CSR: Discovering Subsumption Relations for the Alignment of Ontologies

Vassilis Spiliopoulos, Alexandros G. Valarakos, and George A. Vouros

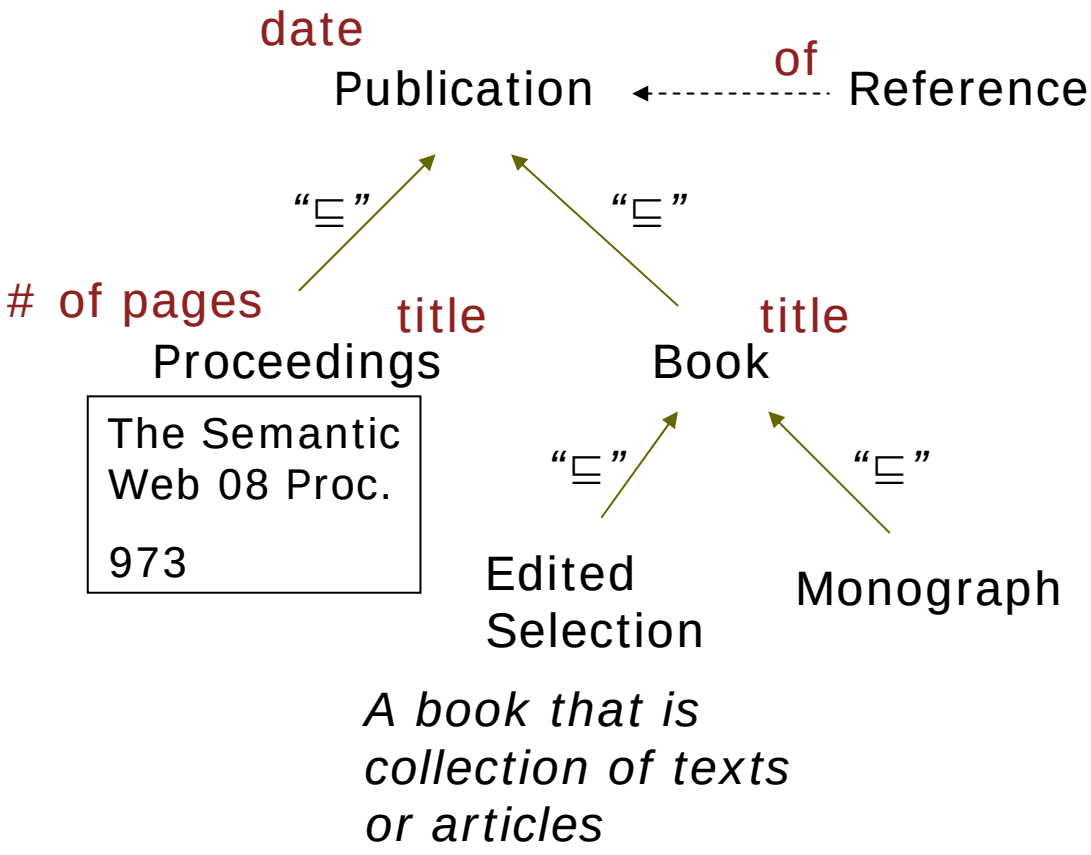
AI Lab
Department of Information and Communication Systems Eng.
University of the Aegean
83200 Karlovassi, Samos, Greece
{vspiliop, alexv, georgev}@aegean.gr

Outline

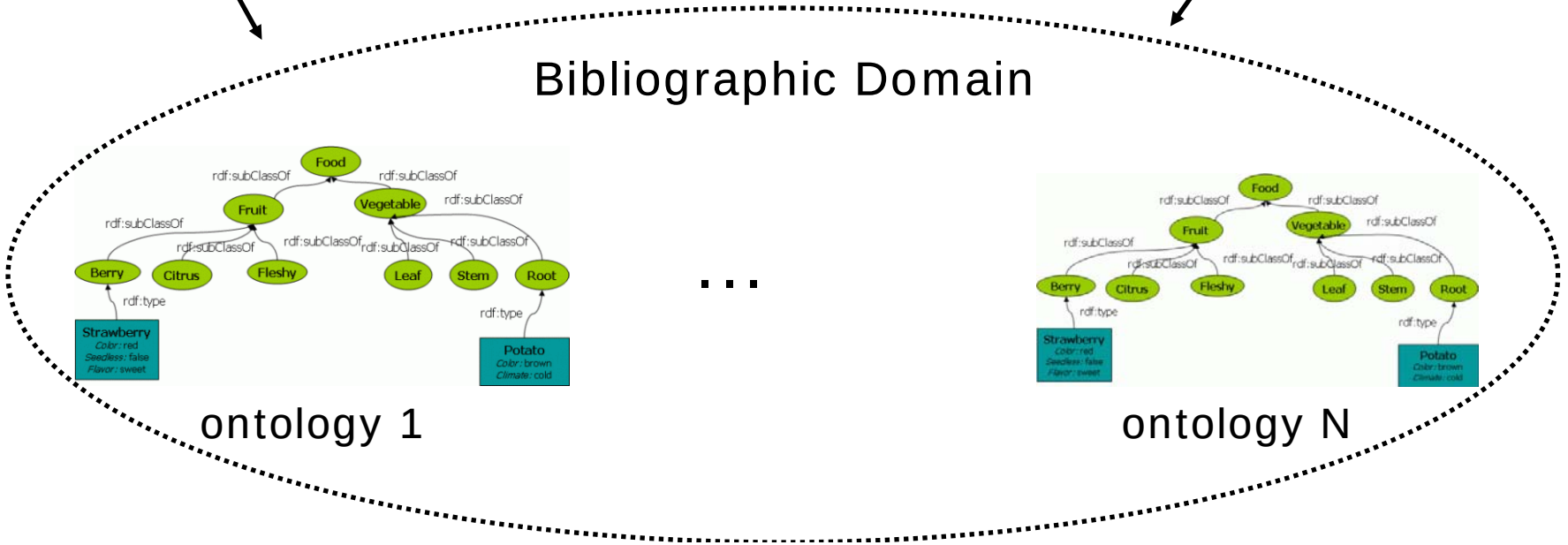
- Introduction
- Problem Definition
- Why Subsumption Relations
- The Method
- Experimental Results
- Conclusions

Ontology

- **Concept** features
  - Properties
    - Data type
    - Object Property (relation)
  - Instances
  - *Comments*
- **Concepts** organized into **hierarchies** (subsumption relation)
- Ontology Languages
  - OWL Family
  - Union, Intersection, Disjointness

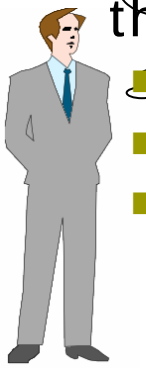


Current Situation



Ontology Mapping

- **Ontology Mapping** is a process that has as input two ontologies and locates relations (i.e. mappings) between their elements
  - Equivalence (≡)
  - Intersection (□)
  - **Subsumption** (⊑ or ⊒)

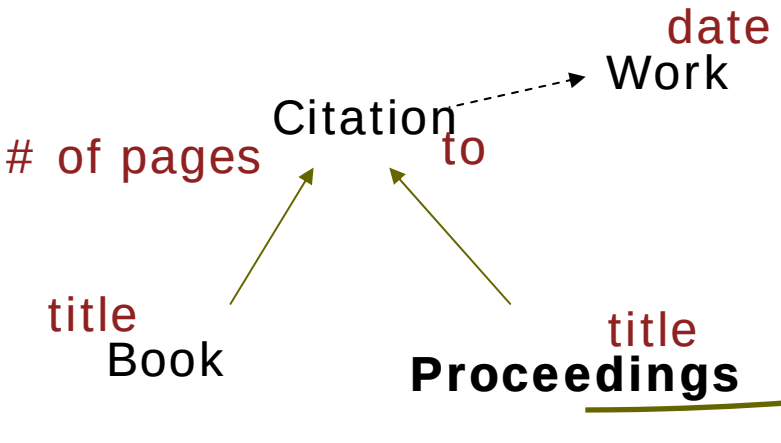


"Find Citations in Proceedings"

