Instance Based Clustering of Semantic Web Resources

Gunnar Aastrand Grimnes

Peter Edwards & Alun Preece

DFKI GmbH, Kaiserslautern
School of Natural & Computing Sciences, University of Aberdeen
School of Computing Science, Cardiff University

Introduction

- We want to apply machine learning techniques to semantic web data
- In this work we focus on the *clustering* of Semantic Web resources – for example for:
  - Visualisation, information-mining, user-recommendations, etc.

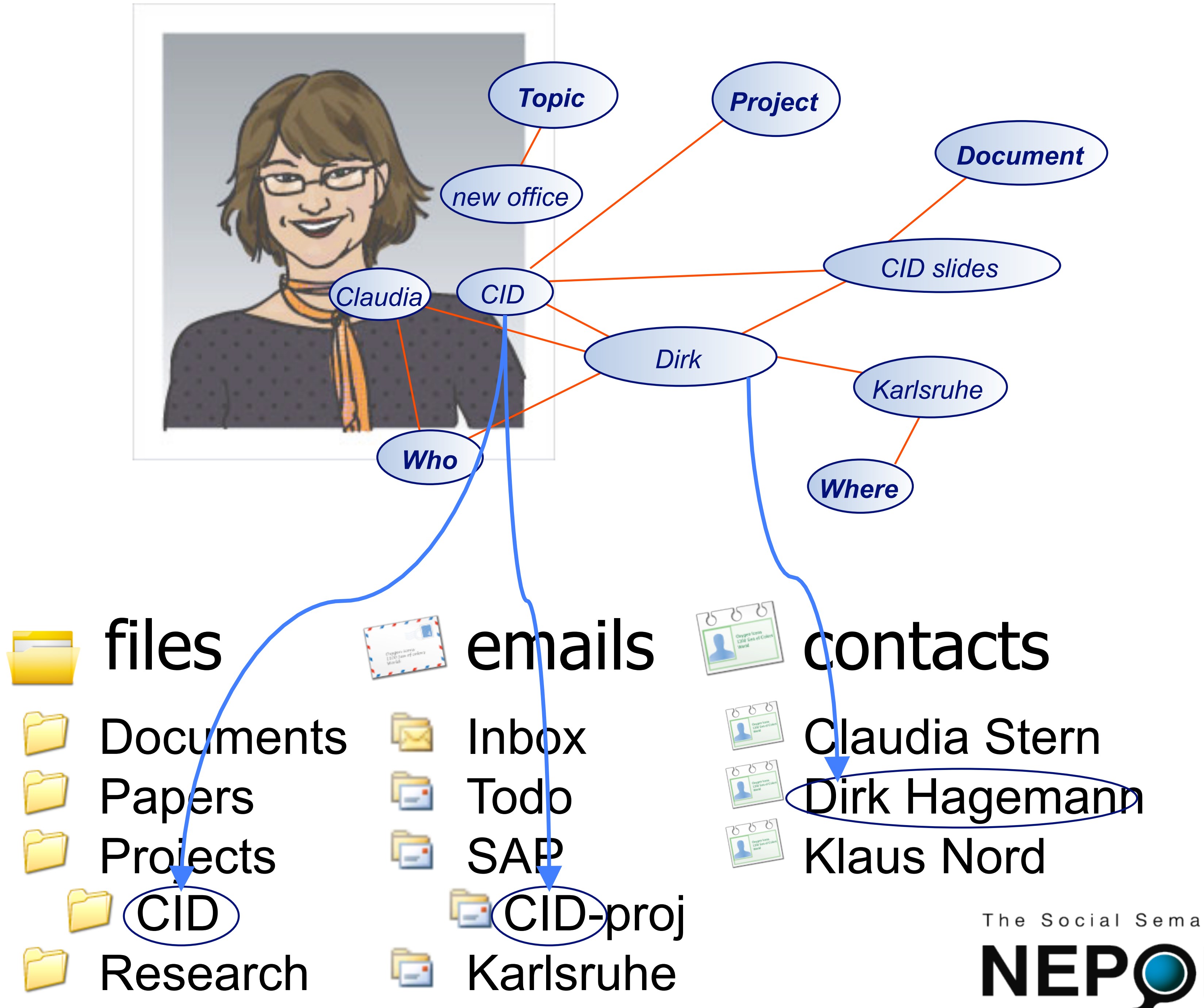
Machine Learning from RDF

- Most people learn to enable the semantic web – we assume it already exists and want to learn more from the semantic data
- Representation is king – finding the "correct" mapping of the RDF Graph to the input format for ML is crucial

