: it is about the concepts and how they relate to each other
- Domain: somewhere on the scale "application-driven – universally true" ("concrete – abstract")

A very common definition [3]:

"A **formal specification** of a **shared conceptualization** of a **domain** of interest."

- Formal specification: given in a formal language, thus executable
- Shared: regards a group of persons who agree on a given representation
- Conceptualization: it is about the concepts and how they relate to each other
- Domain: somewhere on the scale "application-driven – universally true" ("concrete – abstract")

Definition (Ontological Elements)

- C a finite set of concepts
- $is\_a \subseteq C \times C$ a partial order on concepts
- R a set of relations on C
- $l_L : C \rightarrow 2^{\Sigma_L^*}$ a function that assigns to each concept a set of labels from a set of labels Σ_L^* coming from some alphabet Σ_L specific for a language L

Definition (Ontology)

$O = (C, is\_a, R, I)$ forms an ontology.

Definition (Ontological Elements)

- C a finite set of concepts
- $is\_a \subseteq C \times C$ a partial order on concepts
- R a set of relations on C
- $l_L : C \rightarrow 2^{\Sigma_L^*}$ a function that assigns to each concept a set of labels from a set of labels Σ_L^* coming from some alphabet Σ_L specific for a language L

Definition (Ontology)

$O = (C, is\_a, R, I)$ forms an ontology.

Definition (Ontological Elements)

- C a finite set of concepts
- $is\_a \subseteq C \times C$ a partial order on concepts
- R a set of relations on C
- $l_L : C \rightarrow 2^{\Sigma_L^*}$ a function that assigns to each concept a set of labels from a set of labels Σ_L^* coming from some alphabet Σ_L specific for a language L

Definition (Ontology)

$O = (C, is\_a, R, I)$ forms an ontology.

Definition (Ontological Elements)

- C a finite set of concepts
- $is\_a \subseteq C \times C$ a partial order on concepts
- R a set of relations on C
- $l_L : C \rightarrow 2^{\Sigma_L^*}$ a function that assigns to each concept a set of labels from a set of labels Σ_L^* coming from some alphabet Σ_L specific for a language L

Definition (Ontology)

$O = (C, is\_a, R, I)$ forms an ontology.

Definition (Ontological Elements)

- C a finite set of concepts
- $is\_a \subseteq C \times C$ a partial order on concepts
- R a set of relations on C
- $l_L : C \rightarrow 2^{\Sigma_L^*}$ a function that assigns to each concept a set of labels from a set of labels Σ_L^* coming from some alphabet Σ_L specific for a language L

Definition (Ontology)

$O = (C, is\_a, R, I)$ forms an ontology.

Definition (Ontological Elements)

- C a finite set of concepts
- $is\_a \subseteq C \times C$ a partial order on concepts
- R a set of relations on C
- $l_L : C \rightarrow 2^{\Sigma_L^*}$ a function that assigns to each concept a set of labels from a set of labels Σ_L^* coming from some alphabet Σ_L specific for a language L

Definition (Ontology)

$O = (C, is\_a, R, l)$ forms an ontology.

Definition (Ontological Elements)

- C a finite set of concepts
- $is\_a \subseteq C \times C$ a partial order on concepts
- R a set of relations on C
- $l_L : C \rightarrow 2^{\Sigma_L^*}$ a function that assigns to each concept a set of labels from a set of labels Σ_L^* coming from some alphabet Σ_L specific for a language L

Definition (Ontology)

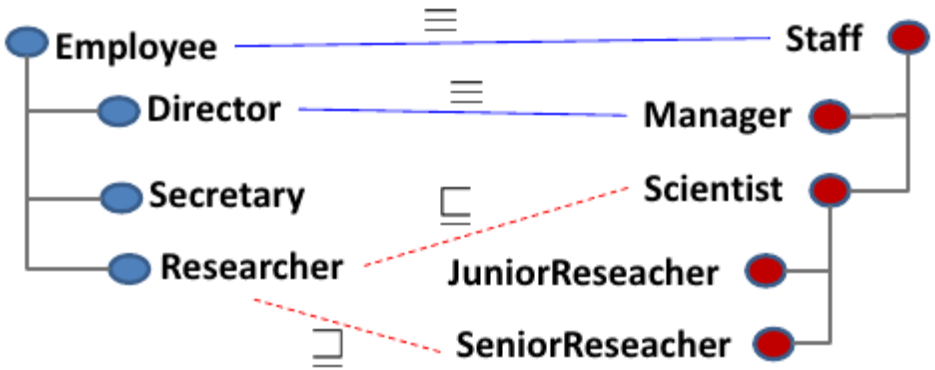
$O = (C, is\_a, R, l)$ forms an ontology.

- A set of concepts: EMPLOYEE, DIRECTOR, SECRETARY, RESEARCHER
- A set of labels: *"employee"*, *"director"*, *"secretary"*, *"researcher"*
- A subsumption relation (is_a) on the set of concepts

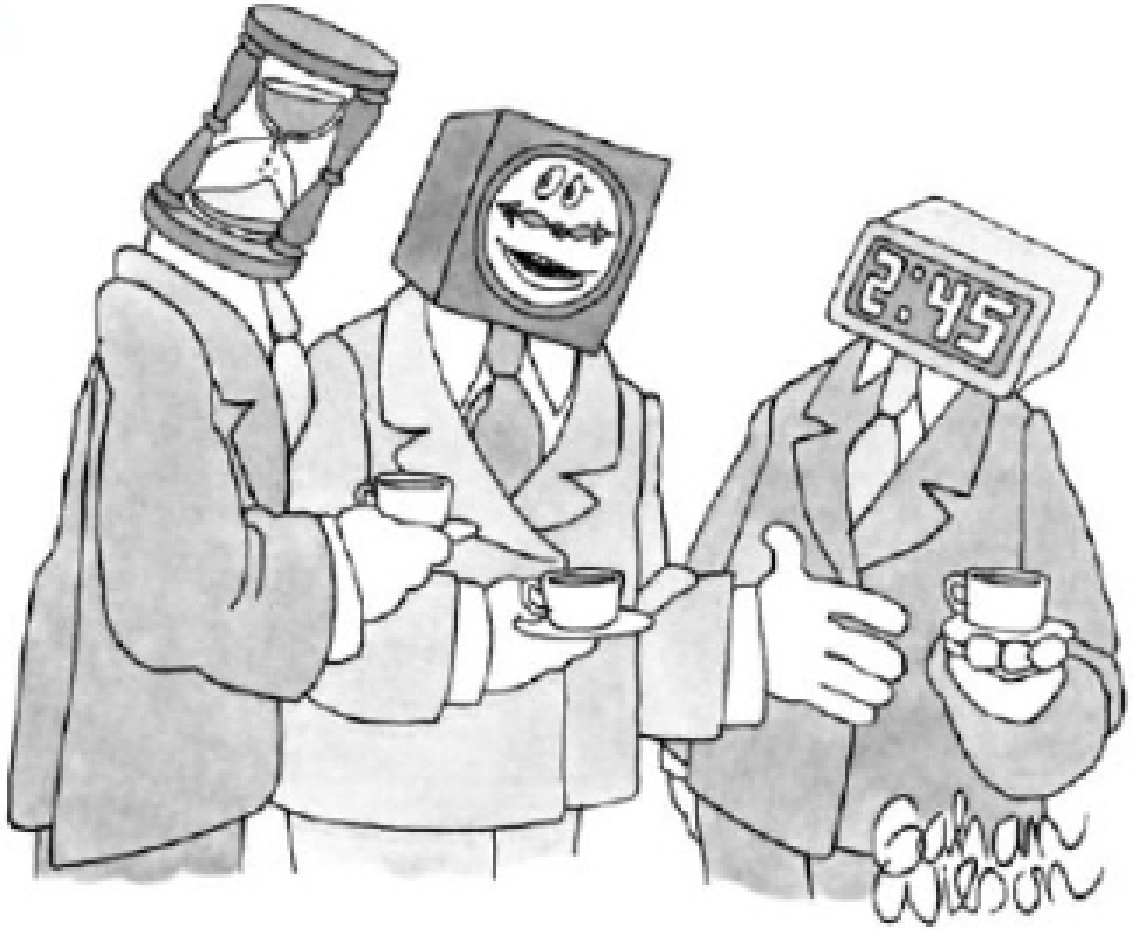
Note: often a set of labels is assigned to a single concept (e.g., a set of synonyms, translations).

Ontologies are created in a **decentralized**, strongly **human biased** manner.

Many ontologies describing the same domain of interest => ontology heterogeneity.



=> **Ontology Matching:** detect the semantic correspondences between the elements of two ontologies.



"Basically, we're all trying to say the same thing."

Borrowed by a tutorial by S. Staab and A. Hotho.

Outline

Heterogeneity Types

- Syntactic

  about the formal expression of ontologies

  example: OWL vs. SKOS

- Terminological

  about the choice of labels

  example: "director" vs. "manager"

- Structural / Conceptual

  about the relations between elements

  example: " is_a(director, person)" vs. " is_a(director, employee)"

  - granularity
  - coverage
  - scope

Heterogeneity Types

- Syntactic

  about the formal expression of ontologies

  example: OWL vs. SKOS

- Terminological

  about the choice of labels

  example: "director" vs. "manager"

- Structural / Conceptual

  about the relations between elements

  example: " is_a(director, person)" vs. " is_a(director, employee)"

  - granularity
  - coverage
  - scope

Heterogeneity Types

- Syntactic

  about the formal expression of ontologies

  example: OWL vs. SKOS

- Terminological

  about the choice of labels

  example: "director" vs. "manager"

- Structural / Conceptual

  about the relations between elements

  example: " is_a(director, person)" vs. " is_a(director, employee)"

  - granularity
  - coverage
  - scope

Ontology Alignment

The process of ontology matching results in an alignment.

An alignment:
a set of correspondances between the elements of two heterogeneous ontologies, derived by resolving the different heterogeneities that they manifest.

Similarity measures on element level or global level are applied for every heterogeneity type (e.g., terminological measures, etc.).

A function $\sigma : o \times o \rightarrow \mathbb{R}$ with some properties:

$$\forall x, y \in o, \quad \sigma(x, y) \geq 0$$
$$\forall x, y, z \in o, \quad \sigma(x, x) \geq \sigma(y, z)$$
$$\forall x, y \in o, \quad \sigma(x, y) = \sigma(y, x)$$

Ontology Alignment

The process of ontology matching results in an alignment.

An alignment:
a set of correspondances between the elements of two heterogeneous ontologies, derived by resolving the different heterogeneities that they manifest.

Similarity measures on element level or global level are applied for every heterogeneity type (e.g., terminological measures, etc.).

A function $\sigma : o \times o \rightarrow \mathbb{R}$ with some properties:

$$\forall x, y \in o, \quad \sigma(x, y) \geq 0$$
$$\forall x, y, z \in o, \quad \sigma(x, x) \geq \sigma(y, z)$$
$$\forall x, y \in o, \quad \sigma(x, y) = \sigma(y, x)$$

Ontology Alignment

The process of ontology matching results in an alignment.

An alignment:
a set of correspondances between the elements of two heterogeneous ontologies, derived by resolving the different heterogeneities that they manifest.

Similarity measures on element level or global level are applied for every heterogeneity type (e.g., terminological measures, etc.).

A function $\sigma : o \times o \rightarrow \mathbb{R}$ with some properties:

$$\begin{aligned} \forall x, y \in o, \quad & \sigma(x, y) \geq 0 \\ \forall x, y, z \in o, \quad & \sigma(x, x) \geq \sigma(y, z) \\ \forall x, y \in o, \quad & \sigma(x, y) = \sigma(y, x) \end{aligned}$$

Ontology Alignment

The similarity measure acts like a confidence value for each pair of concepts.

For a pair of concepts, $c \in O$ and $c' \in O'$, a mapping is defined as $(c, c', \sigma(c, c'))$, where σ is the confidence value of the mapping.

Mapping selection:
Many algorithms use that value to filter out unlikely mapping candidates –> thresholding with respect to σ.

For example: Keep only pairs of concepts with a confidence value higher than 0.55, e.g., only matches $(c, c', \sigma(c, c'))$ such that $\sigma(c, c') > 0.55$.

Ontology Alignment

Further filters: semantic verification (not a subject of this course...)

Ontology Alignment

To produce an alignment, several measures and filters are combined in a common matching algorithm.

The performance of the produced algorithm has to be evaluated.

Performance with respect to what?

- A standard ground-truth approach (just like in information retrieval).
- Qualitative measures (Precision, Recall, F-Measure)
- Benchmark? The Ontology Alignment Evaluation Initiative OAEI[1].

[1] http://oaei.ontologymatching.org

Ontology Alignment

To produce an alignment, several measures and filters are combined in a common matching algorithm.

The performance of the produced algorithm has to be evaluated.

Performance with respect to what?

- A standard ground-truth approach (just like in information retrieval).
- Qualitative measures (Precision, Recall, F-Measure)
- Benchmark? The Ontology Alignment Evaluation Initiative OAEI[1].