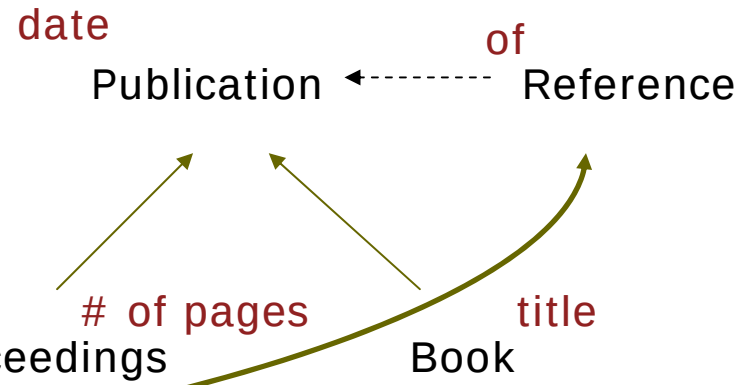
Locates a mapping

Retrieves a superset of what he is looking for

Agents' Ontology

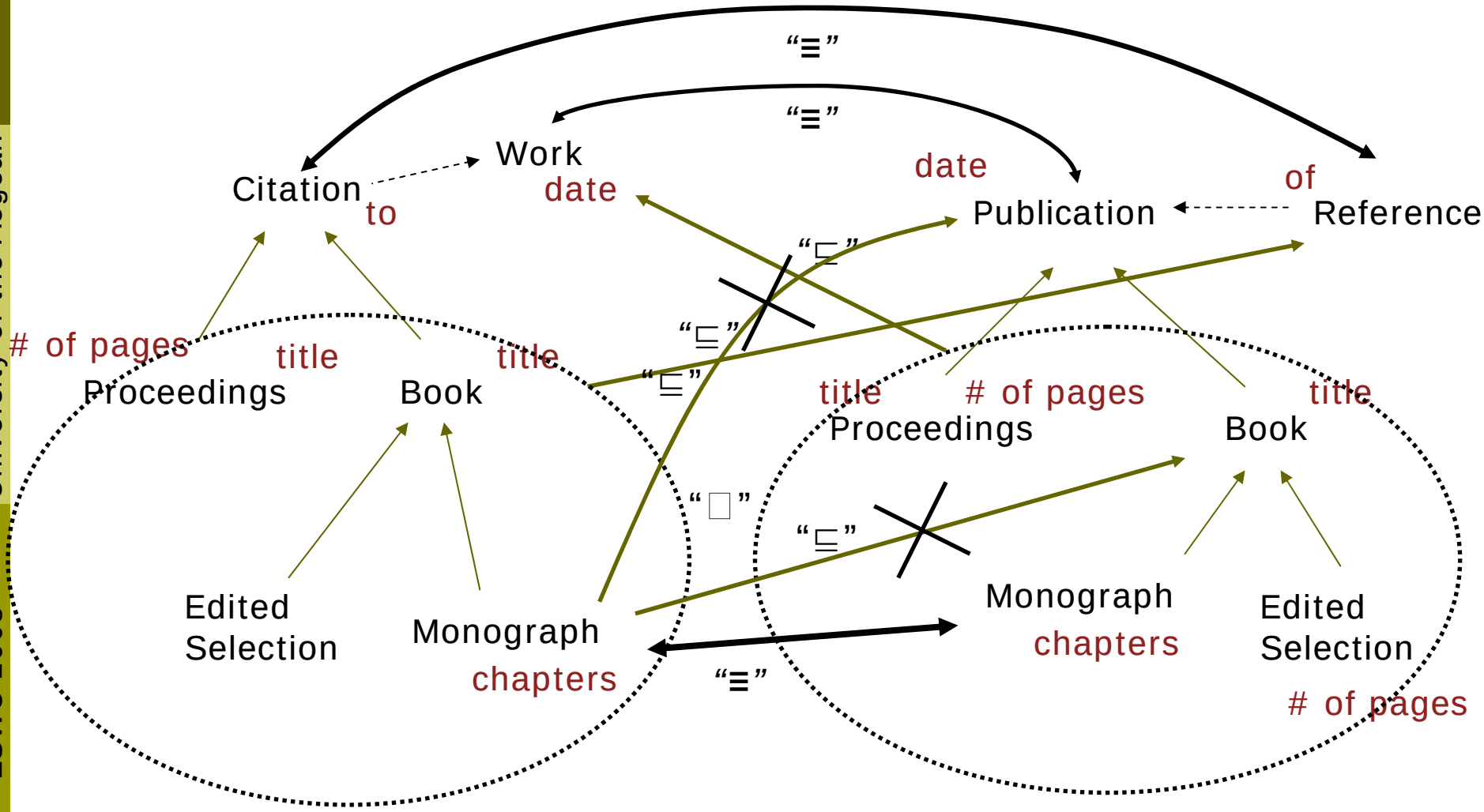


Conference Ontology



"⊑"

Why Subsumption Relations (1/2)



Why Subsumption Relations (2/2)

- Discover subsumption relations **separately** from subsumptions and equivalencies that can be deduced by a **reasoning mechanism**
- May **augment the effectiveness** of current ontology mapping and merging methods
- **No or few equivalences**
- **Web Service matchmaking**
- **Ontology engineering** environments

Problem Definition

- The **subsumption computation problem** is defined as follows:
  - Given two input ontologies
  - **optionally**, specifying **properties' equivalences**
  - Classify each pair (C^1,C^2) of concepts to two distinct classes: To the "subsumption" ($\sqsubseteq$) class ($C^1 \sqsubseteq C^2$), or to the class *"R"*
- **Class *"R"* denotes** pairs of concepts that are not known to be related via the subsumption relation

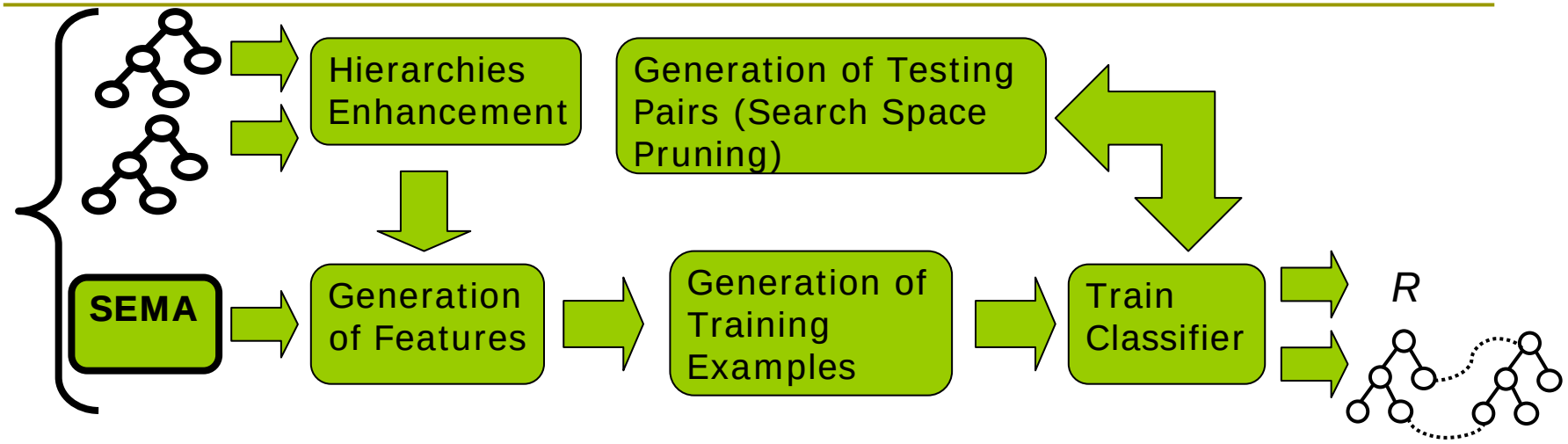
The *CSR* Method At a Glance (1/2)