Data-sets

- We test on three different data-sets:
  - FOAF Crawl – 3755 persons
  - **NEPOMUK** PIMO – 1809 instances
  - Citeseer dump – 4220 papers

Datasets – PIMO



files

- Documents
- Papers
- Projects
  - CID
- Research

emails

- Inbox
- Todo
- SAP
  - CID-proj
- Karlsruhe

contacts

- Claudia Stern
- Dirk Hagemann
- Klaus Nord

The Social Semantic Desktop

NEPOMUK

Datasets - Citeseer

```
citeseer:shannon48 a :article;
    :journal "Bell System Technical Journal";
    :month "July, October";
    :title "A Mathematical Theory of Communication";
    :volume "27";
    :year "1948".
```

Extracting Instances

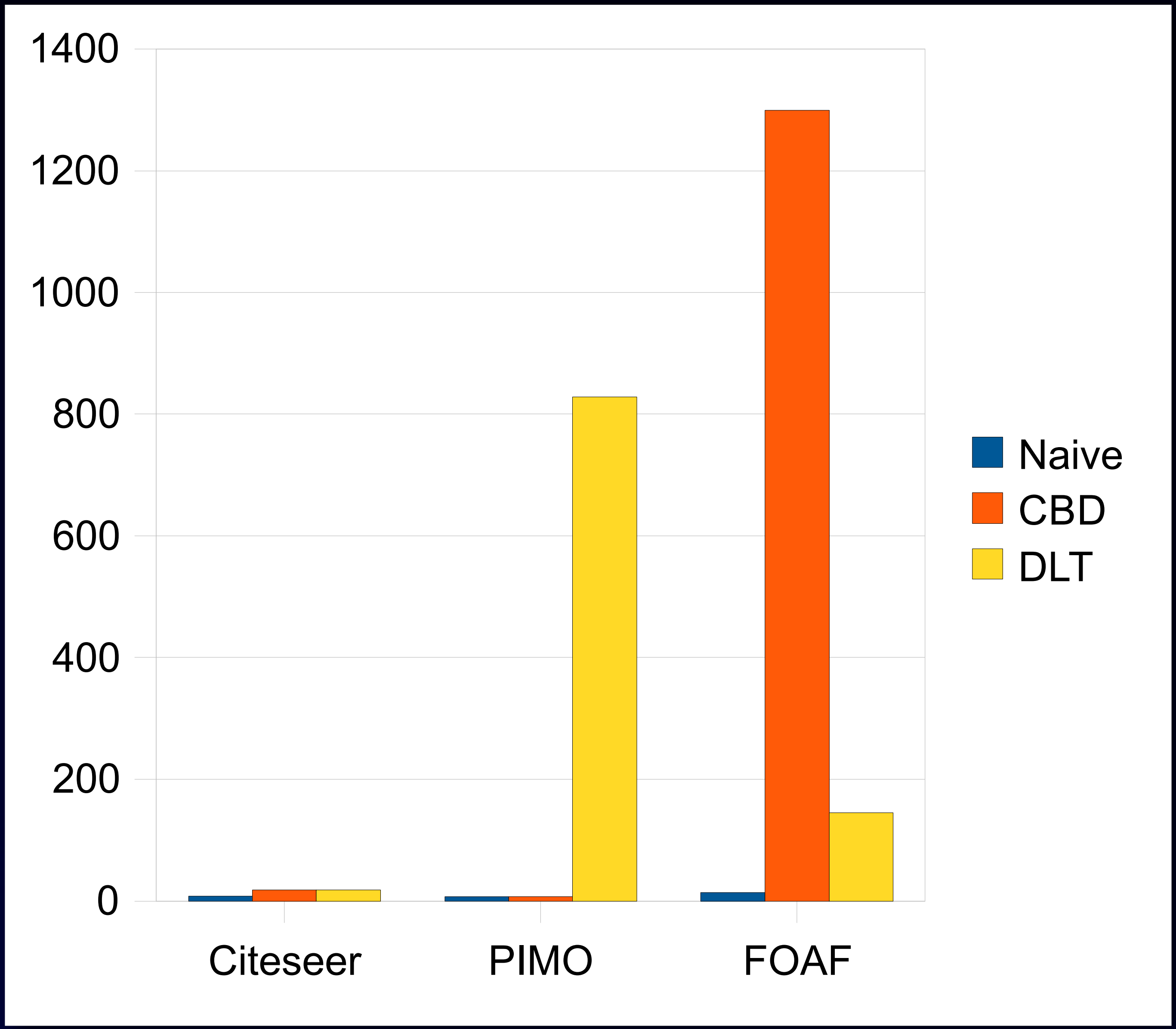
- We are interested in *instance based* clustering...
- ... but Semantic Web data is one big graph
- What part of the graph is relevant to a resource?
- Relevant also for UI creation, SPARQL Describe + +