[1] http://oaei.ontologymatching.org

Ontology Matching

Matching and Evaluation Framework

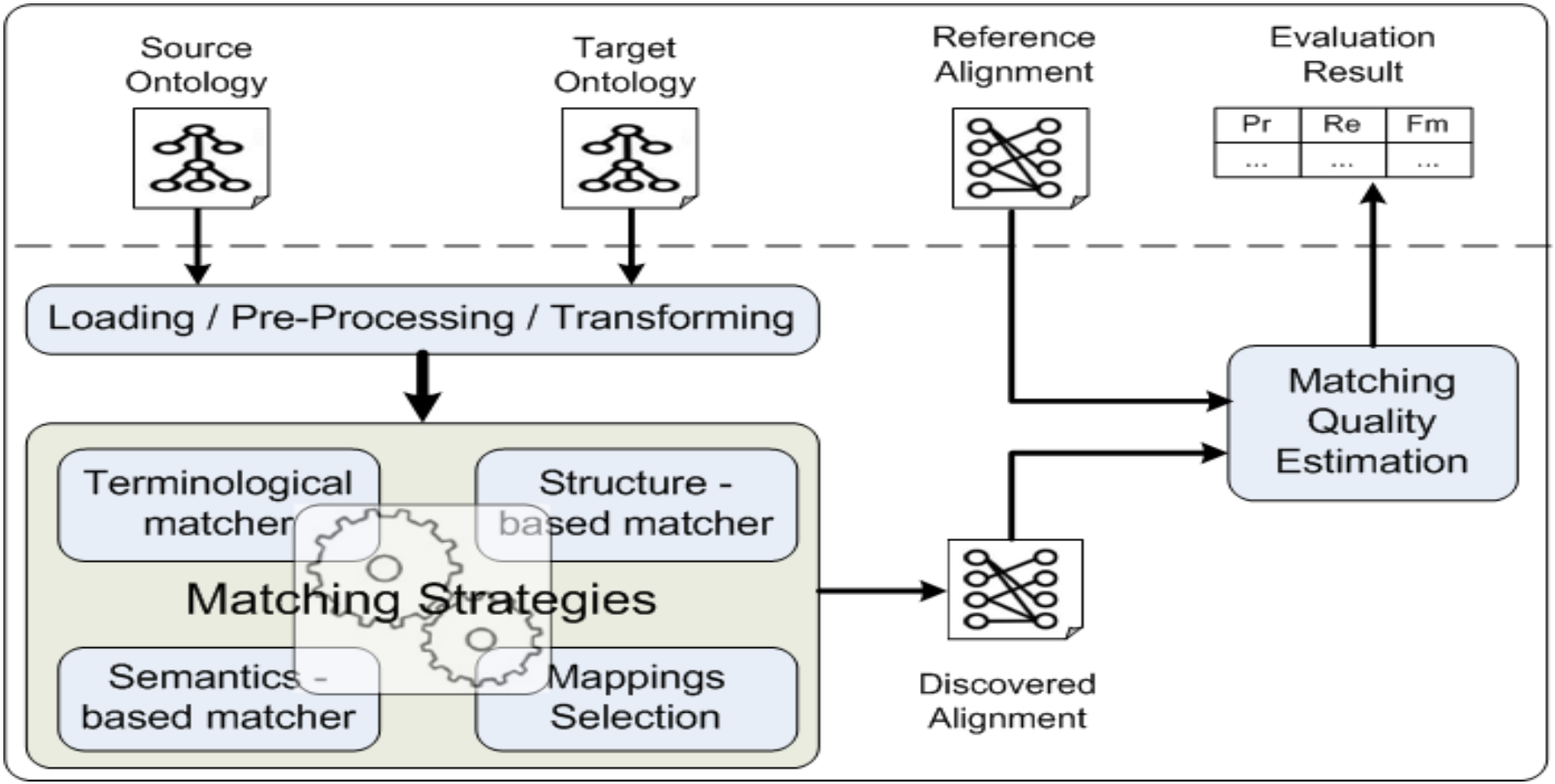


Figure: Ontology Matching: System Architecture and Evaluation Scenario

For more info: [5].

Outline

Outline

Terminological Heterogeneity

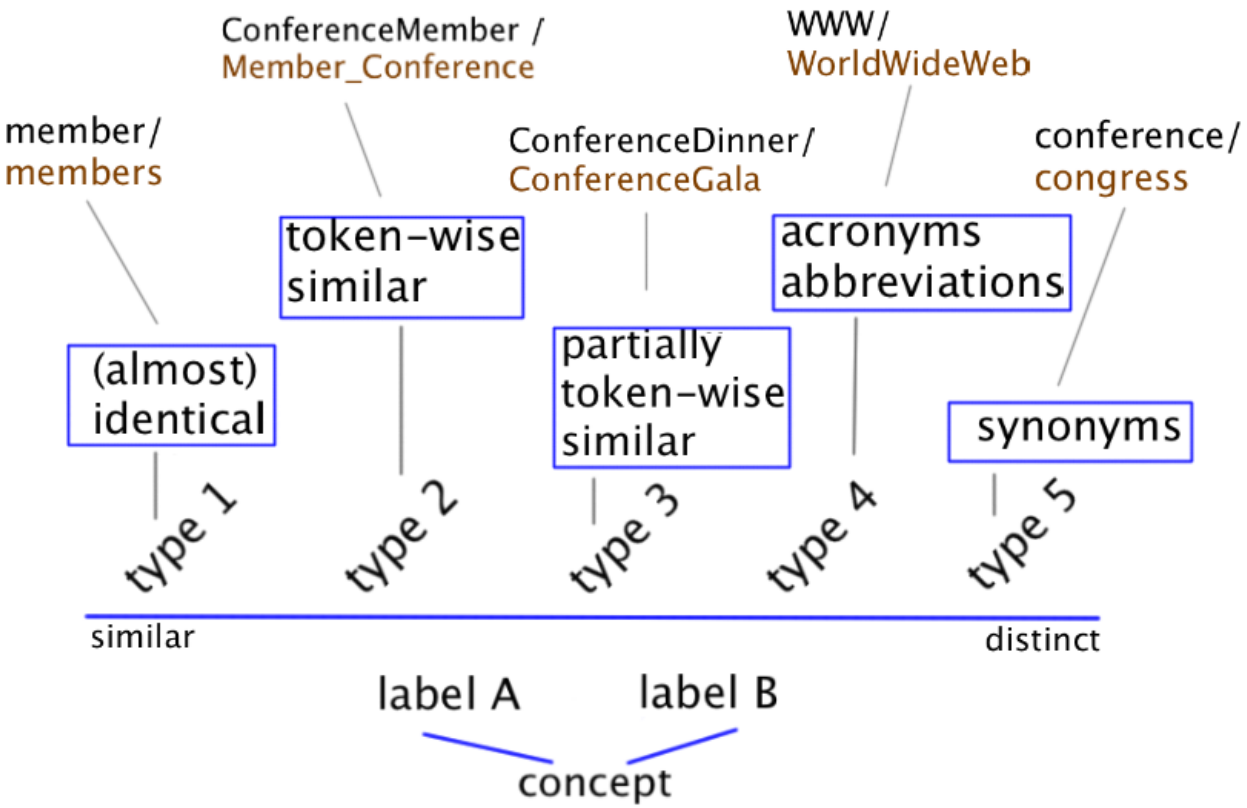
A Typology

Assumption:

A concept is identified by the meaning of its label(s). Therefore ->

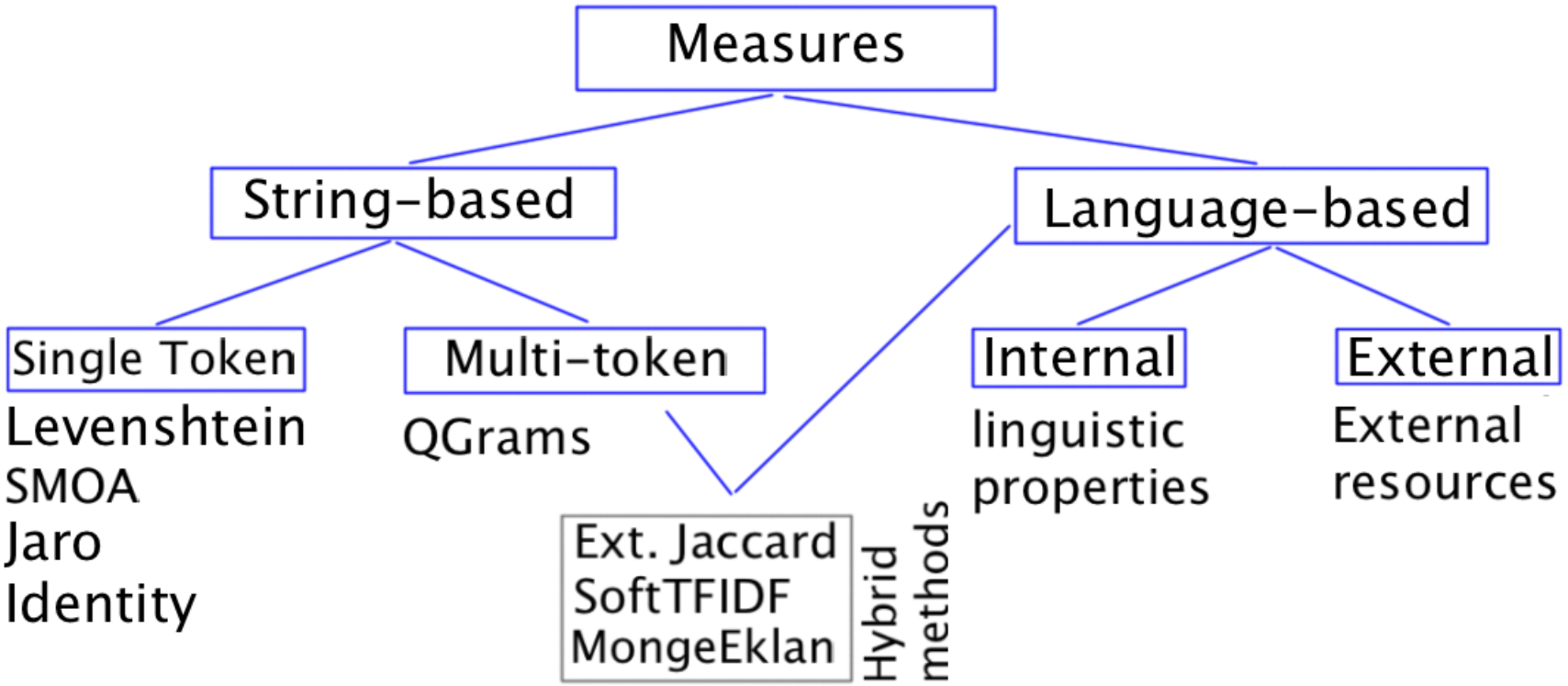
Terminological Heterogeneity:

Any difference in spelling between two terms or labels which are assumed to refer to the same concept [4].



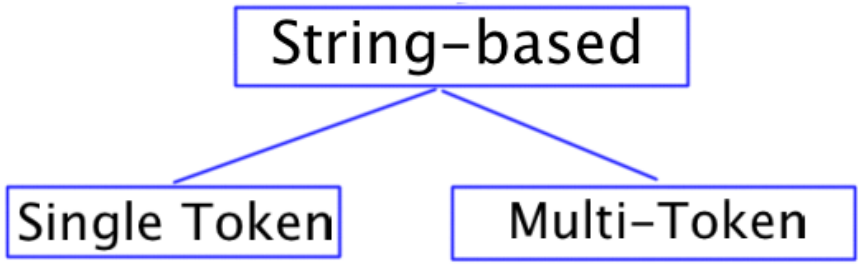
Terminological Heterogeneity

Similarity Measures



Terminological Heterogeneity

Discussion I



- Multi-token

  - can handle compound labels
  - are less sensitive to word-swaps ("ConferenceMember" vs. "MemberConference")
  - sometimes need external resources to assign weights to the composing tokens (large corpus)

- Single-token

  - can handle one-token labels with tiny variations in spelling
  - often used inside of a token-based measure

Single-token measures

The simplest similarity measure: the identity.

Definition (Identity Similarity Measure)

Let s_1 and s_2 be two single-token concept labels. We define

$$\sigma_{id}(s_1, s_2) = \begin{cases} 1, \text{if } s_1 = s_2 \\ 0, otherwise. \end{cases}$$

In other words, s_1 and s_2 are the same string of characters.

Example: s_1 =Conference, s_2 = Conference, $s_1 \in O$, $s_2 \in O'$

Single-token measures

Edit-distance: the minimal cost of operations to be applied on an object A in order to transform it into the object B.

Definition (Levenshtein distance.)

Let s_1 and s_2 be two concept labels. The Levenshtein distance, denoted

$$\delta_{Lev}(s_1, s_2),$$

is the minimal number of insertions, deletions and substitutions of characters required to transform s_1 into s_2. It follows that the cost of each operation is equal to 1.

Example: $\delta_{Lev}(\mathit{Conferences}, \mathit{Conferenc}) = 2$.
(We apply two times the deletion operation.)

Definition (Levenshtein Normalized Distance.)

Let s_1 and s_2 be two concept labels. The normalized Levenshtein distance is given by

$$\delta_{LevN} = \frac{\delta_{Lev}(s_1, s_2)}{max(|s_1|, |s_2|)},$$

where $|s_1|$ denotes the number of characters of the string s_1.

Example: $\delta_{LevN}(Conferences, Conferenc) = 2/11 = 0.1818$

Definition (Levenshtein Normalized Similarity.)

Let s_1 and s_2 be two concept labels. The normalized Levenshtein similarity is given by

$$\sigma_{LevN}(s_1, s_2) = 1 - \delta_{LevN}(s_1, s_2).$$

Example: $\sigma_{LevN}(Conferences, Conferenc) = 1 - 0.1818 = 0.8181$

Jaro Similarity

Definition (Jaro Similarity)

Let s_1 and s_2 be two concept labels. The Jaro similarity is given by

$$\sigma_{Jaro}(s_1, s_2) = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right),$$

where m is the number of common characters of s_1 and s_2, and t is half the number of transpositions - the common elements that occur in different order.