- **Purpose**
  - We try to **learn patterns of features** that indicate a subsumption relation between two concepts belonging to two different ontologies
- **How**
  - By exploiting **supervised machine learning** techniques (binary classification),
  - and the **ontology specification semantics**

The *CSR* Method At a Glance (2/2)

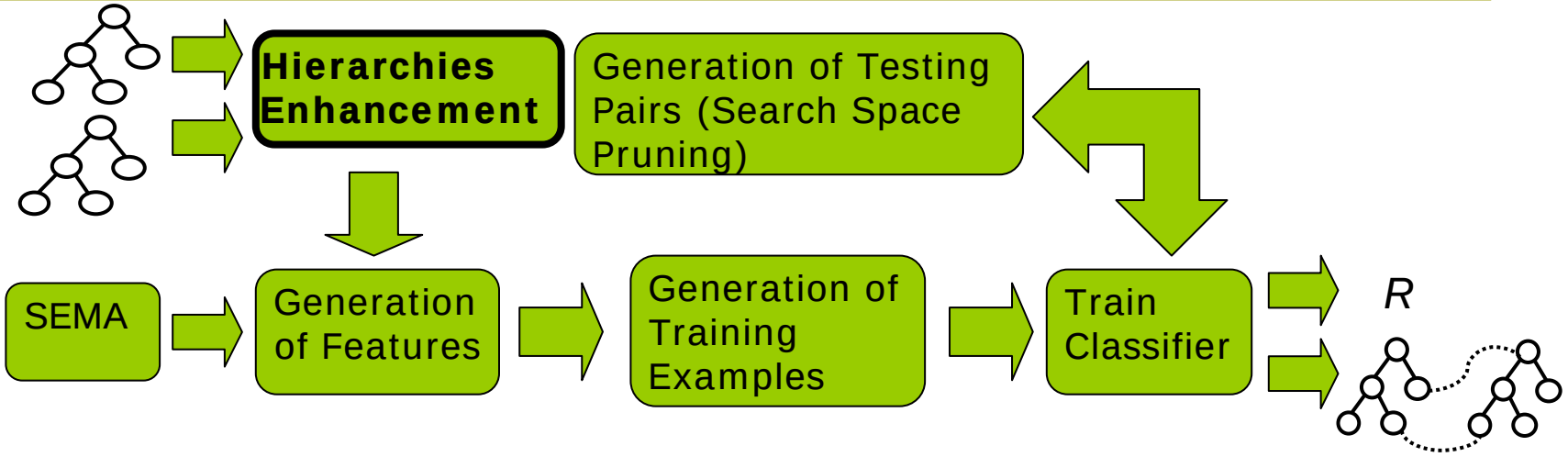
- **Why machine learning?**
  - There are **no evident generic rules directly** capturing the existence of a subsumption relation (e.g. labels/vicinity similarity or dissimilarity)
  - Learn patterns of features **not evident to the naked eye**
  - **Self-adapting** to idiosyncrasies of specific domains
  - Non-dependant to **external resources**

The *CSR* Method (1/12)



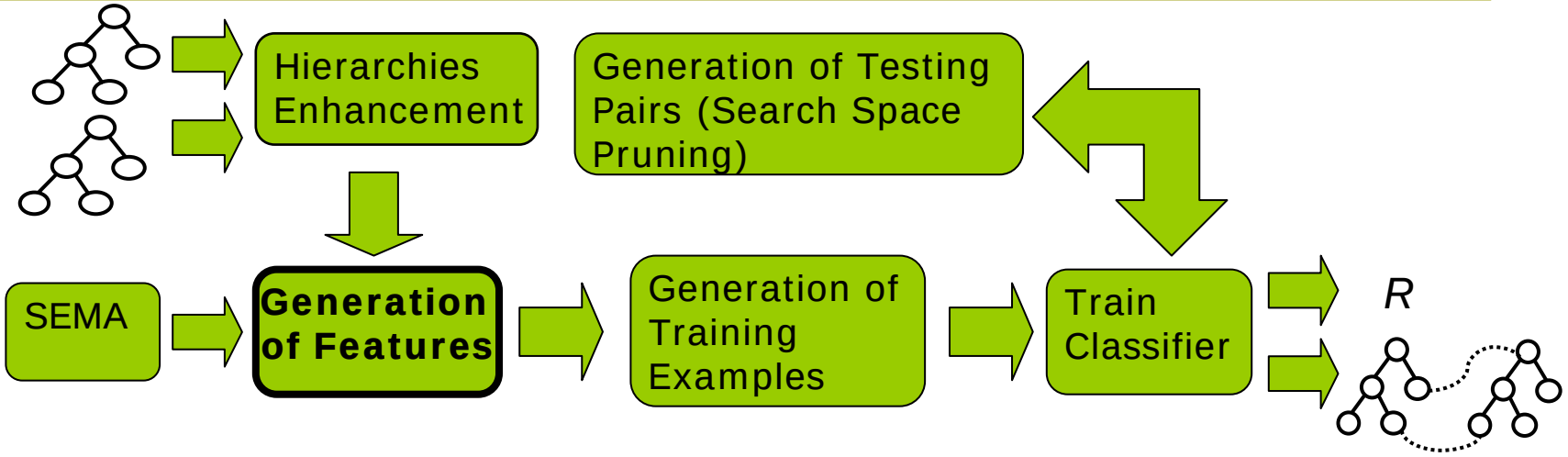
- Input
  - **Two OWL-DL ontologies** (the process is not ontology language specific)
  - **Optionally**, **property equivalencies** computed by SEMA mapping tool
  - The method requires the existence of subsumption relations between concepts

The *CSR* Method (2/12)



- Hierarchies Enhancement
  - Inferring all indirect subsumption relations
  - Influences the generation of training examples and feature vectors

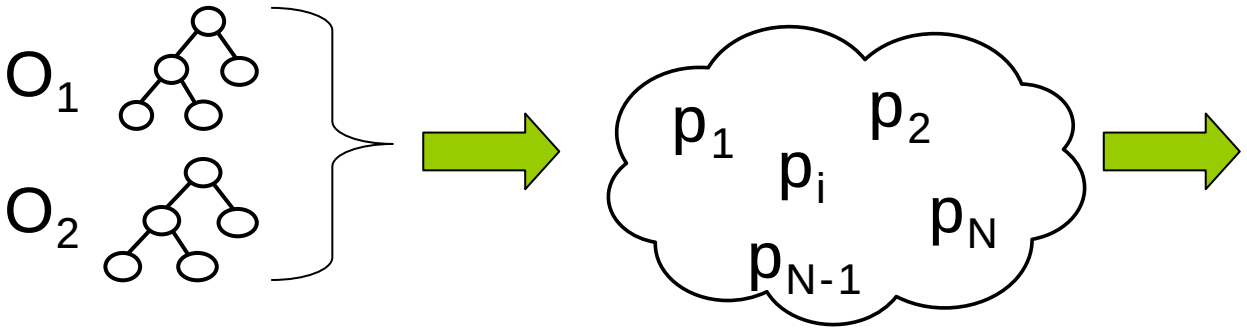
The *CSR* Method (3/12)



- Generation of Features
  - *CSR* exploits two types of features: **Concepts' properties** or **words** appearing in the "vicinity" of concepts

The *CSR* Method (4/12)