Instance Extraction

Three approaches

- Immediate properties
- Concise Bounded Description
- Depth Limited Crawling

Extracted Instance Graphs



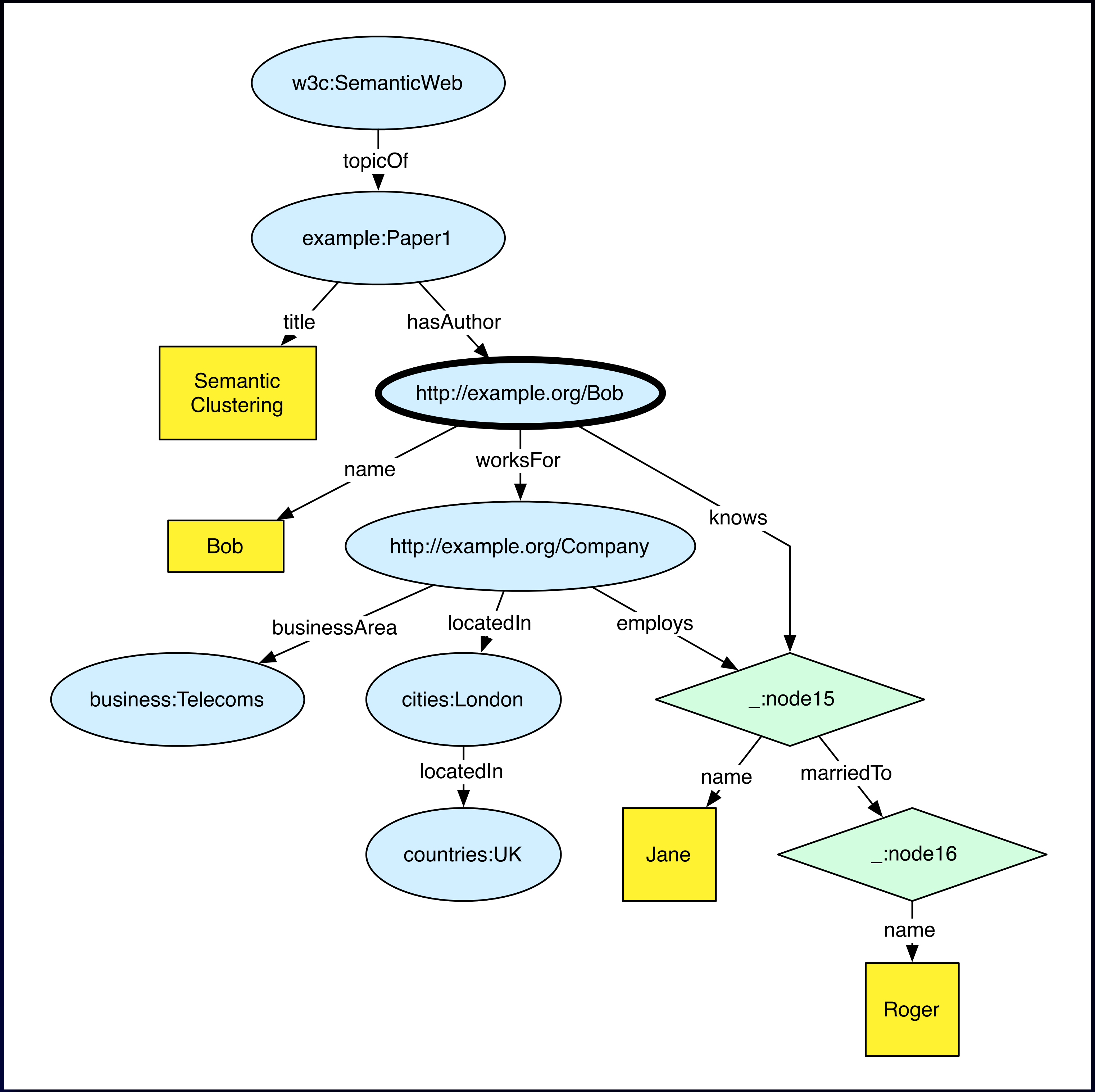
Distance Measure

- Given some RDF "instances" (i.e. resource + relevant graph) how can we compute distances between them?
- Tricky to transform RDF into some N-space for Euclidian distance
- Again three approaches:
  - feature-vector, graph-based & ontological