Jaro Similarity: an example

	M	A	R	T	H	A
M	1	0	0	0	0	0
A	0	1	0	0	0	0
R	0	0	1	0	0	0
H	0	0	0	0	1	0
T	0	0	0	1	0	0
A	0	0	0	0	0	1

m=6 (number of 1 in the matrix)

t = 2/2 (two elements in common, but not in the corresponding order)

|s1| = |s2| = 6

$$\sigma_{Jaro}(MARTHA, MAHRTA) = \frac{1}{3}\left(\frac{6}{6} + \frac{6}{6} + \frac{5}{6}\right) = 0.944$$

Taken from Wikipedia.

SMOA similarity: String Metric for Ontology Matching. Based on 2 elements:
(1) lengths of the common substrings,
(2) lengths of the remaining unmatched substrings

Definition (SMOA similarity.)

Let s_1 et s_2 be two strings.

$$\sigma_{SMOA}(s_1, s_2) = c(s_1, s_2) - d(s_1, s_2) + w(s_1, s_2),$$

where
– $c(s_1, s_2)$ is a function of the common substrings and contributes positively,
– $d(s_1, s_2)$ is a function of the unmatched remaining parts and contributes negatively.

Simple multi-token measures

Definition (NGrams similarity.)

Let s_1 and s_2 be two concept labels. We define the function $ngram(s, n)$ as the set of substrings of size n of the string s. On this basis, we define the NGram similarity measure as follows:

$$\sigma_{NG}(s_1, s_2) = \frac{|ngram(s_1, n) \cap ngram(s_2, n)|}{min(|s_1|, |s_2|) - n + 1}$$

Example *ngram*:

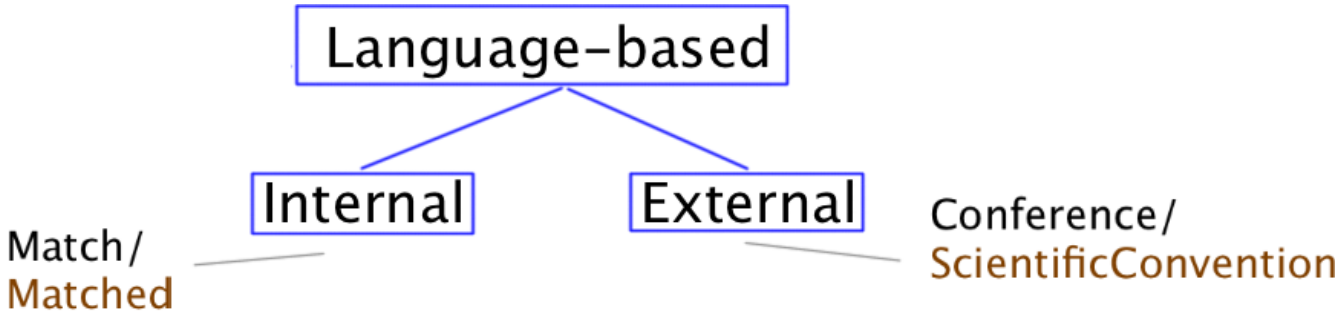
$n = 3$

–> set of trigrams for "article": {art, tri, tic, icl, cle}
–> set of tirgrams for "aricle": {ari, ric, icl, cle}.

$\sigma_{3G}(ARTICLE, ARICLE) = \frac{2}{6-3+1} = \frac{2}{4} = 0.5$

Terminological Heterogeneity

Discussion II



Sources of external information: dictionaries, thesauri, lexical databases (WordNet).

- Two common problems (for both internal and external measures)

  - dealing with single words and not compound ones ("PhDThesis" is not found in WN, although "PhD" and "Thesis" are)
  - typos or non-conventional abbreviations prevent from finding the words in dictionaries

Terminological Heterogeneity

Internal Language-based Methods

Linguistic normalizations:

- Tokenization: segmenting strings into sequences of tokens (terms)
  Example: "ConferenceOrganizer" –> {conference, organizer}
- Lemmatization: reduction to basic morphological forms
  Example: matched –> match
- Stopword pruning: use stop-word lists to eliminate noisy words
  Example: "Organizer_of_a_Conference" –> "Organizer_Conference"

Terminological Heterogeneity

External Language-based

Use external linguistic sources in order to compute similarity of labels.

What sources?

- Lexicons and dictionaries: contain word definitions
- Thesauri and lexical databases: contain relational information about terms, e.g., synonyms (article = paper), hypernyms ("author" is less specific than "article author")

  Example: WordNet is a lexical database where words are grouped in sets of synonyms (synsets) and semantic relations are defined between synsets (hypernymy and hyponymy, meronymy and holonymy).

Similarity measures

Definition (Synonymy similarity)

Let s_1 and s_2 be two terms and Σ be a synonym resource. We define

$$\sigma_{syn}(s_1, s_2) = \begin{cases} 1, \text{if } \Sigma(s_1) \cap \Sigma(s_2) \neq \emptyset \\ 0, \textit{otherwise}. \end{cases}$$

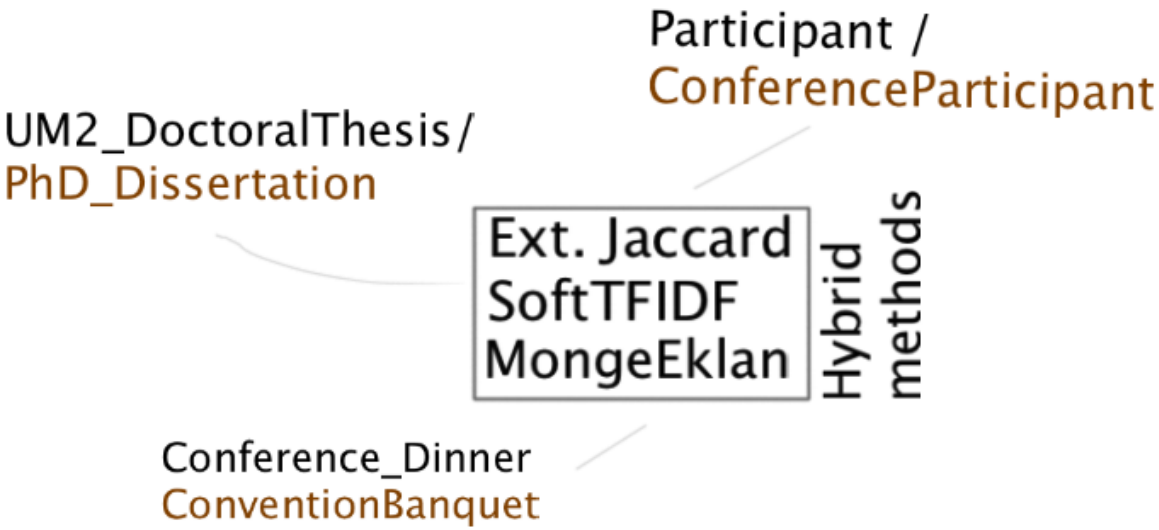
Definition (Co-synonymy similarity)

Let s_1 and s_2 be two terms and Σ be a synonym resource. We define

$$\sigma_{syn}(s_1, s_2) = \frac{|\Sigma(s_1) \cap \Sigma(s_2)|}{|\Sigma(s_1) \cup \Sigma(s_2)|}$$

Terminological Heterogeneity

Hybrid Measures Discussion III



- Limitations

- fail when the number of shared tokens is small
- require large corpus for weight computation
- MongeElkan and softTFIDF are asymmetric

Let s_1 and s_2 be two compound labels and let $\sigma_{token}(t_1, t_2)$ be a similarity measure that compares two tokens $t_1 \in \texttt{tokenize}(s_1)$ and $t_2 \in \texttt{tokenize}(s_2)$, where `tokenize` is a function returning the set of tokens of a compound label. Let θ_{token} be a similarity threshold. We define the following support functions:

$$
\begin{aligned}
\texttt{Shared}(s_1, s_2) &= \left\{(t_i, t_j) | t_i \in \texttt{tokenize}(s_1) \wedge t_j \in \texttt{tokenize}(s_2) : \sigma_{token}(t_i, t_j) \geq \theta_{token}\right\} \\
\texttt{Unique}(s_1) &= \left\{t_i | t_i \in \texttt{tokenize}(s_1) \wedge t_j \in \texttt{tokenize}(s_2) : (t_i, t_j) \notin \texttt{Shared}(s_1, s_2)\right\} \\
\texttt{Unique}(s_2) &= \left\{t_j | t_j \in \texttt{tokenize}(s_2) \wedge t_i \in \texttt{tokenize}(s_1) : (t_i, t_j) \notin \texttt{Shared}(s_1, s_2)\right\}
\end{aligned}
$$

Extended Jaccard

The **Jaccard** similarity is extended by introducing a weight function w for matching and non-matching tokens. Tokens in a pair of `Shared` get a weight corresponding to their similarity, whereas tokens in `Unique` get a weight equal to 1.0.

Definition (Extended Jaccard Similarity)

$$ExtendedJaccard(s_1, s_2) = \frac{\sum_{(t_i,t_j)\in Shared(s_1,s_2)} w(t_i,t_j)}{\sum_{(t_i,t_j)\in Shared(s_1,s_2)} w(t_i,t_j) + \sum_{(t_i)\in Unique(s_1)} w(t_i) + \sum_{(t_j)\in Unique(s_2)} w(t_j)} \quad (1)$$

Monge-Elkan: two labels are similar if their tokens are one by one similar. A token t_i from s_1 is matched to the token t_j in s_2 that has the maximum similarity to t_i. These maximum similarity scores obtained for every token of s_1 are then summed up, and the sum is normalized by the number of tokens in s_1. The similarity is defined as follows:

Definition (Monge-Elkan Similarity)

$$\sigma_{ME}(s_1, s_2) = \frac{1}{|tokenize(s_1)|} \sum_{i=1}^{|tokenize(s_1)|} max \{\sigma_{token}(t_i, t_j)\}_{j=1}^{|tokenize(s_2)|}$$

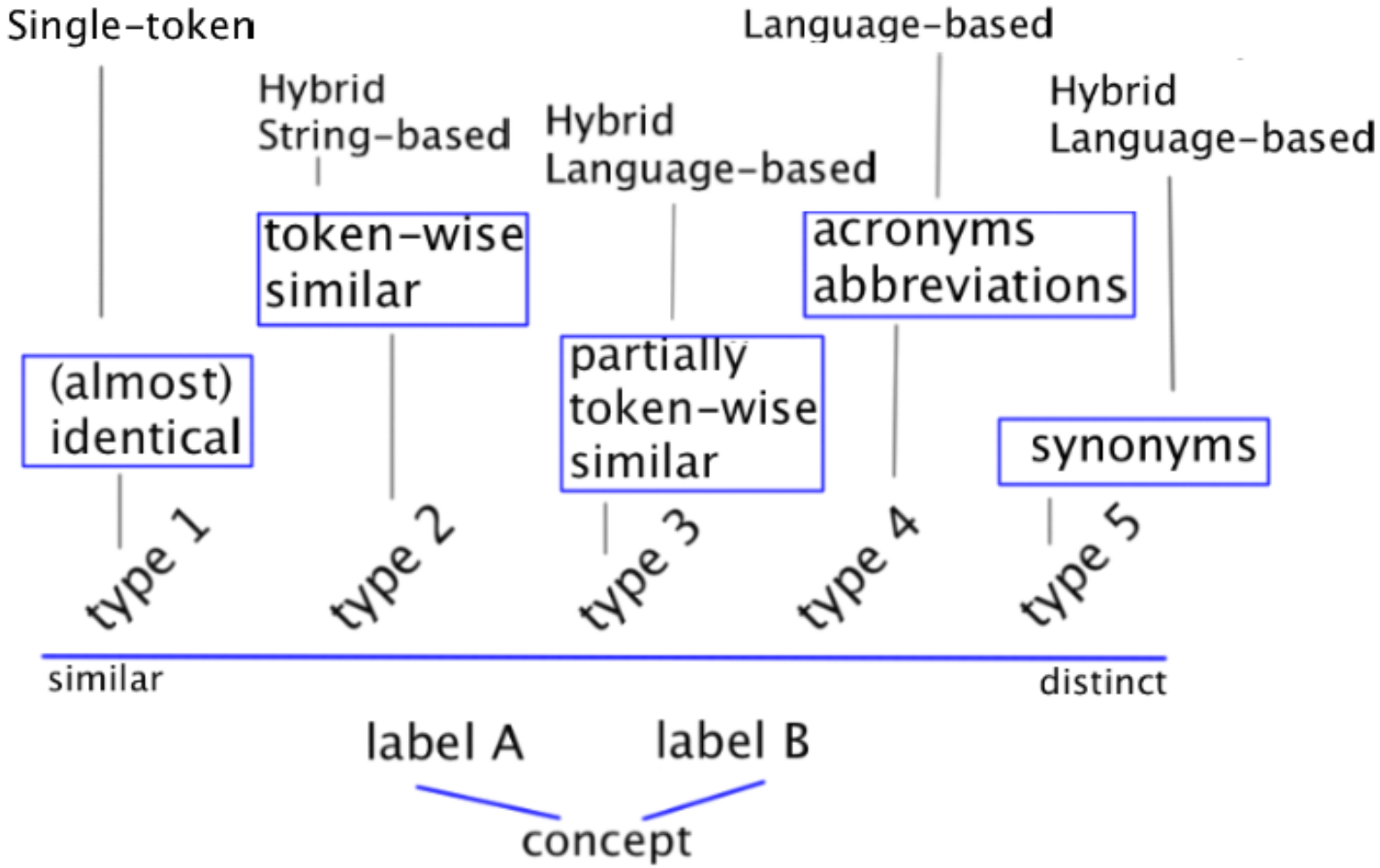
Asymmetry: the similarity score between "UM2_DoctoralThesis" and "PhdDissertation" is 0.57, whereas, the similarity score between "PhdDissertation" and "UM2_DoctoralThesis" is 0.85. To make this measure symmetric, we take the average of the similarity scores:

Definition (Monge-Elkan Symmetric Similarity)

$$\sigma_{symME}(s_1, s_2) = \frac{\sigma_{ME}(s_1, s_2) + \sigma_{ME}(s_2, s_1)}{2}$$

Terminological Heterogeneity

Measures and Heterogeneity Types



Outline

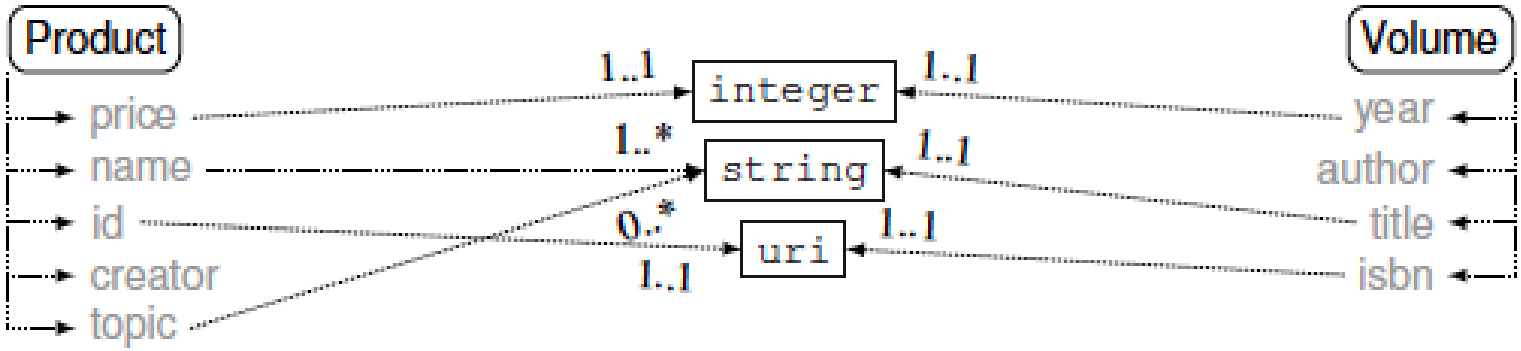
Structural Matchers

Internal methods

Compute similarity based on the internal structure of elements (e.g., classes)

- their properties
- range
- cardinalities, etc

Usually combined with terminological techniques



Taken from [1].

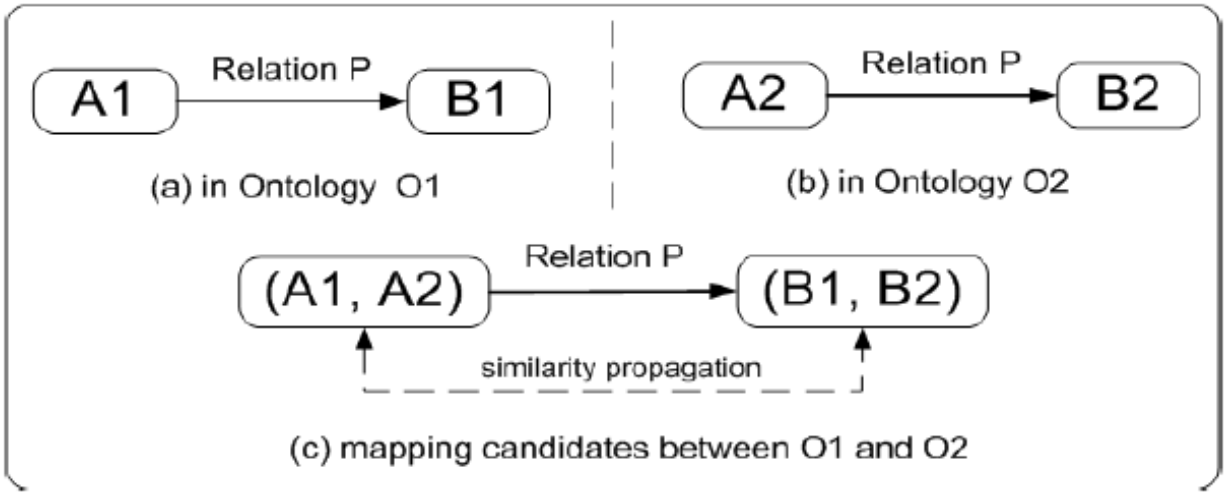
Structural Matchers

External (relational) methods

Consider the relations of concepts to other concepts. Rely on already discovered similarities.

- Standard methods
  - exploring standard structural relations between entities within the ontologies:

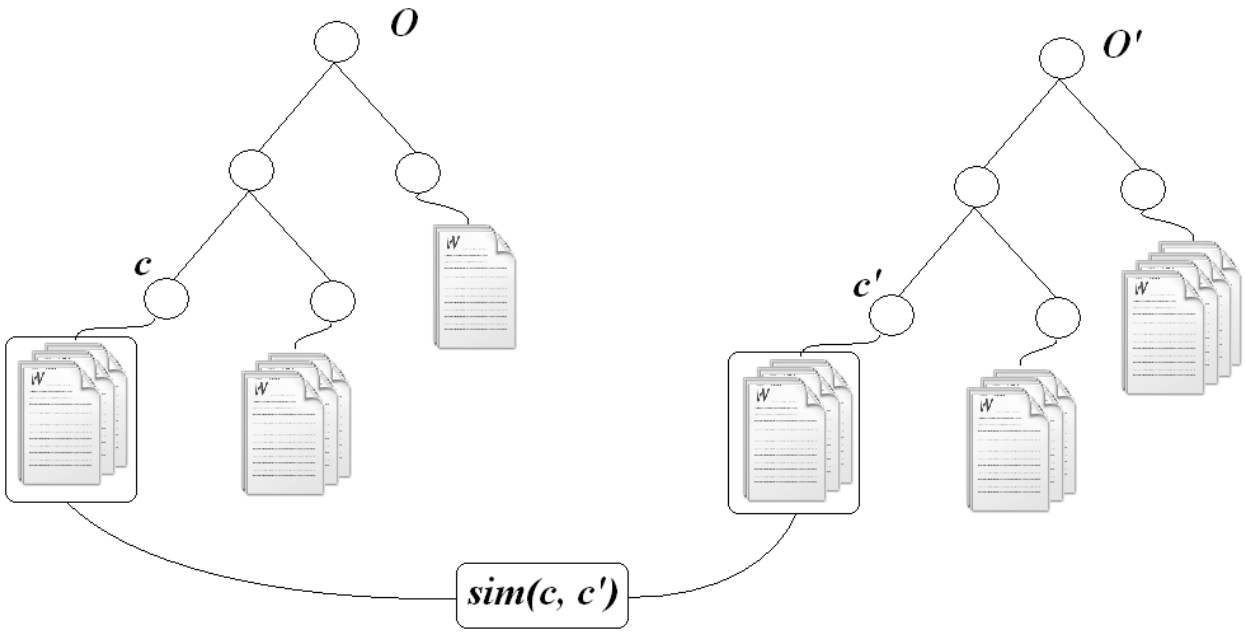
    descendants, ancestors, leaves, adjacent, etc.
- Similarity Propagation



Outline

Ontology Matching

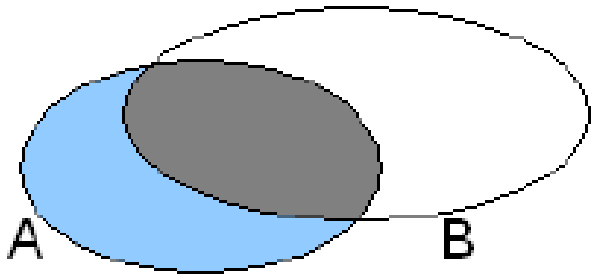
Instance-based concept similarity



The similarity of two cross-ontology concepts is assessed by the help of the instances of these concepts

-> Many possible measures.

Intersection of class instance sets

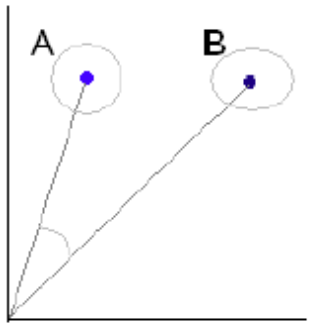


-> Same instances need to be found in both ontologies.

The cosine of the prototypes

$$sim(A,B) = s\left(\frac{1}{|A|}\sum_{j=1}^{|A|}\mathbf{i}_j^A, \frac{1}{|B|}\sum_{k=1}^{|B|}\mathbf{i}_k^B\right),$$

with $s(x,y)$ the cosine similarity of x and y.

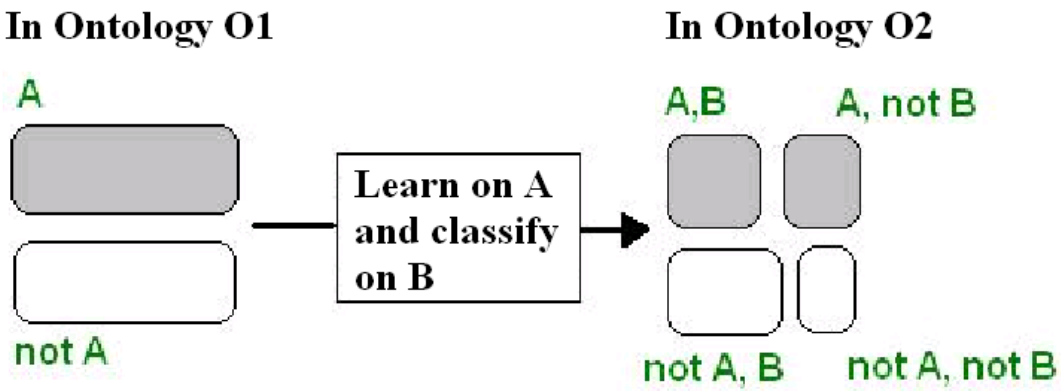


-> Flattening class structure

The Jaccard coefficient

$$Jacc(A,B) = Pr(A \cap B)/Pr(A \cup B).$$

Machine learning is used to estimate the joint probabilities.