Properties Case



p_1 p_2 p_N

$(f_1, f_2, \ldots, f_N)$

f_i: i-th feature

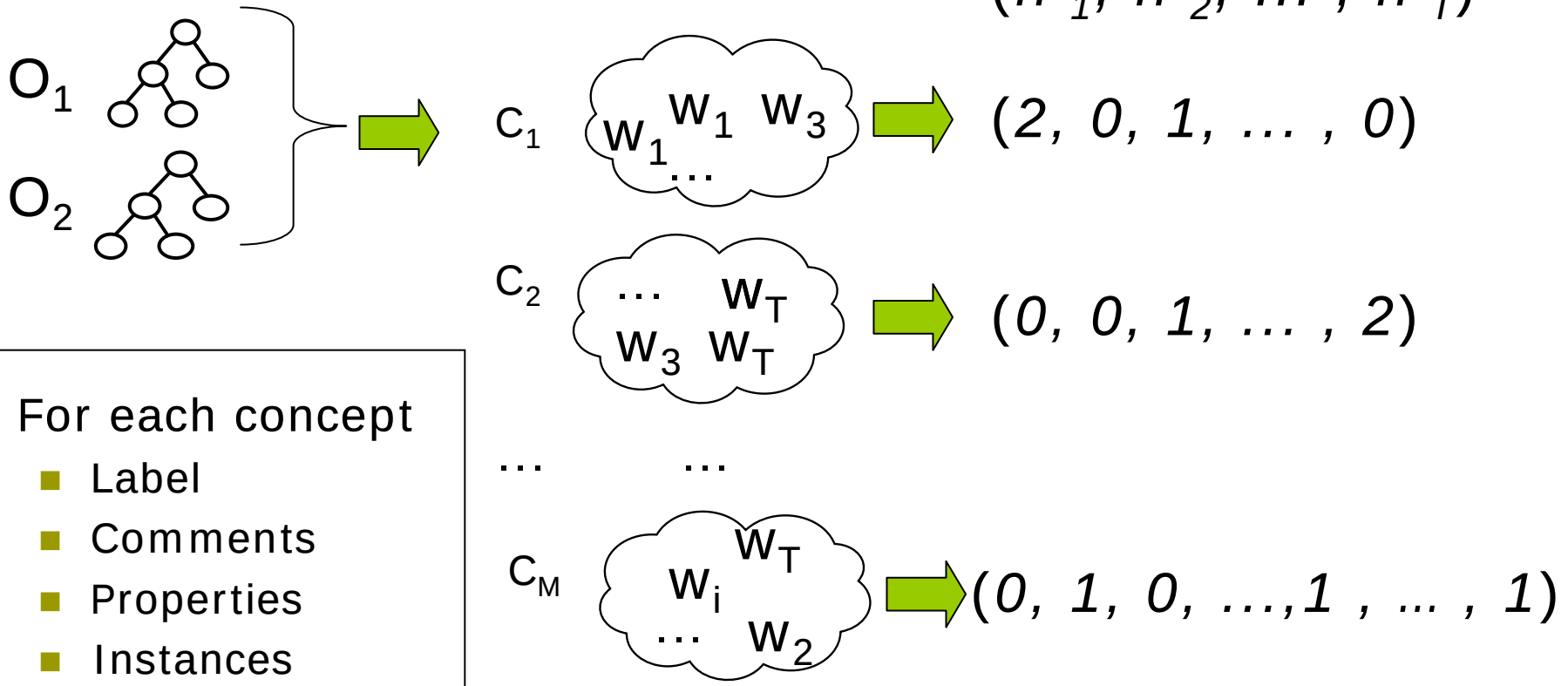
p_i: i-th property

(C^1, C^2) ⇒

$$f_i = \begin{cases} 0, \text{ if } p_i \text{ does not appear in } C^1 \text{ nor } C^2 \\ 1, \text{ if } p_i \text{ appears only in } C^1 \\ 2, \text{ if } p_i \text{ appears only in } C^2 \\ 3, \text{ if } p_i \text{ appears in both } C^1 \text{ and } C^2 \end{cases}$$

The *CSR* Method (5/12)

Words Case

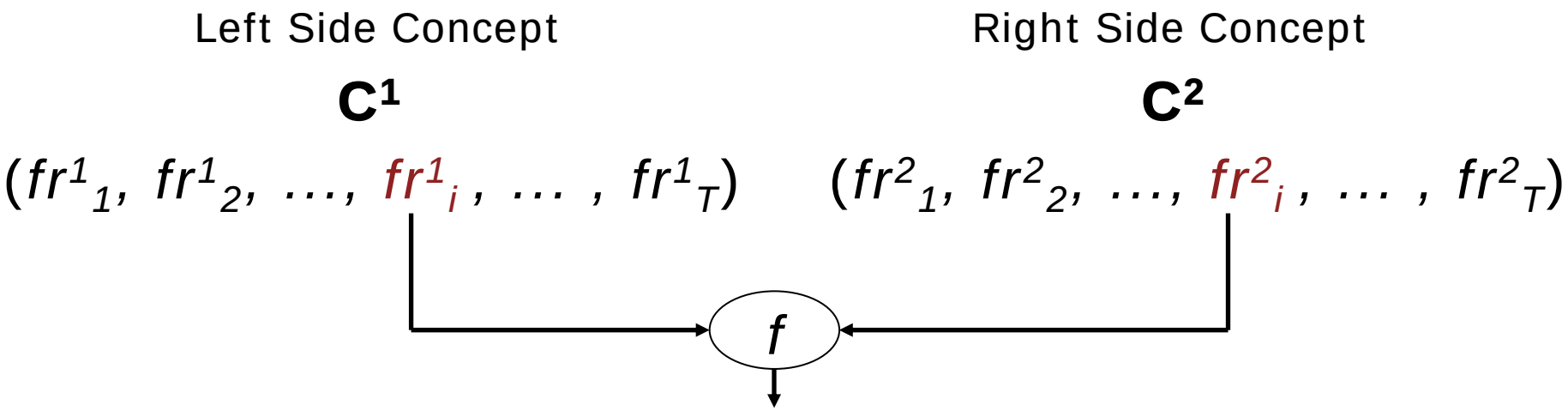


- For each concept
  - Label
  - Comments
  - Properties
  - Instances
  - Related Concepts

fr_i: frequency of i-th word

T: number of distinct words

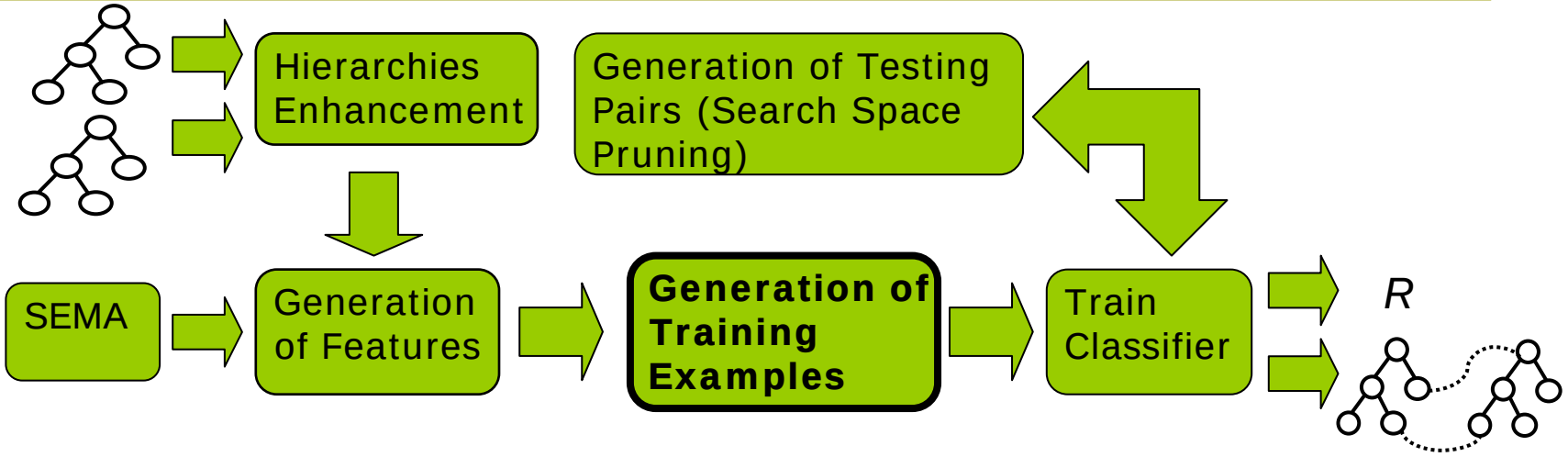
The *CSR* Method (6/12)



where

$$f_i = \begin{cases} 0, \text{ if } fr_i^1 = 0 \text{ and } fr_i^2 = 0 \\ 1, \text{ if } fr_i^1 \neq 0 \text{ and } fr_i^2 = 0 \\ 2, \text{ if } fr_i^1 = 0 \text{ and } fr_i^2 \neq 0 \\ 3, \text{ if } fr_i^1 \neq 0 \text{ and } fr_i^2 \neq 0 \end{cases}$$