Feature Vector Distance

- How can we extract a feature vector from an RDF graph?
- Naive solution: make a feature for each property
  - Does not handle deeper relations in graph
- We do slightly better – create features for all paths in the data
  - Limit to top X paths occurring in the data

Feature Vector Example



[name, worksFor, knows,
worksFor→businessArea,
worksFor→locatedIn,
knows→name,
knows→marriedTo,
worksFor→locatedIn→locatedIn,
knows→marriedTo→name]

[{“bob”}, { ex:TheCompany },
{ :node15 }, { business:Telecoms },
{ cities:London }, {“Jane”},
{ :node16 }, { countries:UK },
{“Roger”}]

Feature Vector Distance

- Distance for features *FV* and vectors *X* & *Y:*

$$simFV(X,Y,FV) = \frac{1}{|FV|}\sum_{f\in FV}\frac{2 * |X_f \cap Y_f|}{|X_f| + |Y_f|}$$

Graph Based Distance

- Combination of level of overlap of nodes and edges
- Designed for conceptual graphs, but works fine with RDF graphs

M. Montez-y-Gómes, A. Gelbukh and A. López-López: Comparison of Conceptual Graphs, 2000.

Ontological Distance

- Made for formal ontologies, minor modifications needed for noisy semantic web data:
  - Multiple super-classes / types
  - Well defined range/domains
  - Distinction between object/literal properties
- Combination of taxonomy similarity, attribute similiarity & relational similarity
- Works directly on RDF graph

A. Maedche and V. Zacharias: Clustering Ontology-based Metadata in the Semantic Web, 2002.

Experiments

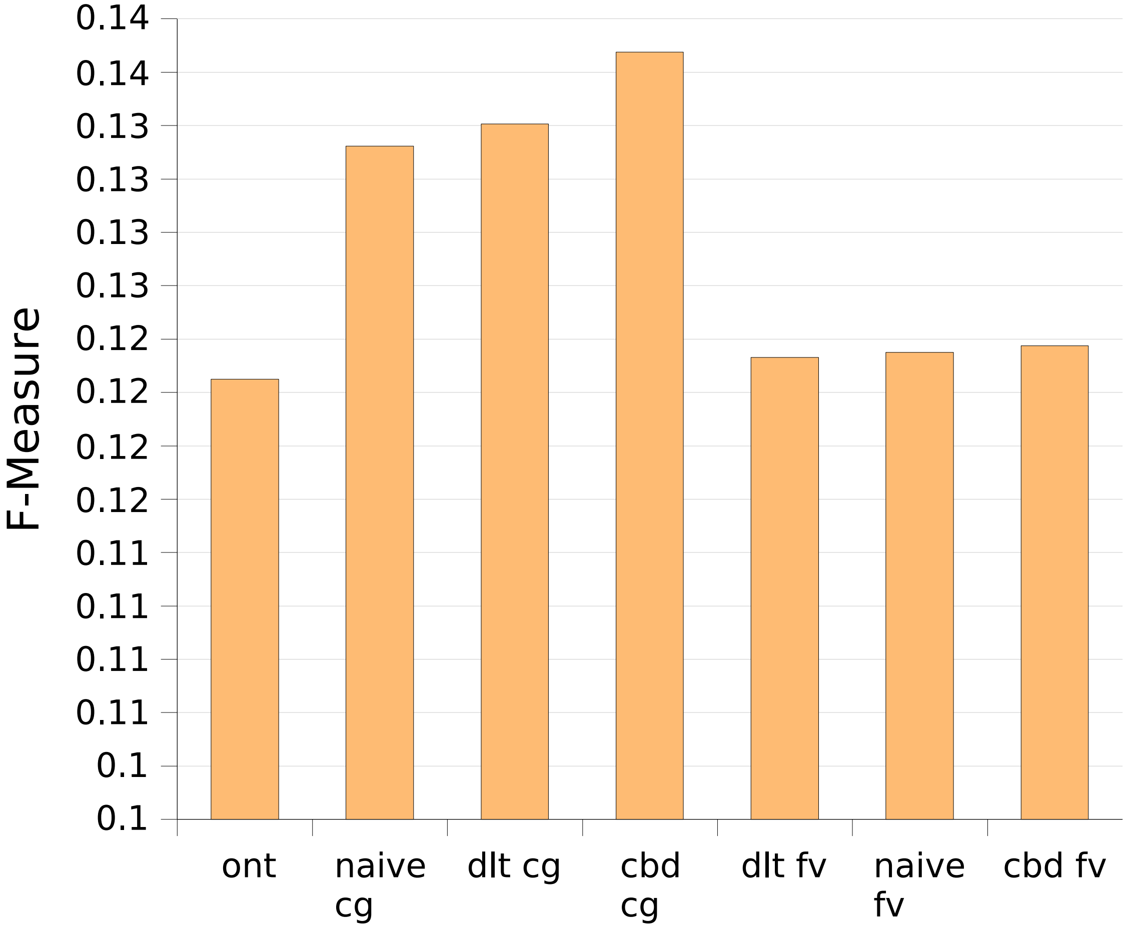
- We used a very simple HAC algorithm
- Supervised evaluation for Citeseer and PIMO data:
    - F-measure, Heß measures, entropy and purity
- Unsupervised evaluation for FOAF & PIMO:
    - Zamir's Quality Metric