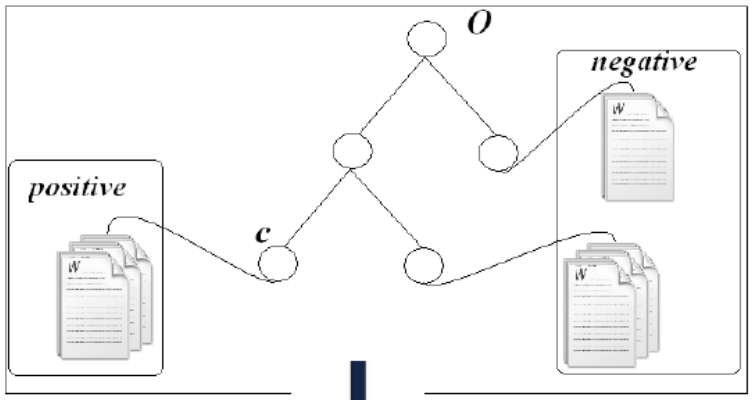


-> Insensitive to instance set intersection size

Ontology Matching

Instance-based concept similarity

Variable selection based measure



$$L(c) = (s_1^c, s_2^c, ..., s_n^c)$$

-> Time complexity is high

Outline

Ontology Matching

Matching and Evaluation Framework

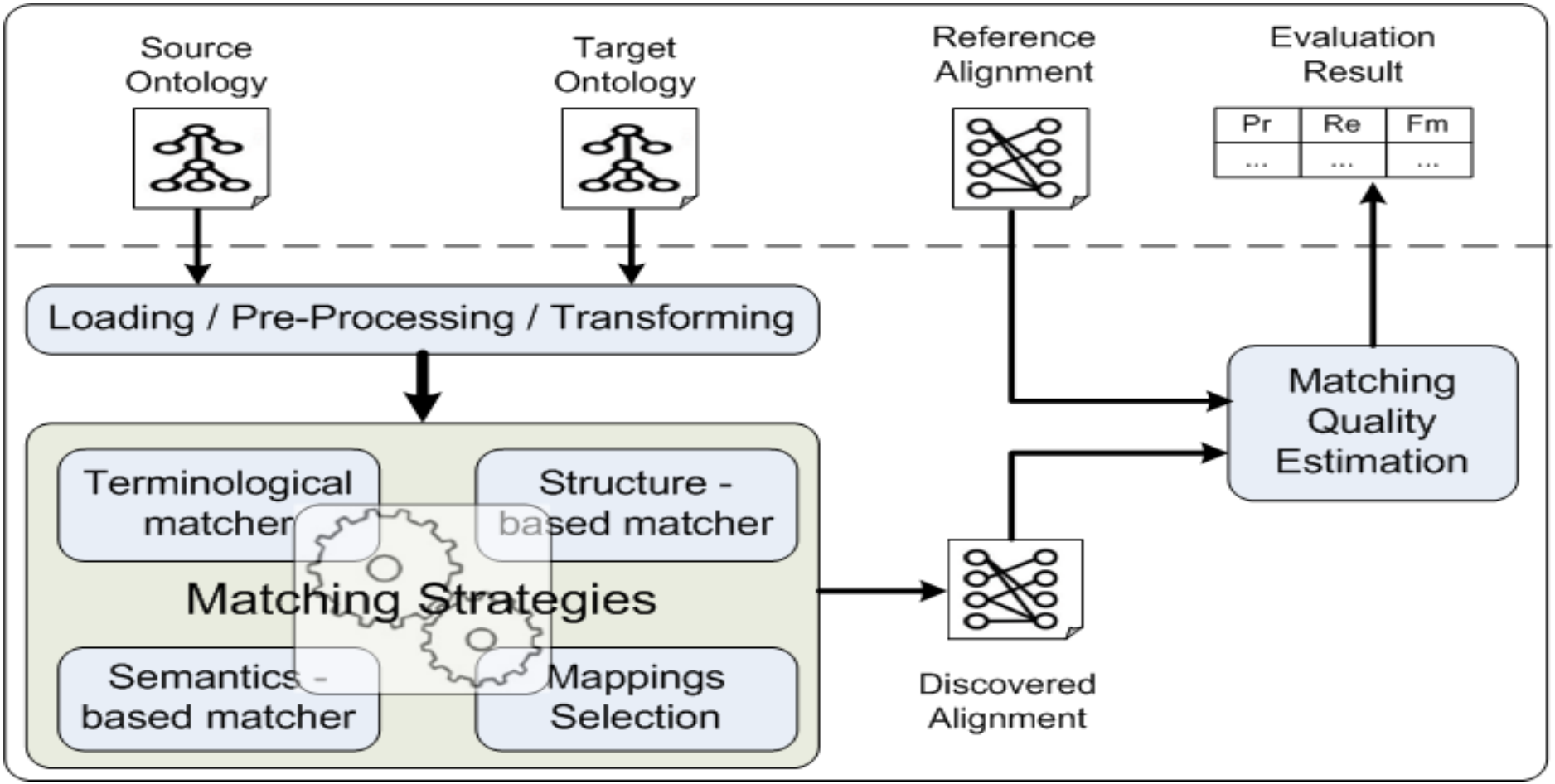
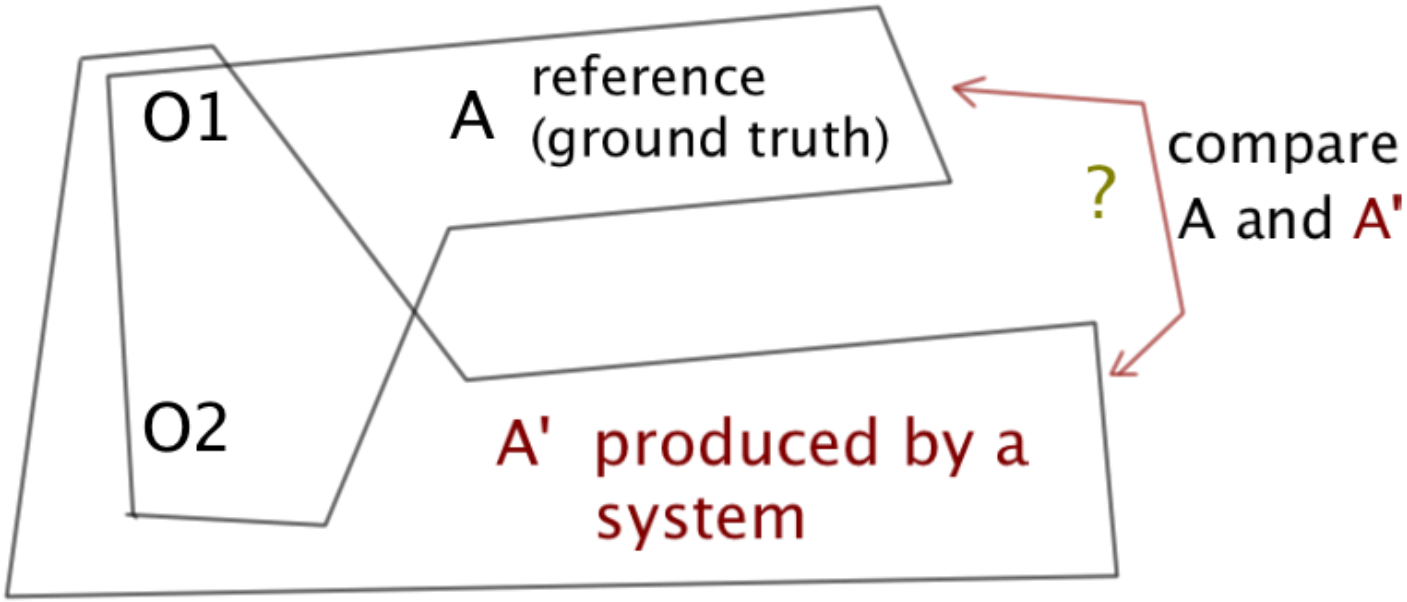


Figure: Ontology Matching: System Architecture and Evaluation Scenario

A standard benchmark approach, an annual evaluation campaign OAEI[2]

- Ground truth: a set of pairs of ontologies with alignments.
- Test: align two ontologies from this set and compare the produced alignment with the ground truth.



[2] http://oaei.ontologymatching.org

On n tests, we compute:

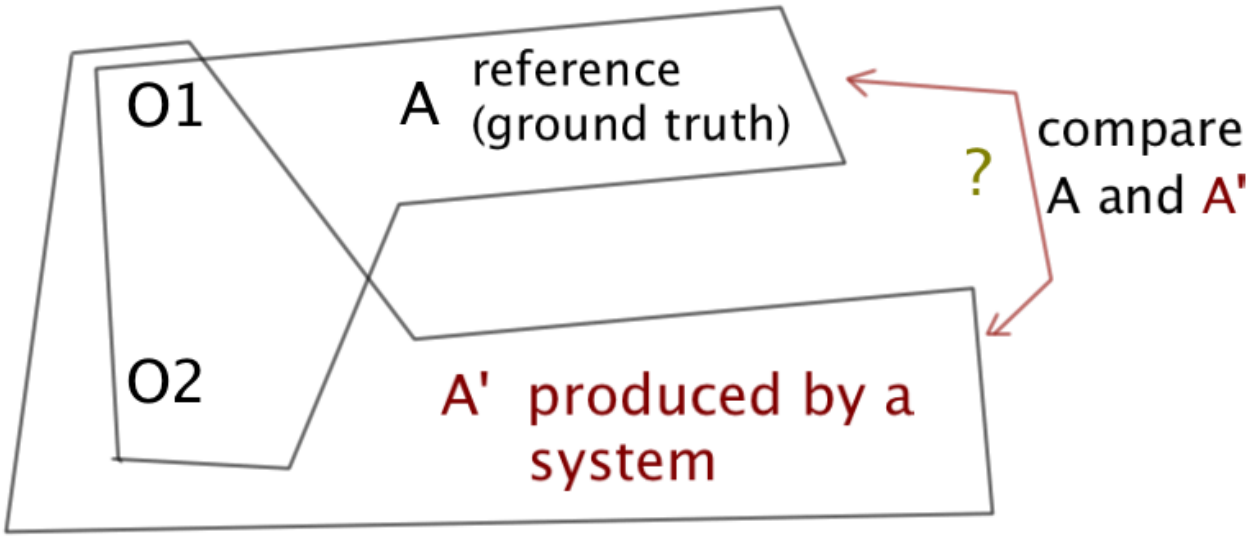
$$H(p) = \frac{\sum_{i=1}^{n} |C_i|}{\sum_{i=1}^{n} |A_i|}; \quad H(r) = \frac{\sum_{i=1}^{n} |C_i|}{\sum_{i=1}^{n} |R_i|}; \quad H(fm) = \frac{2 * H(p) * H(r)}{H(p) + H(r)}.$$

For the ith test:

- $|A_i|$ – the total number of mappings discovered by a matching system,
- $|C_i|$ – the number of correct mappings,
- $|R_i|$ – the number of reference mappings (expert).

Evaluation

Precision and Recall: Revision

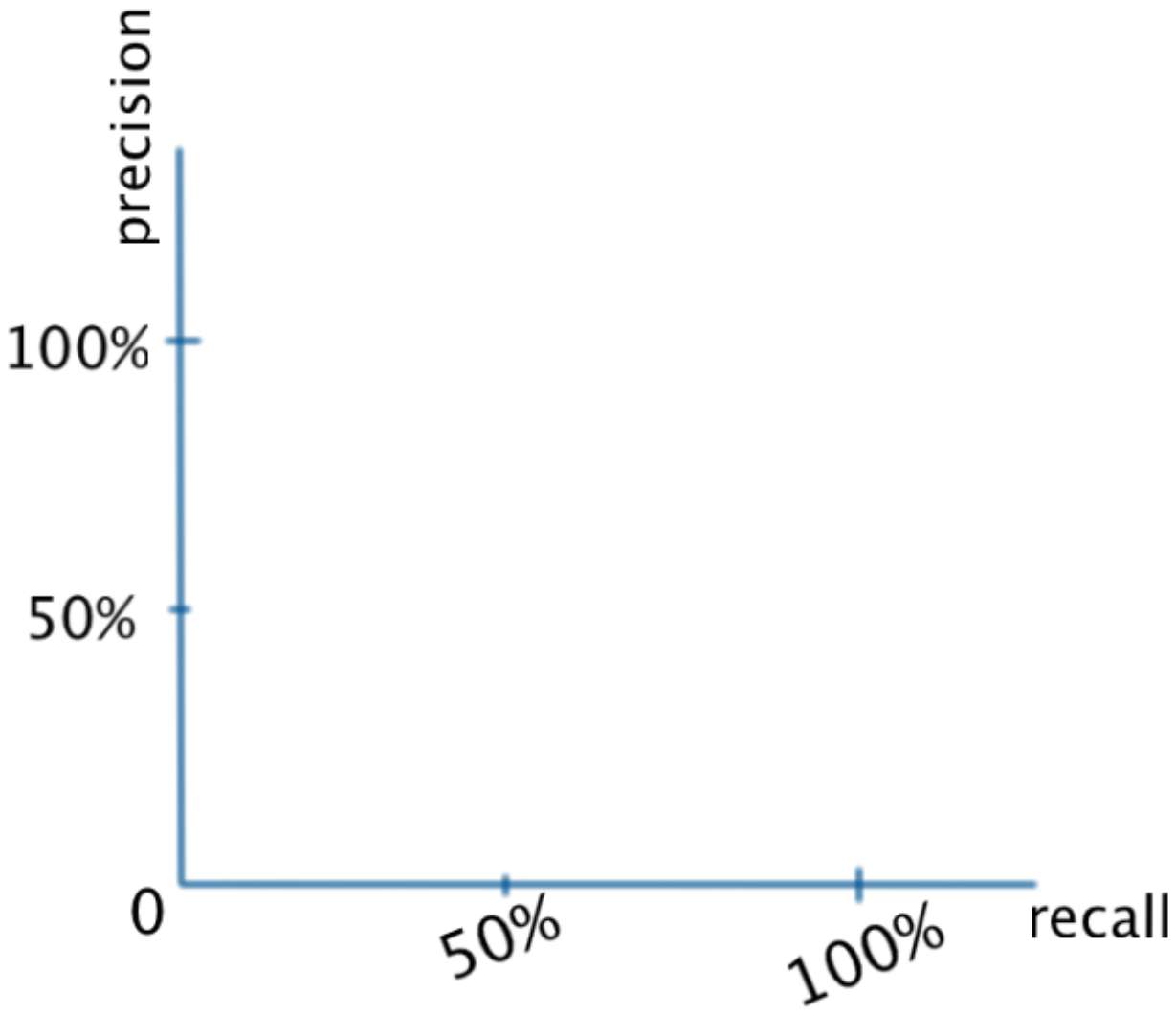


Precision and Recall

$$P = \frac{true\_Matches\_found}{all\_found}, \qquad R = \frac{true\_Matches\_found}{all\_true\_Matches}.$$