The *CSR* Method (7/12)



- Generation of Training Examples
  - Classes: "$\sqsubseteq$" and *R*
  - Training examples are being generated by exploiting the input ontologies in isolation
  - According to the semantics of specifications

The *CSR* Method (8/12)

- **Class "$\sqsubseteq$"**
  - ***Subsumption Relation***. Include all concept pairs from both input ontologies that belong in the subsumption relation (**direct or indirect**)
  - ***Equivalence Relation***. Any concept in a training pair can be substituted by any of its equals
  - ***Union Constructor***. E.g. $C_4 \sqcup C_5 \sqsubseteq C_2$ => $C_4 \sqsubseteq C_2$ and $C_5 \sqsubseteq C_2$

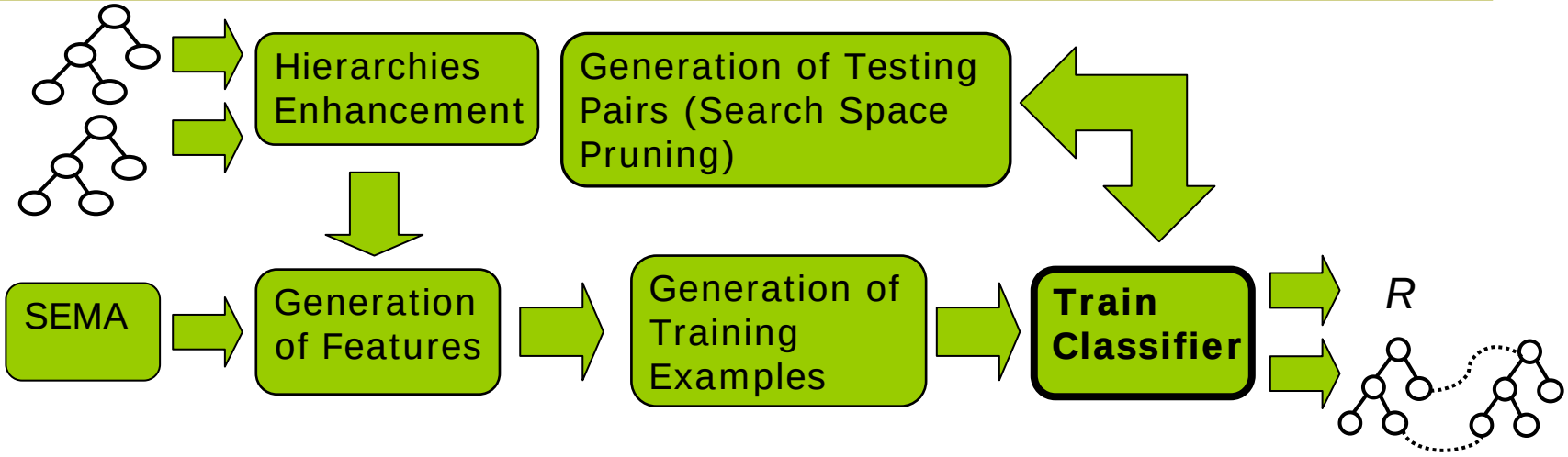
The *CSR* Method (9/12)

- ***Generic class "R"***
  - If there is not an axiom that specifies the subsumption relation between a pair of concepts
  - Categories of *class "R"*
    - Concepts belonging to different hierarchies
    - Siblings at the same hierarchy level
    - Siblings at different hierarchy levels
    - Concepts related through a non-subsumption relation
    - Inverse pairs of class "$\sqsubseteq$"

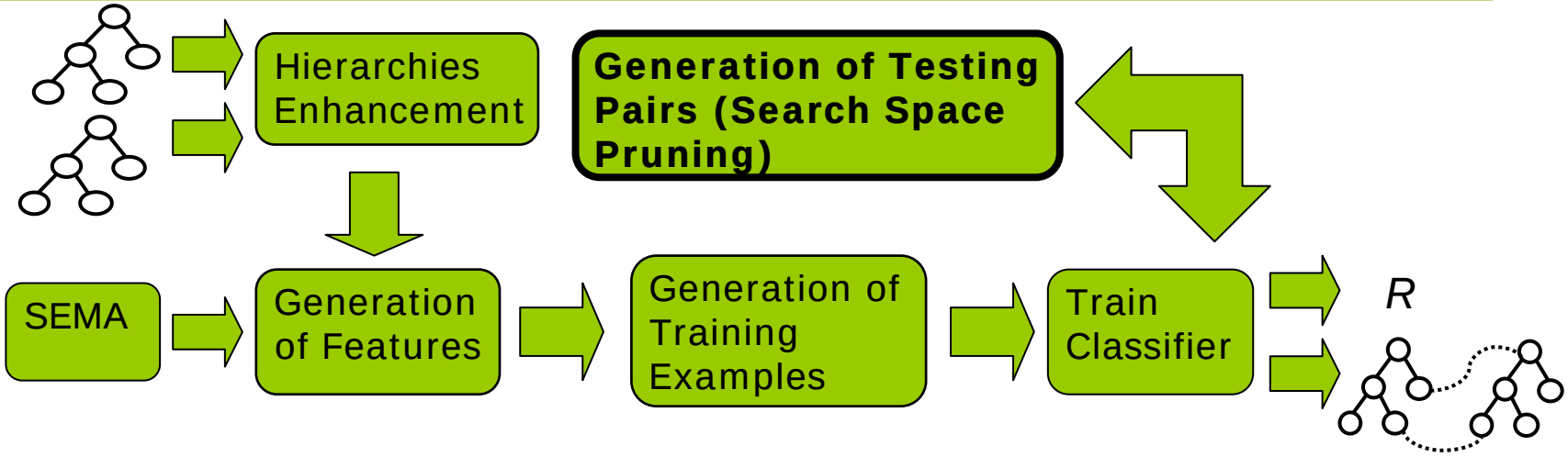
The *CSR* Method (10/12)

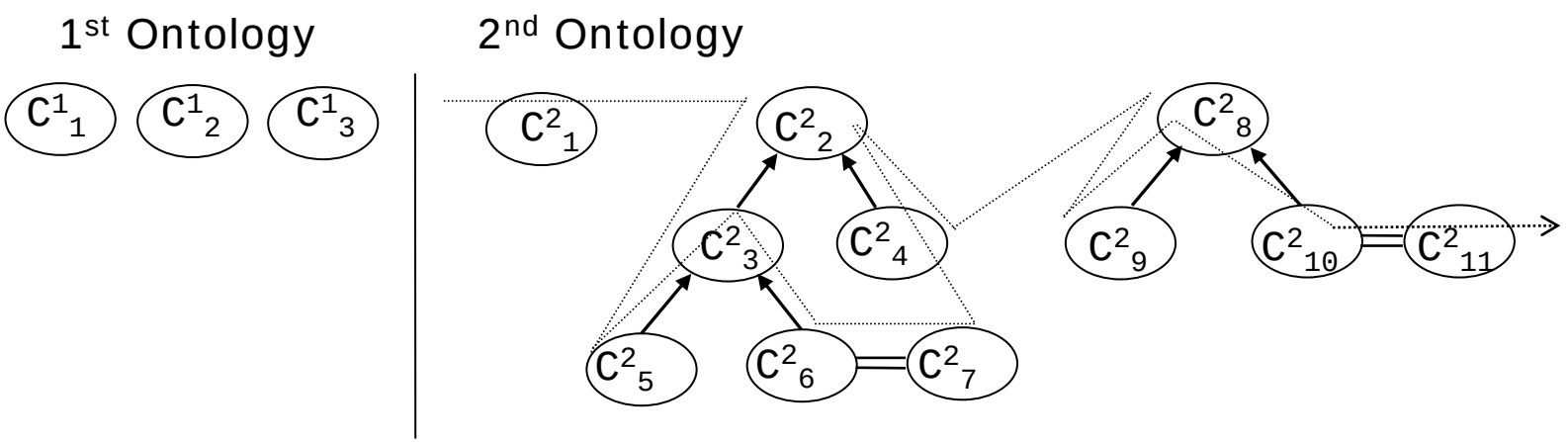
- **Balancing the Training Dataset**
  - The number of training examples for the class "$\sqsubseteq$" are much less than the ones for class "*R*"
  - Dataset imbalance problem
  - Two balancing strategies:
    - *Random under-sampling variation*
    - *Random over-sampling*