Results

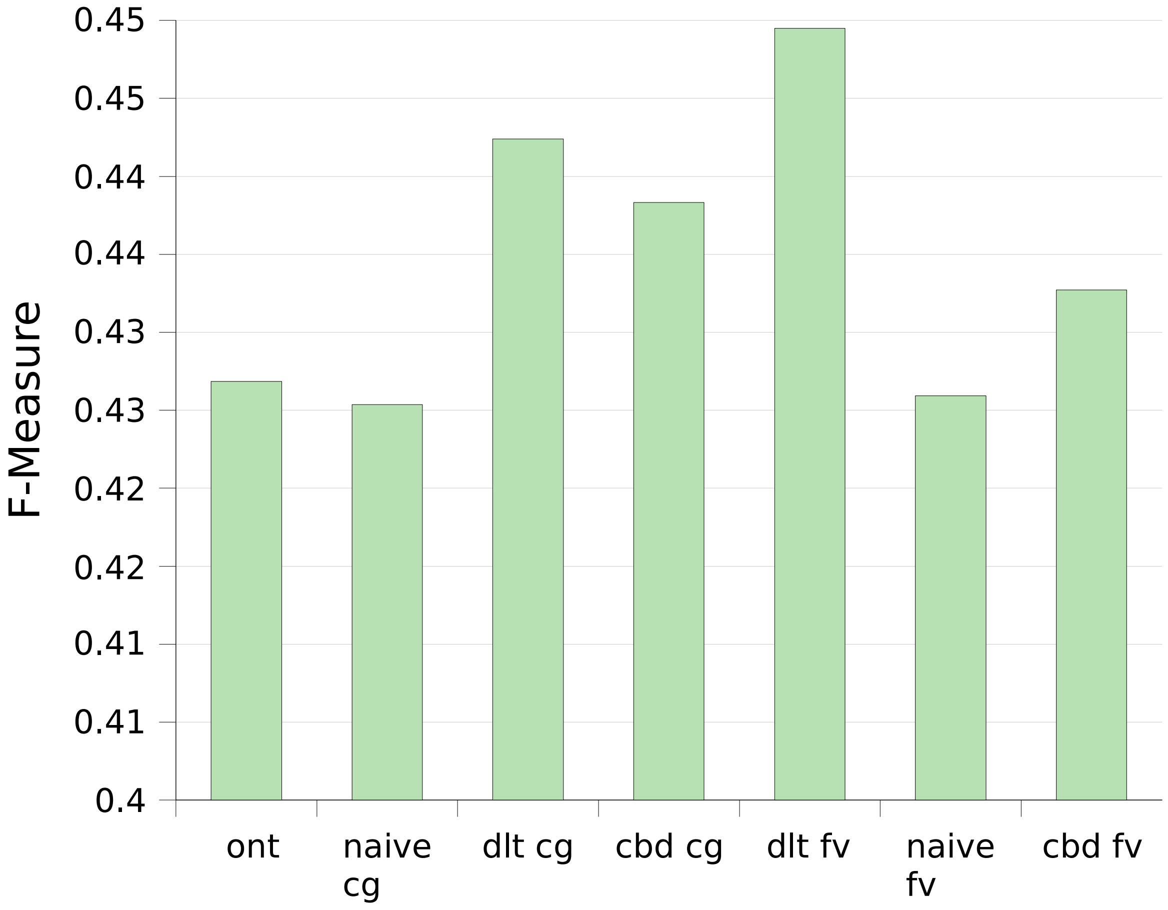
- Very uneven cluster sizes – all solutions had several singleton clusters
    - A more sophisticated clustering algorithm may be in order
    - Feature-vector based approach especially bad – largest cluster contained 85% of all instances