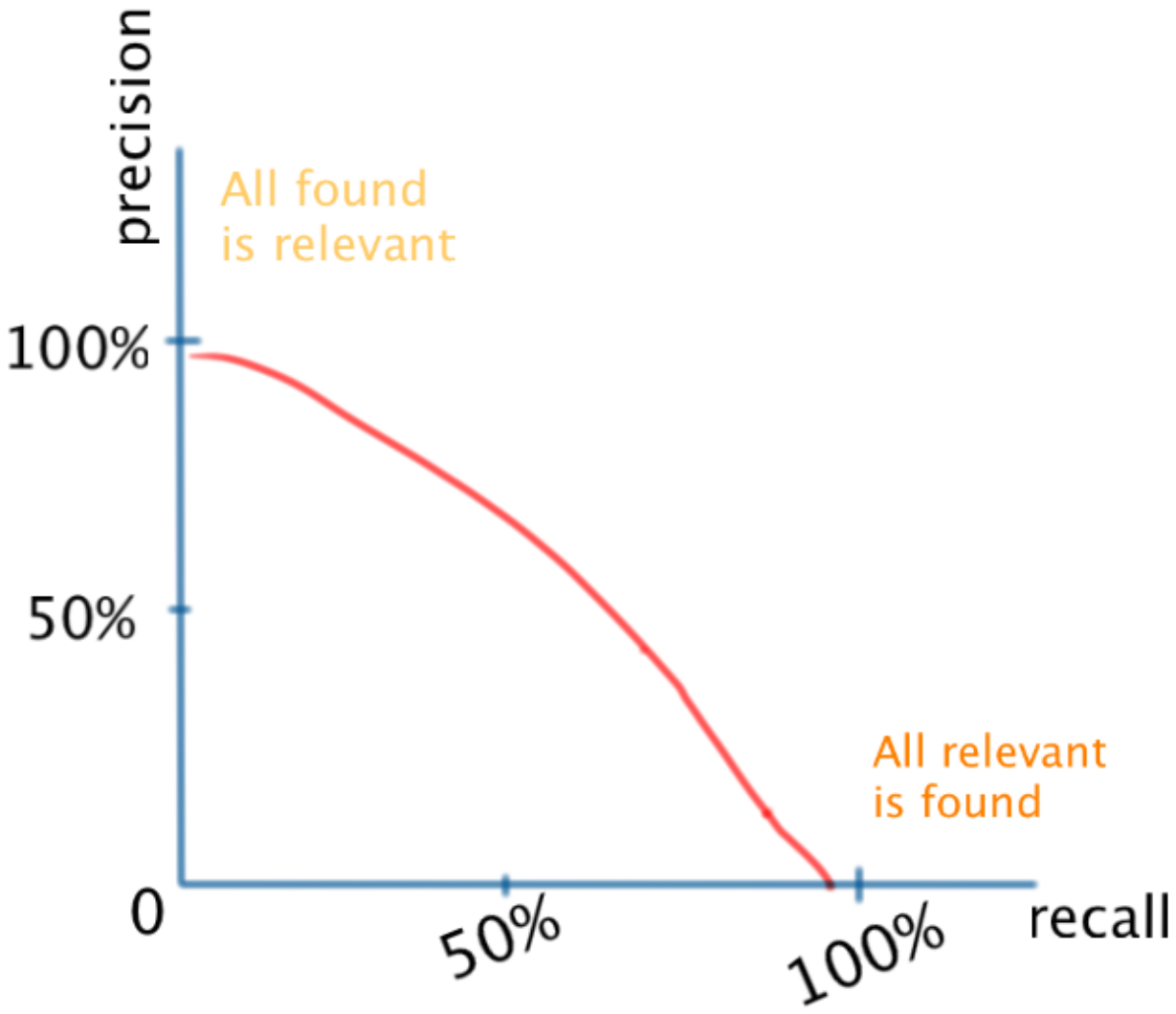
Evaluation

Precision and Recall: Revisionl



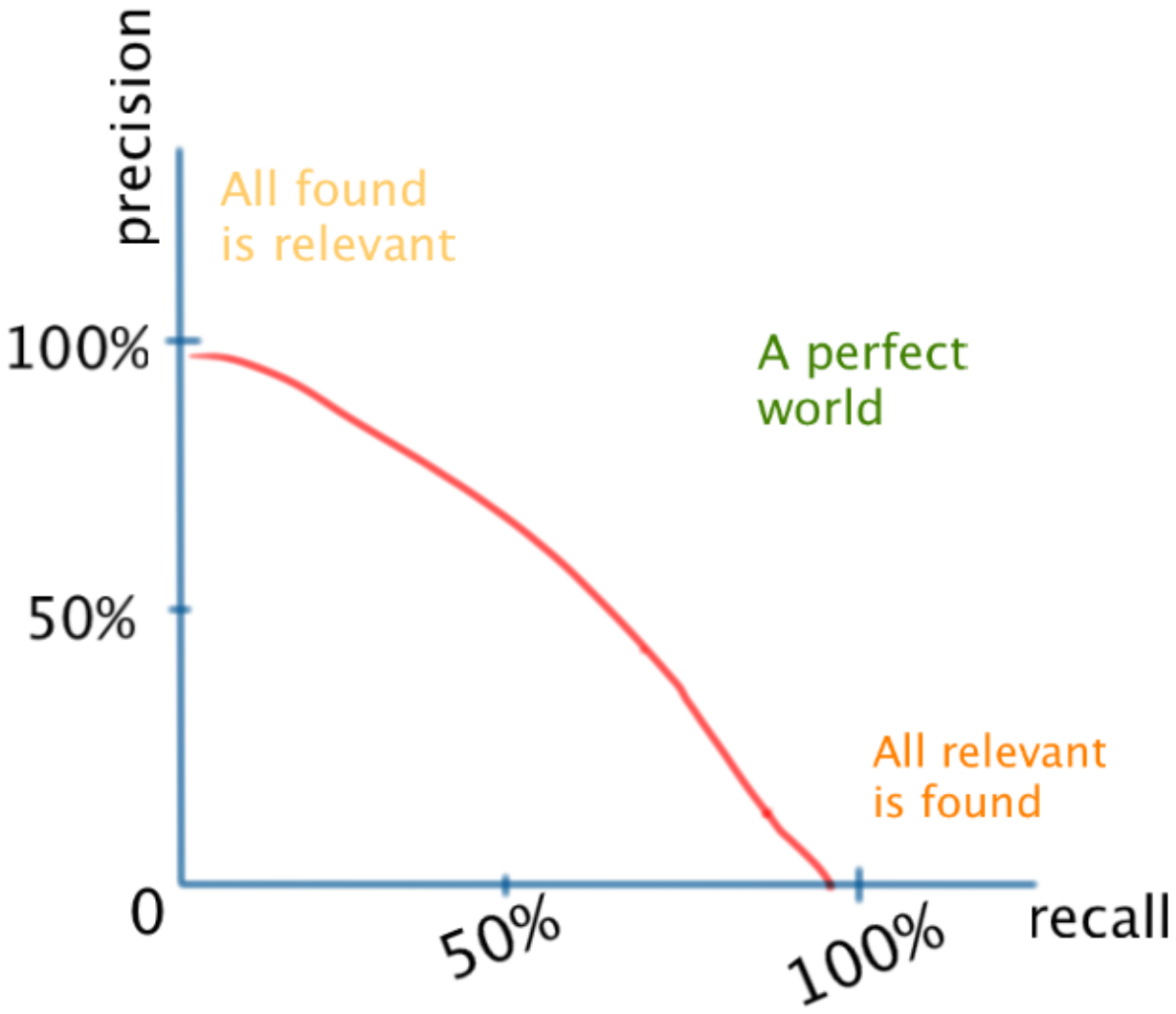
Evaluation

Precision and Recall: Revision



Evaluation

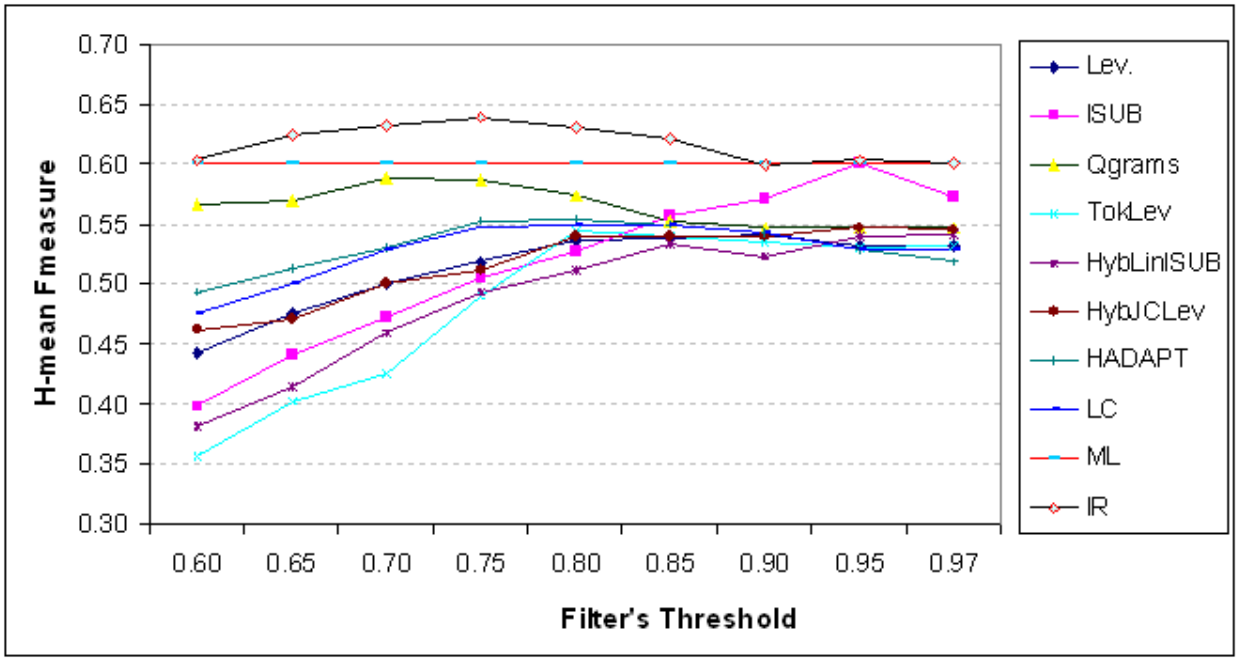
Precision and Recall: Revision



Evaluation

F-measure

Usually, a unique measure is used: the F-measure. The results are presented as the F-measure in a function of the mapping threshold:



The number of correctly aligned concepts changes with respect to the mapping selection threshold (precision and recall change respectively).

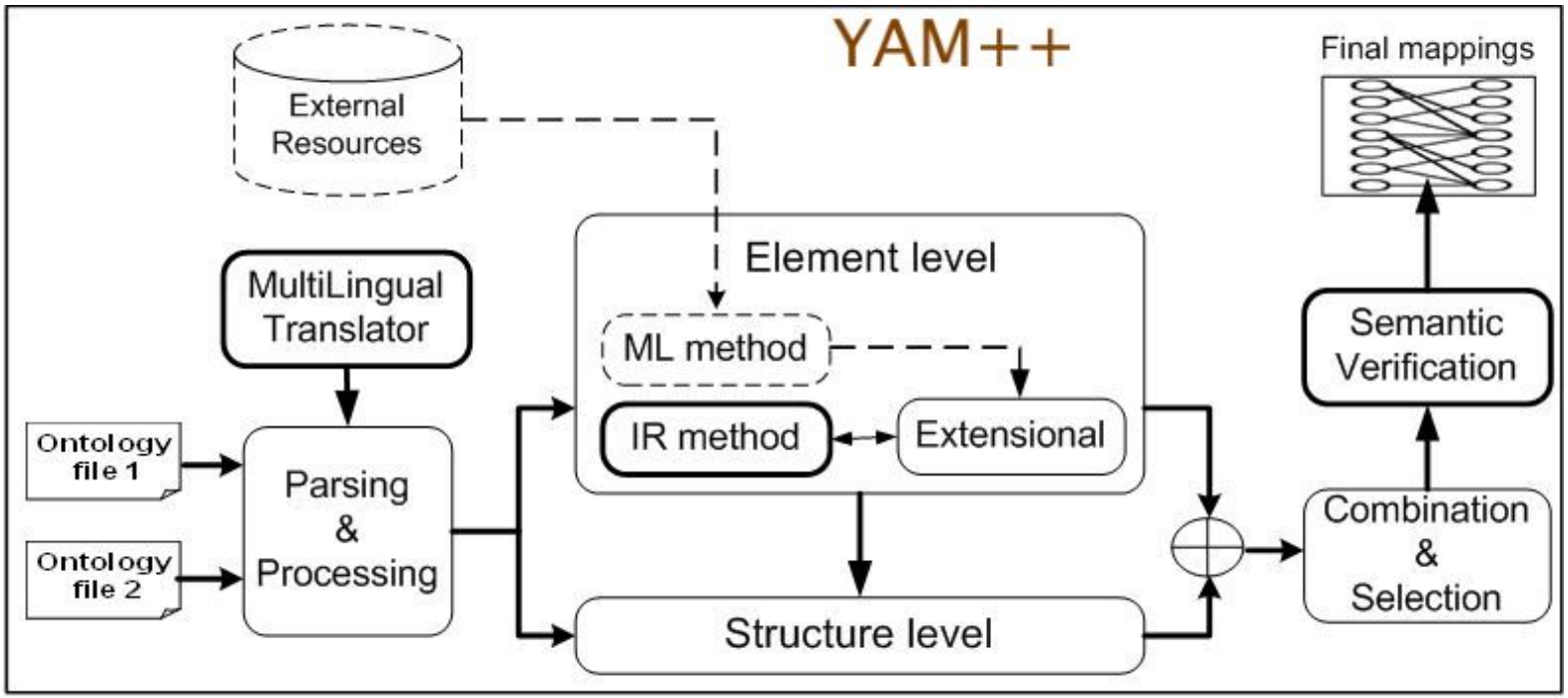
An OM System

YAM++ (not) Yet Another Matcher

Many matching systems are out there. Here are some of the pluses of YAM++:

- Automatic configuration: similarity measures selection, tuning, and combination
- A novel terminological measure based on Tversky's similarity
- Able to deal with large ontologies

Among the best performing systems in the current state-of-the-art (cf. OAEI reports)



[Ngo, Bellahsene, EKAW 2012], [http://oaei.ontologymatching.org]

Outline

Background knowledge (BK) – any piece of external information that improves or enables the alignment [8].

- Dictionaries, thesaurus, previous alignments, ontologies, the web...
- A domain specific source of knowledge
- Any external source of knowledge

- The use of BK results in a transformation of the input ontologies

Where to look for BK [6]?

1. the web and specifically linked data, Wikipedia [8];
2. domain specific corpora (of schemas and mappings);
3. domain specific ontologies, e.g., in the field of anatomy, upper-level ontologies, or all the ontologies available on the semantic web;
4. auxiliary sources: dictionaries, lexical databases, thesauri, ...

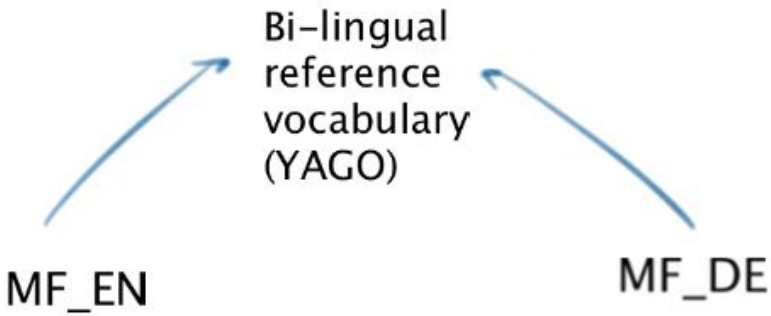
Current Topics in OM

Multilingualism

Motivation

- No one-to-one correspondence between the majority of terms across different languages
- Machine translation still tolerates low precision levels
- No large training corpora with OM data

Use of background knowledge [7]



- Implicit alignment of cross-lingual ontologies (mediated by a YAGO/Wordnet taxonomy with multilingual labels)
- No use of automatic translation
- Allows to capture various aspects of the similarity of concepts given in different languages

Current Topics in OM

...and also

- User Involvement: include the user in the matching process
- Large-scale matching (large ontologies or multiple ontologies)
- Many-to-many type alignment
- Matcher evaluation
- ...