The *CSR* Method (11/12)



The *CSR* Method (12/12)

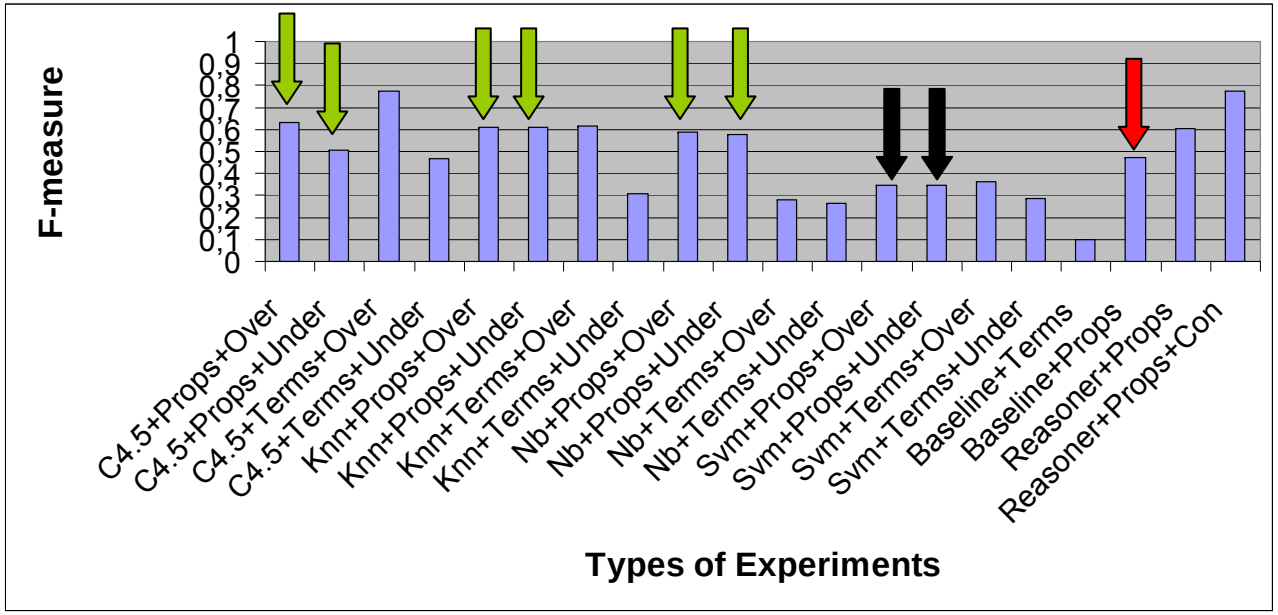




Experimental Settings

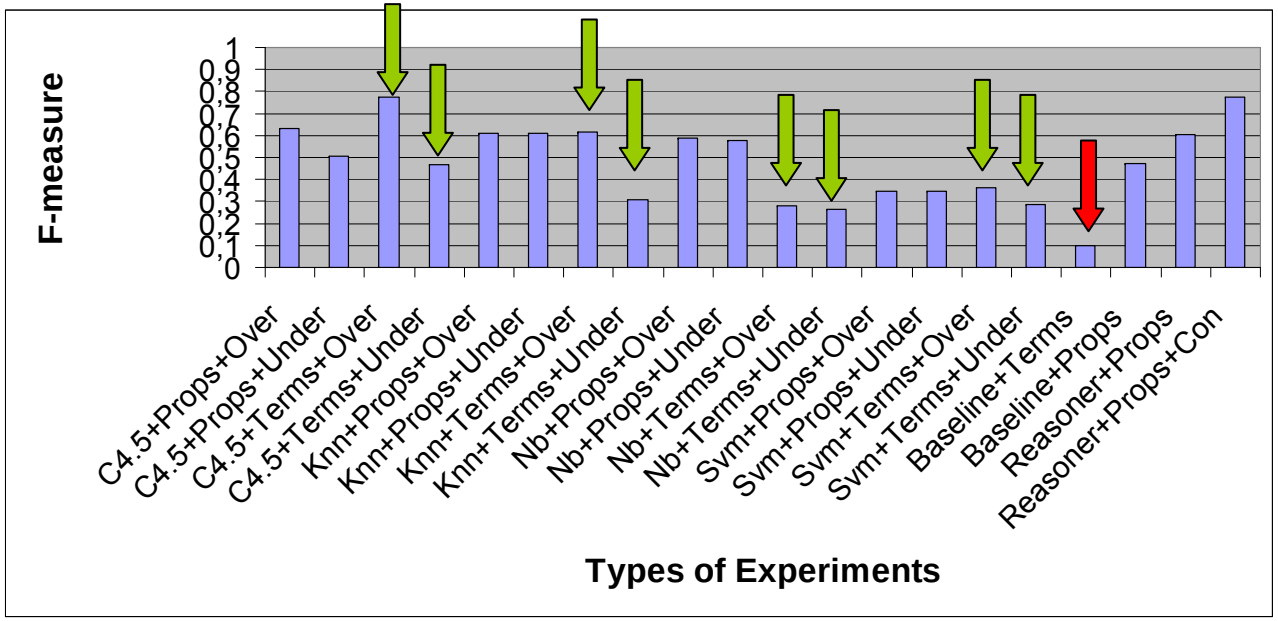
- The testing dataset has been derived from the benchmarking series of the OAEI 2006 contest
- The compiled corpus + gold standard is available at http://www.icsd.aegean.gr/incosys/csr
- Classifiers used: C4.5, Knn (2 neighbors), NaiveBayes (Nb) and Svm (radial basis kernel)
- We denote each type of experiment with A+B+C
  - **A: classifier**,
  - **B: type of features** ("Props" for properties or "Terms" for words) and
  - **C: dataset balancing method** ("over" and "under" for over- and under-sampling)
- **Baseline:** Consults the vectors of the training examples of the class "$\sqsubseteq$", and selects the first exact match (No generalization)
- **Description Logics' Reasoner**: We specify axioms concerning only properties' equivalencies (**Reasoner+Props**), or alternatively, both properties' and concepts' equivalencies (**Reasoner+Props+Con**)

Overall Results



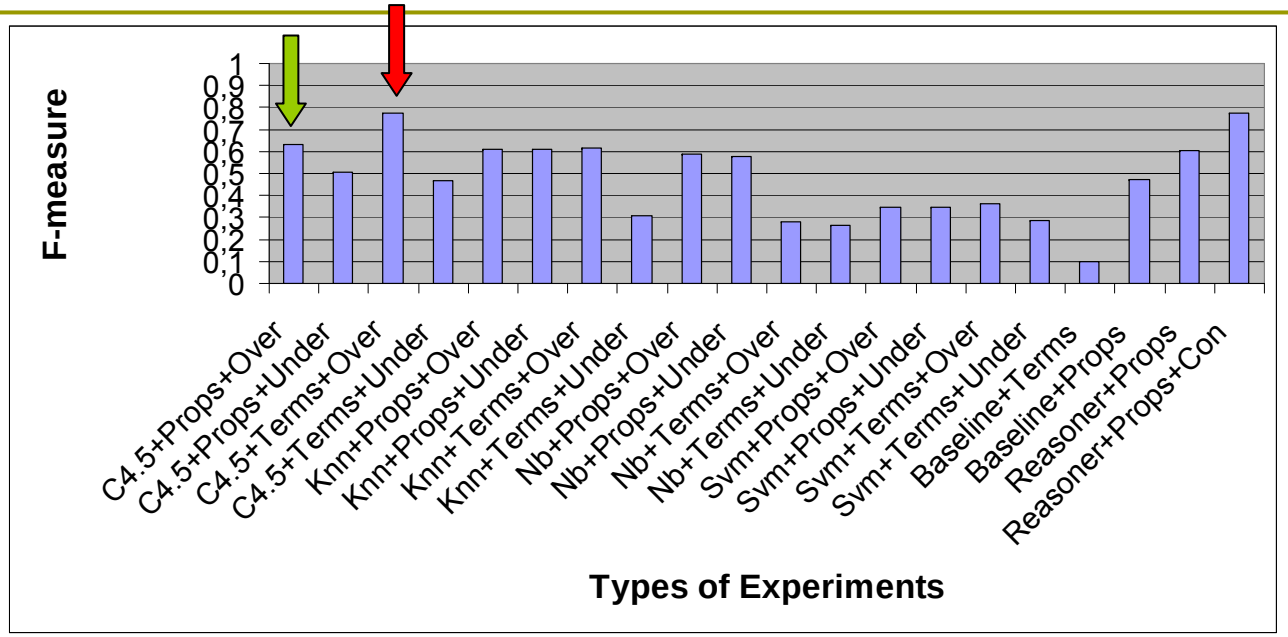
- All classifiers (except **Svm**) based on properties outperform Baseline+Props
- Generalization – location of pairs not in the training dataset

Overall Results



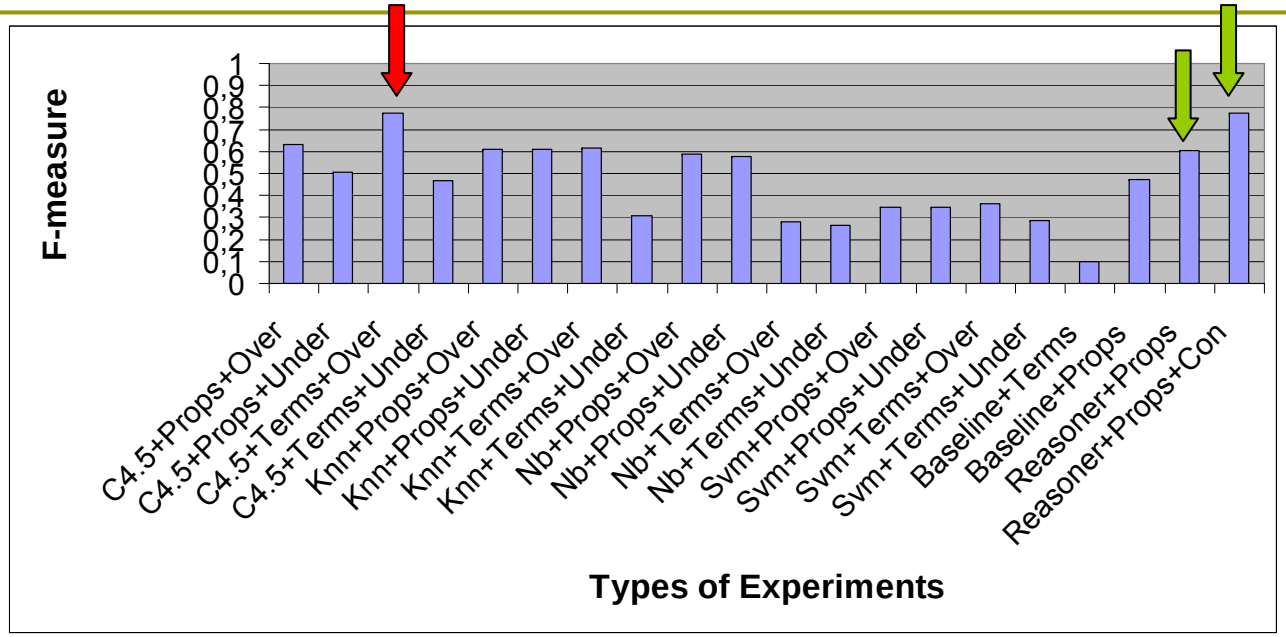
- All classifiers based on words outperform Baseline+Words
- Generalization – location of pairs not in the training dataset

Overall Results



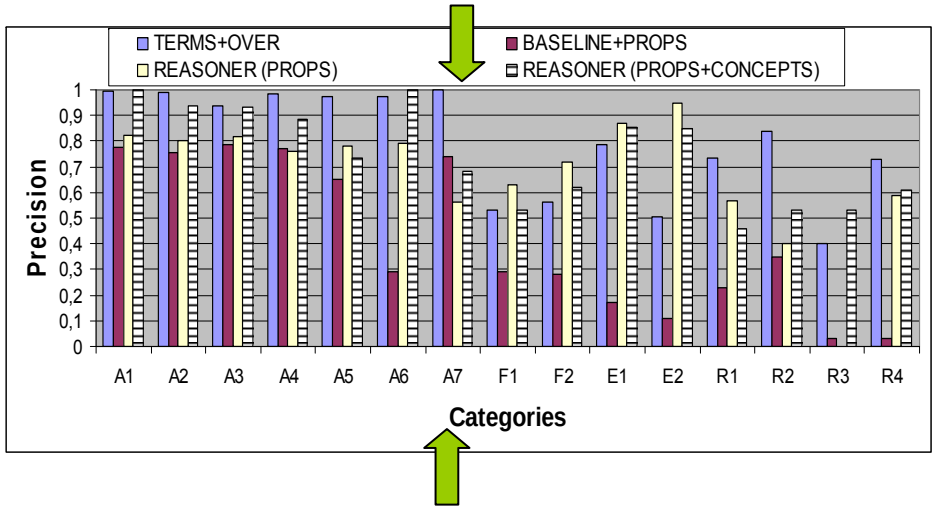
- C4.5 (using either words or properties) performs best comparing to all other CSR experimentation settings
- Disjunctive descriptions of cases: More than one features may indicate whether a specific concept pair belongs in the class "$\sqsubseteq$"
- Decision trees are very tolerant to errors in the training set. Both to feature vectors and training examples

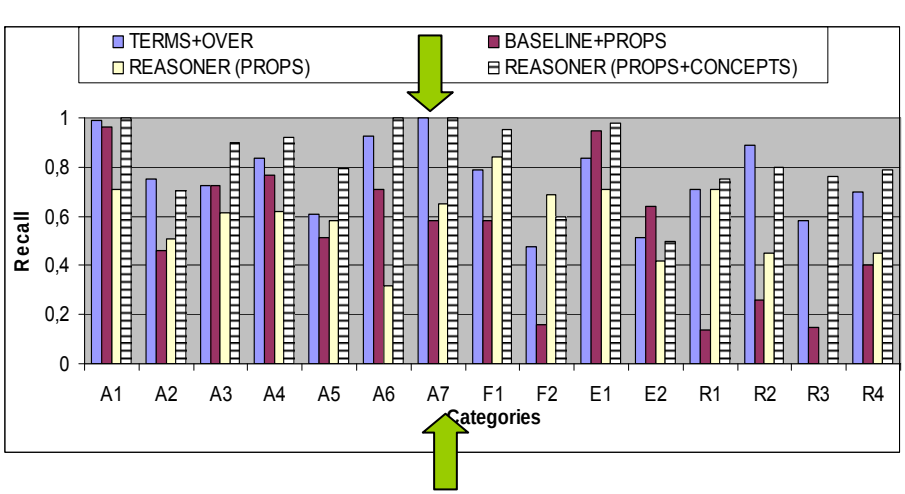
Overall Results



- *CSR* exploiting words does not require neither properties nor concepts equivalencies
- Reasoner exploits such equivalencies
  - Depends on the mapping tool