Outline

Data Linking

The 4th principle of the web of data:

when publishing data, provide links to other, already published data!

Connect datasets on the web!

Data Linking

The link-statement is a triple, as any other triple,

- linking an instance from one dataset (the subject)
- to an instance of another dataset (the object)
- via a link-predicate given by established vocabularies, for example:
  - owl:sameAs (meaning that 2 instances (or resources) are identical),
  - but also skos:closeMatch, rdf:seeAlso, etc.,

Example:
(`http://yago-knowledge.org/resource/Ludwig_van_Beethoven`, owl:sameAs, `http://dbpedia.org/resource/Ludwig_van_Beethoven`)

Data Linking: Instance Matching

Data Linking vs. Instance Matching ? Question of link-predicate...

Definition (Instance Matching)

Let i_1 and i_2 be two instances (or resources) across two sets of instances I_1 and I_2 (respectively). We define the function:

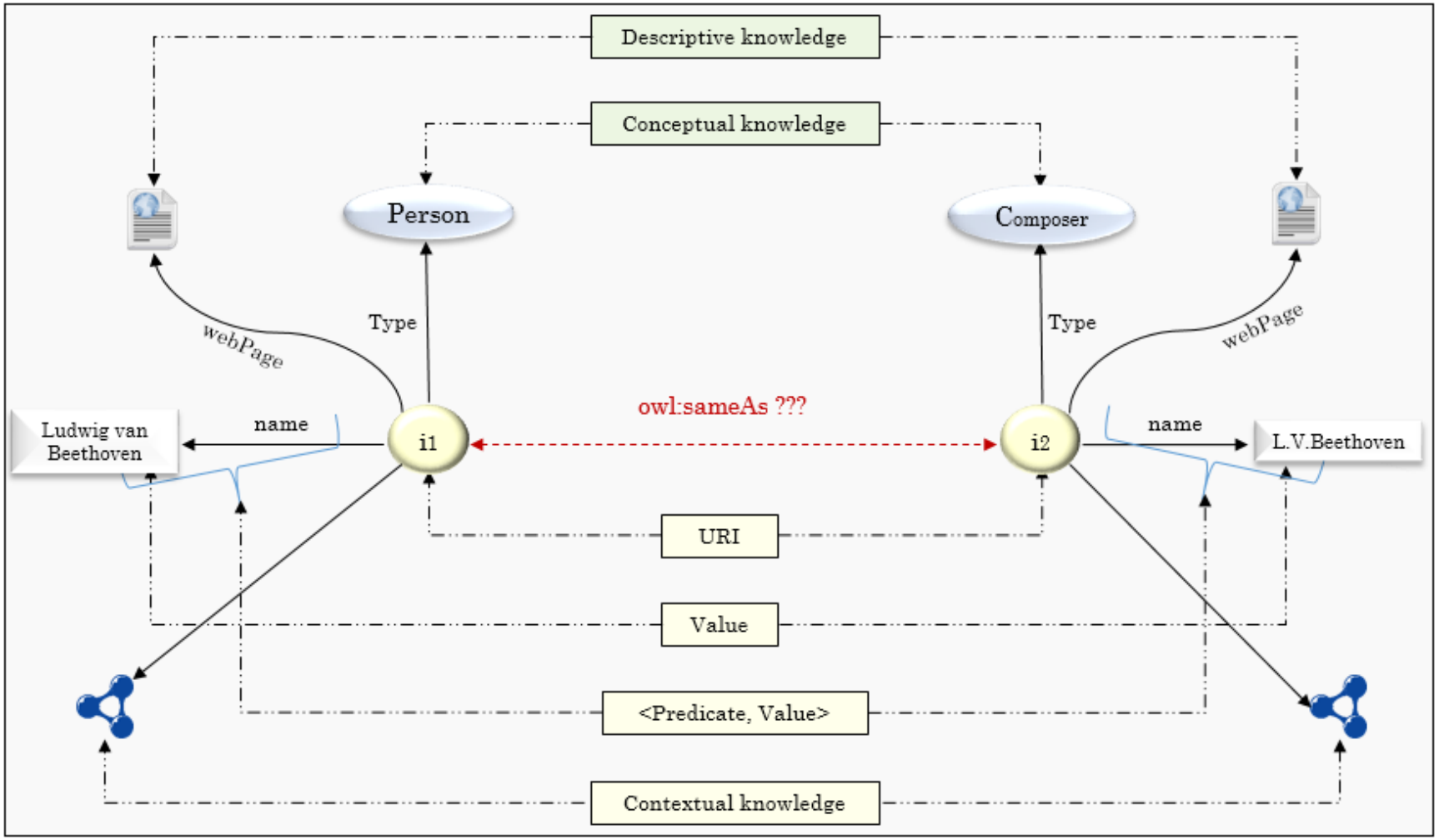
$$f: I_1 \times I_2 \rightarrow [0,1]$$
$$(i_1, i_2) \mapsto s,$$

where $s \in [0,1]$ and $s = f(i_1, i_2)$.

The function f produces a similarity value s measuring the proximity between two RDF resources i_1 and i_2. These resources are linked together (i.e., i_1 and i_2 represent the same real world object) if s is greater than a given threshold $\sigma \in [0,1]$.

Data Linking: levels of instance comparison

Where to look for information to compare instances? What describes them?

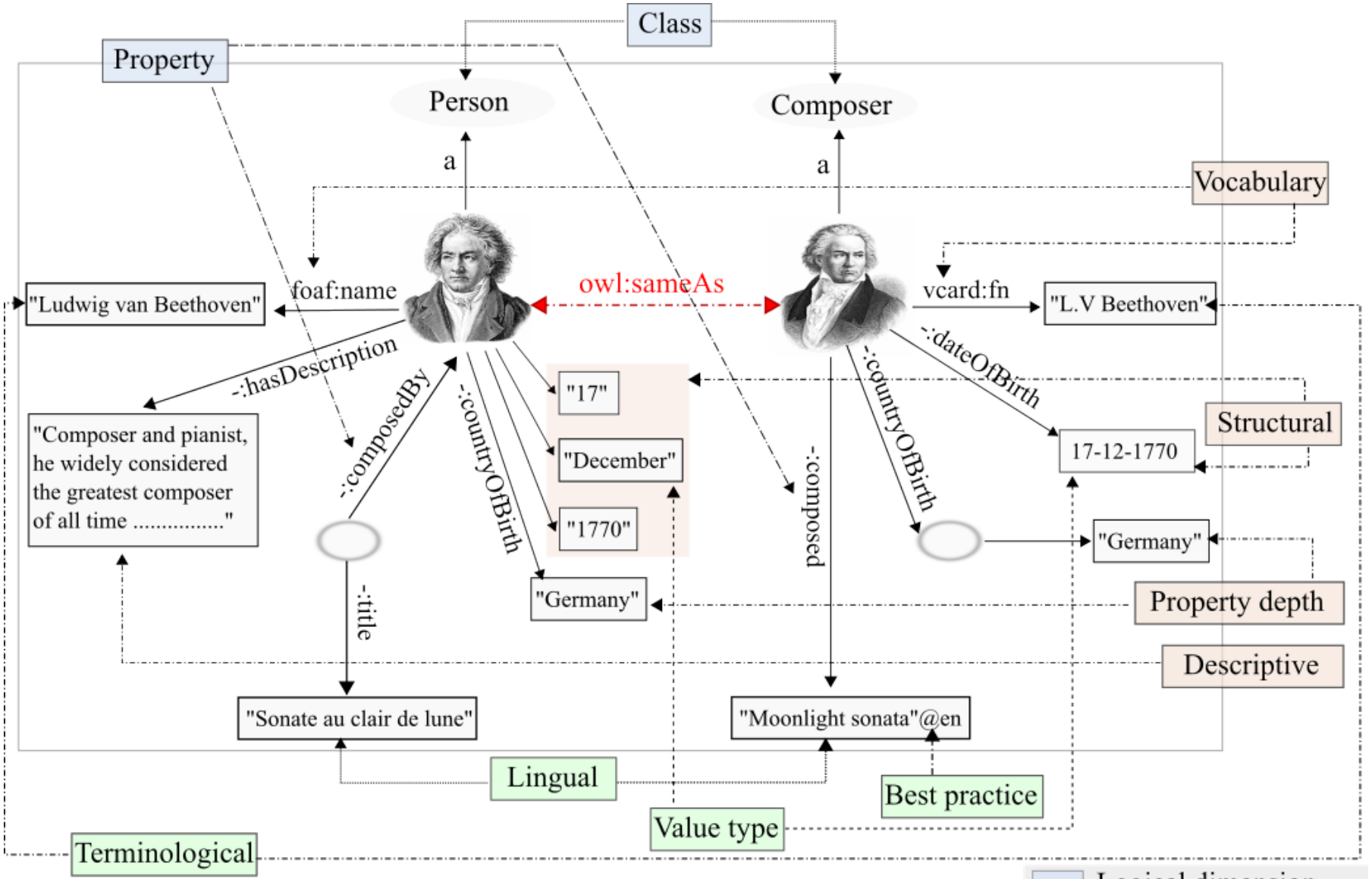


Data Linking: Heterogeneities of instances

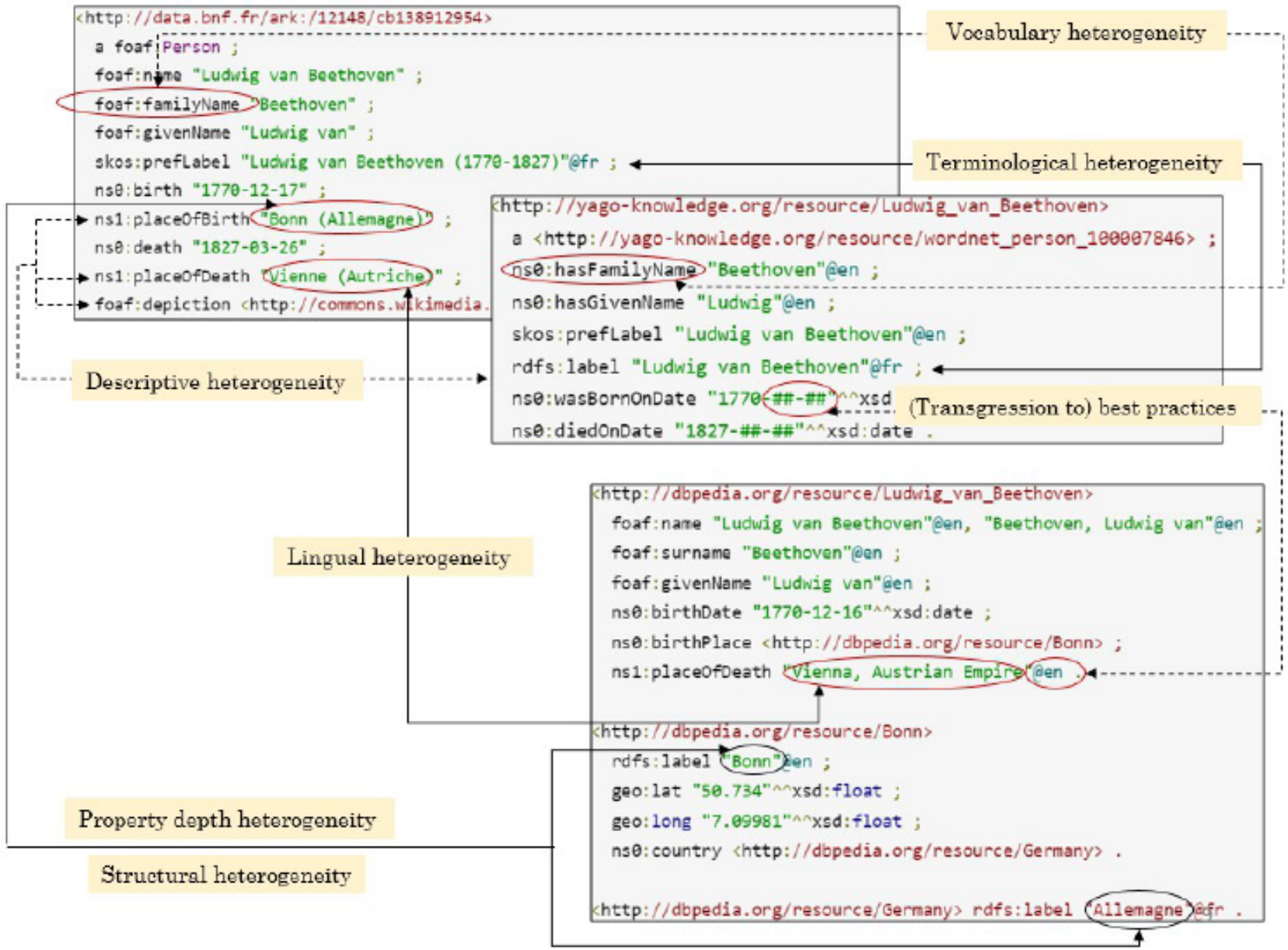
Ontological, logical and value levels... data quality... For example:

- Ontological: use of different vocabularies
  (1) Concept level. A given entity can belong to several types (ontological concepts).
  (2) Property level. Properties are often described by different vocabularies depending on the intension.
- Terminological: String literals can be given by using different terms (minor differences in spelling, synonyms)
- Structural: The description of an entity can be done at different levels of detail.
- Linguistic: Information can be expressed in different natural languages.
- Descriptive: A resource can be described with more information (a larger set of properties) in one dataset compared to another.