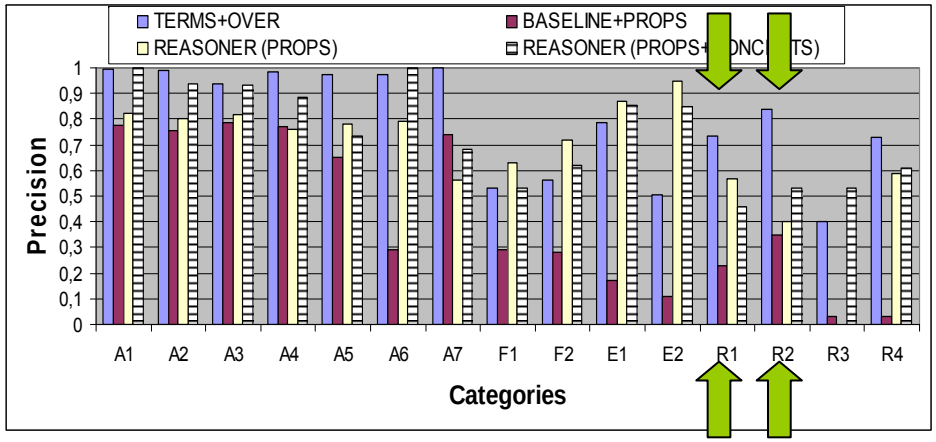
Closer Look

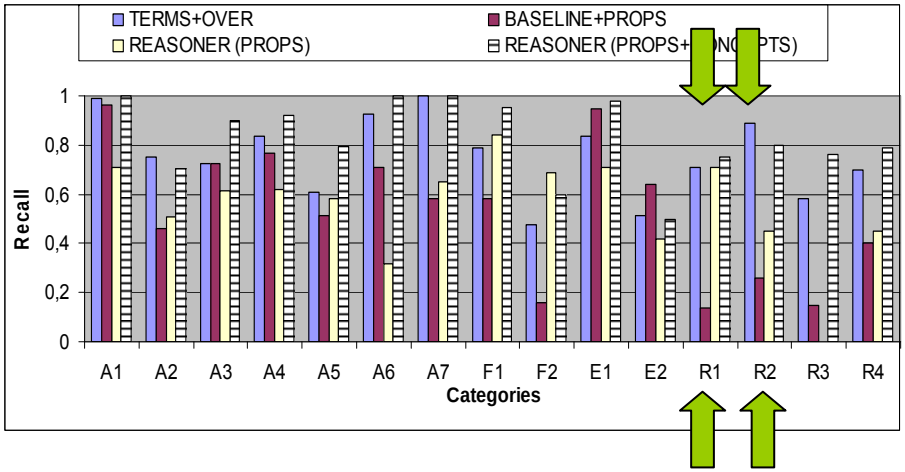




- A7 Category
  - Different conceptualizations
  - Flattened classes in target ontology + props defined in a more detailed manner
  - SEMA: 74% Precision – 100% Recall (Props+Cons)

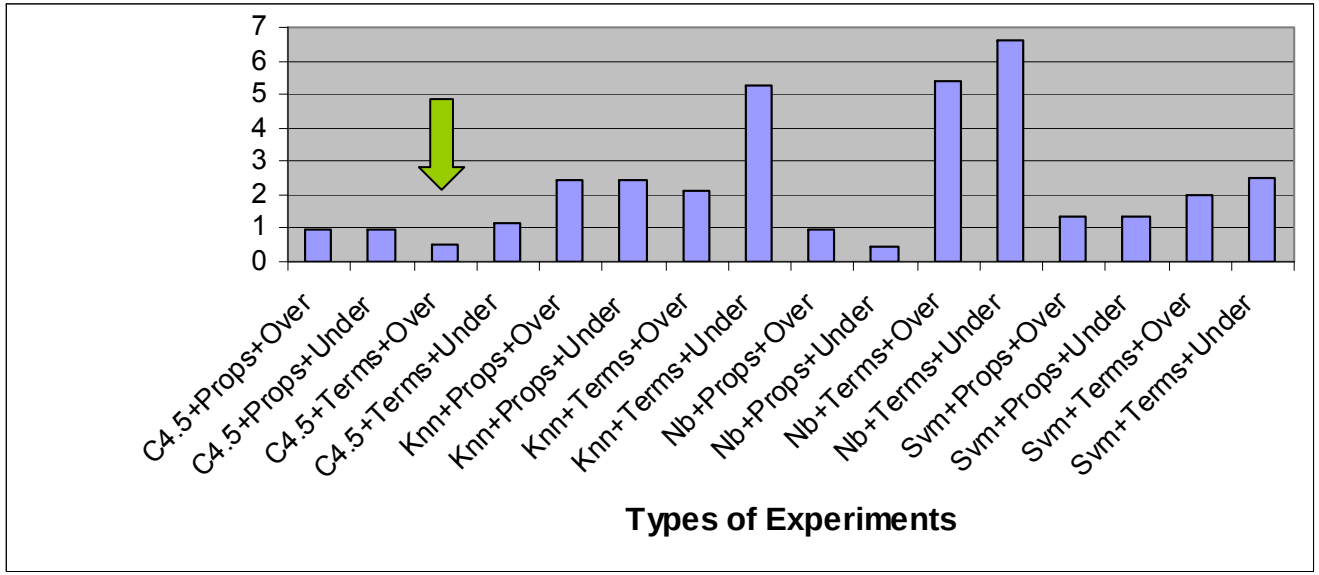
Closer Look





- R1-R2 Category
  - Different conceptualizations
  - *CSR* locates subsumptions that the reasoner cannot infer (R2), without using equivalencies

"Confused" Equivalencies



- *CSR* is very tolerant in "confusing" equivalence relations as subsumption ones
- Without using them also as input
- Can be used for filtering

Conclusions

- *CSR method:*
  - Learns **patterns of concepts' features** (properties or terms) that provide evidence for the subsumption relation among these concepts, using machine learning techniques
  - **Generates training datasets** from the source ontologies specifications
  - Tackles the **problem of imbalanced training datasets**
  - **Generalizes effectively** over the training examples
  - Does **not** exploits **equivalence mapping** (words case as features)
  - Does **not** easily "confuse" equivalence mappings as subsumption ones
  - Is independent of **external resources**

Questions? Comments?

Thank you!

Related Work

1. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic Matching: Algorithms and implementation. Journal on Data Semantics, IX (2007)
2. Aleksovski, Z., Klein, M., Kate, W, Harmelen F.: Matching Unstructured Vocabularies Using a Background Ontology. In: EKAW, Podebrady, Czech Republic (2006)
3. Gracia, J., Lopez, V., D'Aquin, M., Sabou, M, Motta, E., Mena, E.: Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching, In: Ontology Matching Workshop, Busan, Korea (2007)
4. Bouquet, P., Serafini, L., Zanobini, S., and Sceffer, S. 2006: Bootstrapping semantics on the web: meaning elicitation from schemas. In: WWW, Edinburgh, Scotland (2006)
5. Cimiano P., Staab, S.: Learning by googling, In: SIGKDD Explor. Newsl., USA (2004)
6. Hage, W.R. Van, Katrenko, S., Schreiber, A.Th.: A Method to Combine Linguistic Ontology Mapping Techniques, In: ISWC, Osaka, Japan (2005)
7. Risto G., Zharko A., Warner K.: Using Google Distance to weight approximate ontology matches. In: WWW, Banff, Alberta, Canada (2007)
8. Jerome D., Fabrice G., Regis G., Henri B.: An interactive, asymmetric and extensional method for matching conceptual hierarchies. In: EMOI – INTEROP Workshop, Luxem-bourg (2006)

Related Work

- **Satisfiability Based Approaches** [1]
  - Transformation of the ontology mapping problem in a satisfiability one
- **Exploitation of Domain Knowledge** [2], [3], and [4]
  - Exploit domain ontologies as an intermediate ontology for bridging the semantic gap [2], [3]
  - WordNet is used for the same purpose (WordNet Description Logics) [4]
- **Google Based Approaches** [5], [6], and [7]
  - Exploit the hits returned by Google to test if subsumption relation holds [5], [6] or to loosen the formal constrains [7]
- **Machine Learning Approaches** [8]
  - A method based on Implication Intensity theory (Unsupervised Learning) is proposed