Data Linking: Heterogeneities of instances



Data Linking: Heterogeneities of instances



Data Linking: a complexe process

The data linking processing chain:

(1) preprocessing —> (2) instance matching —> (3) post-processing.

1. preprocessing:
   - reduce the search-space, identify a set of pairs of linking candidates: keys-identification
   - make instances comparable: models of presentation, handling multilingualism
2. instance matching: define a link between two resources, give its type and confidence value for the match
3. post-processing: filter out erroneous matches, infer new ones

Many ontology matching techniques are used in the data linking process.

Data Linking: a complexe process

The data linking processing chain:

(1) preprocessing —> (2) instance matching —> (3) post-processing.

1. preprocessing:
   - reduce the search-space, identify a set of pairs of linking candidates: keys-identification
   - make instances comparable: models of presentation, handling multilingualism
2. instance matching: define a link between two resources, give its type and confidence value for the match
3. post-processing: filter out erroneous matches, infer new ones

Many ontology matching techniques are used in the data linking process.

Data Linking: a complexe process

The data linking processing chain:

(1) preprocessing —> (2) instance matching —> (3) post-processing.

1. preprocessing:
   - reduce the search-space, identify a set of pairs of linking candidates: keys-identification
   - make instances comparable: models of presentation, handling multilingualism
2. instance matching: define a link between two resources, give its type and confidence value for the match
3. post-processing: filter out erroneous matches, infer new ones

Many ontology matching techniques are used in the data linking process.

Data Linking: a complexe process

The data linking processing chain:

(1) preprocessing —> (2) instance matching —> (3) post-processing.

1. preprocessing:
   - reduce the search-space, identify a set of pairs of linking candidates: keys-identification
   - make instances comparable: models of presentation, handling multilingualism
2. instance matching: define a link between two resources, give its type and confidence value for the match
3. post-processing: filter out erroneous matches, infer new ones

Many ontology matching techniques are used in the data linking process.

Data Linking: a complexe process

The data linking processing chain:

(1) preprocessing —> (2) instance matching —> (3) post-processing.

1. preprocessing:
   - reduce the search-space, identify a set of pairs of linking candidates: keys-identification
   - make instances comparable: models of presentation, handling multilingualism
2. instance matching: define a link between two resources, give its type and confidence value for the match
3. post-processing: filter out erroneous matches, infer new ones

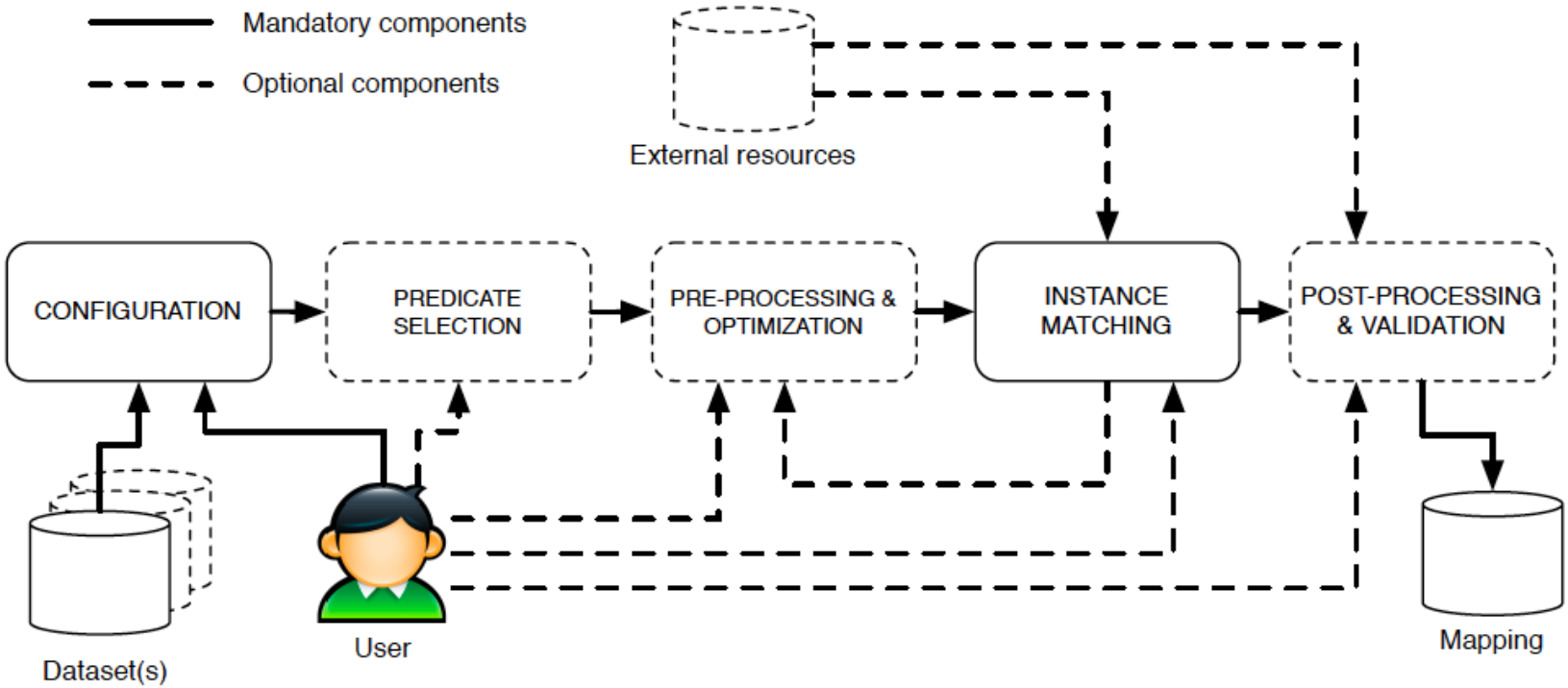
Many ontology matching techniques are used in the data linking process.

Data Linking: a complexe process

A generic architecture of a tool

A large choice of tools: LIMES[3], SILK[4], RiMOM, RDF-AI,...

From a user perspective, the tool configuration is 90% of the task.



Taken from [2].

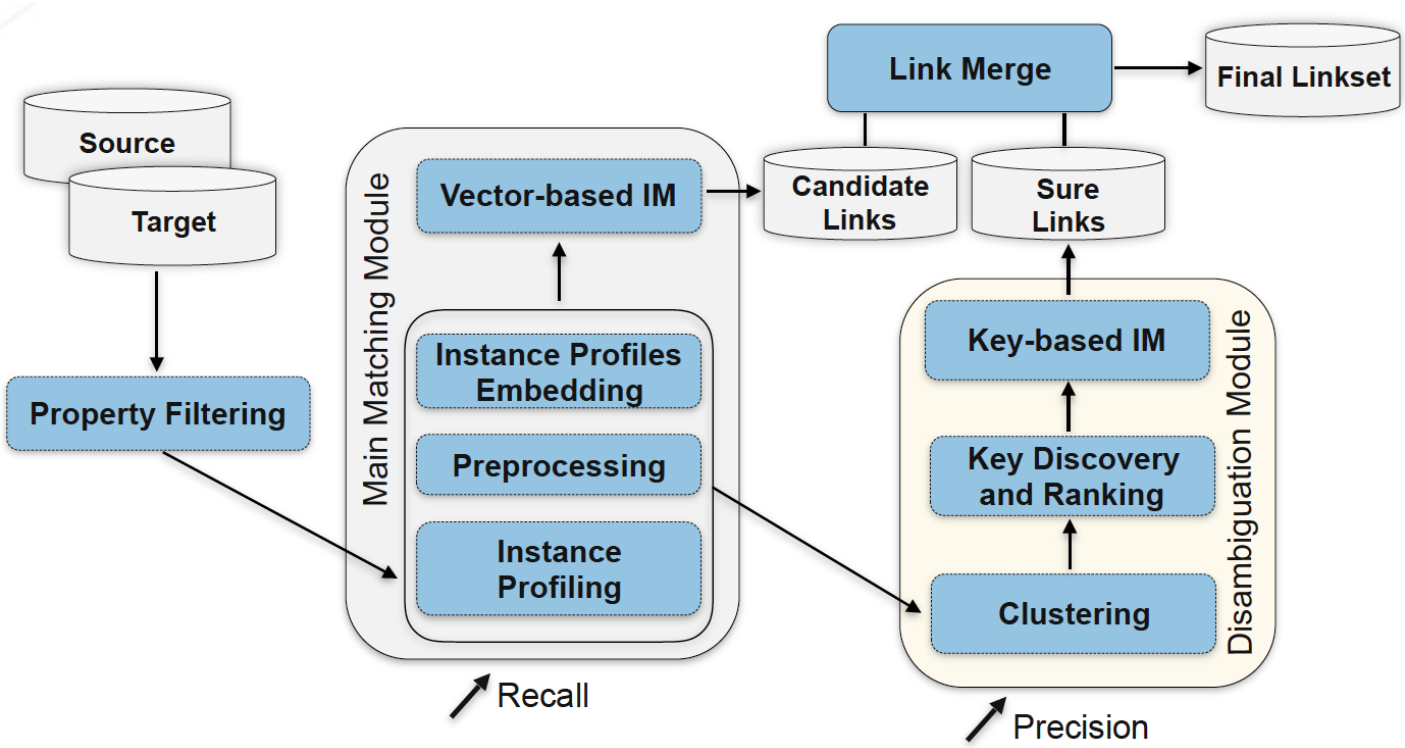
[3] http://aksw.org/Projects/LIMES.html

[4] http://silkframework.org

Data Linking: Legato

A tool developed at LIRMM

Source code, user interface and documentation:
`https://github.com/DOREMUS-ANR/legato`



Data Linking: Evaluation

Instance Matching track of the OAEI[5].

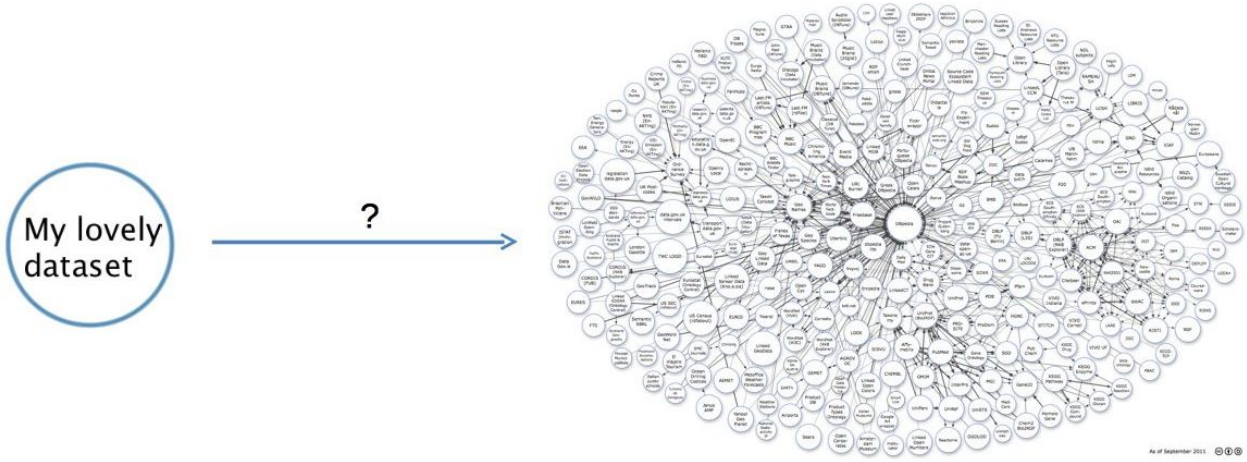
—> The same principle as what we saw in the previous chapter regarding ontology matching:

- reference data in the form of ground truth (e.g., 2 datasets and a link set)
- a tool is evaluated against these reference data by the help of the standard evaluation metrics (Precision, Recall and F-measure)

[5] http://islab.di.unimi.it/im_oaei_2014/index.html

Data Linking: current topics

- Dataset recommendation for linking



- Dataset profiling
- Key discovery
- Linking multilingual data
- Post-processing (errors, missing links)
- Semantics of links: “owl:sameAs” is too restrictive...
- ...

Questions?


J. Euzenat and P. Shvaiko.

Ontology matching.
Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.


Alfio Ferrara, Andriy Nikolov, Jan Noessner, and François Scharffe.

Evaluation of instance matching tools: The experience of OAEI.
J. Web Sem., 21:49–60, 2013.


T.R. Gruber et al.

Toward principles for the design of ontologies used for knowledge sharing.
Int. J. of Hum. Comp. Stud., 43(5):907–928, 1995.


DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov.

Extended tversky similarity for resolving terminological heterogeneities across ontologies.
In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 711–718. Springer, 2013.


DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov.

Opening the black box of ontology matching.
In *The Semantic Web: Semantics and Big Data*, pages 16–30. Springer, 2013.


P. Shvaiko and J. Euzenat.

Ontology matching: state of the art and future challenges.
Knowledge and Data Engineering, IEEE, 25(1):158–176, 2013.


Konstantin Todorov, Celine Hudelot, and Peter Geibel.

Fuzzy and cross-lingual ontology matching mediated by background knowledge.
In Fernando Bobillo, Rommel N. Carvalho, Paulo C.G. Costa, Claudia d'Amato, Nicola Fanizzi, Kathryn B. Laskey, Kenneth J. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web III*, Lecture Notes in Computer Science, pages 142–162. Springer International Publishing, 2014.


Konstantin Todorov, Céline Hudelot, Adrian Popescu, and Peter Geibel.

Fuzzy ontology alignment using background knowledge.
International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 22(1):75–112, 2014.