Supervised Results I

Citeseer

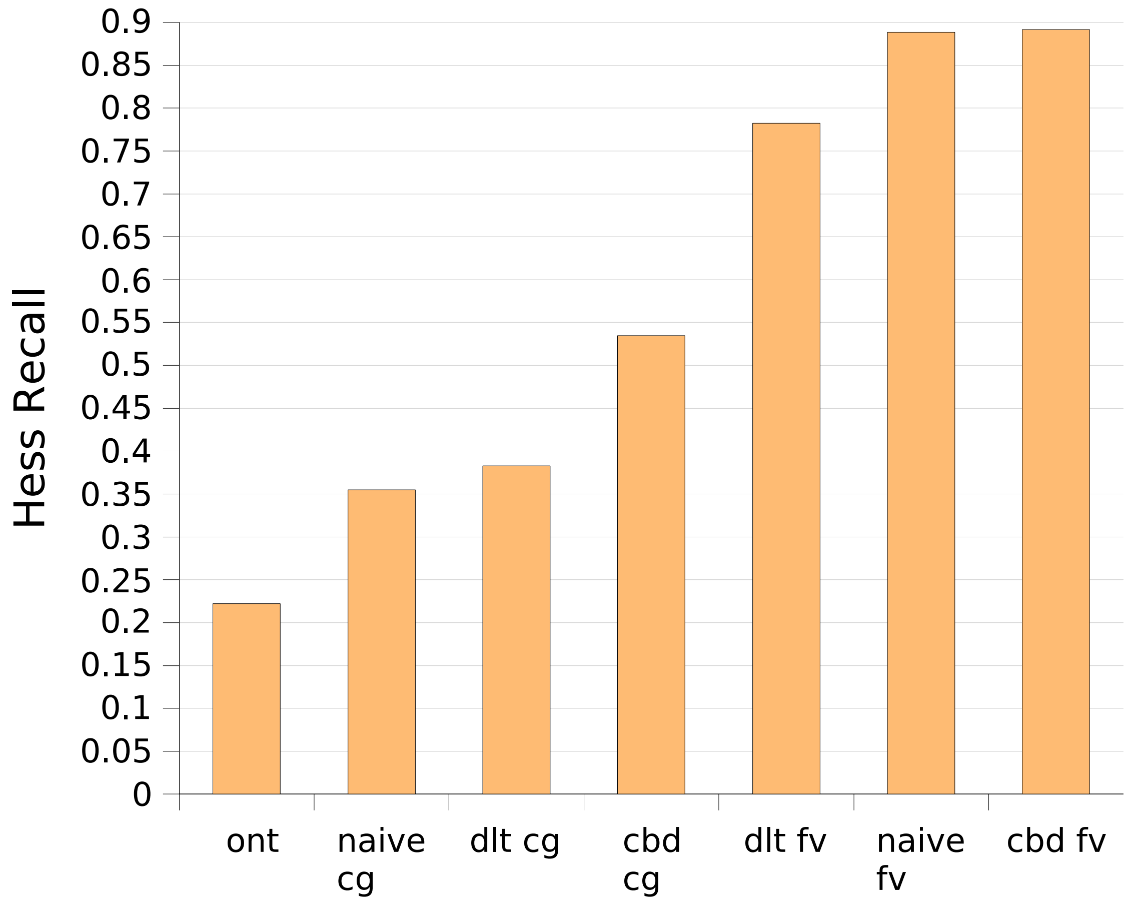


PIMO

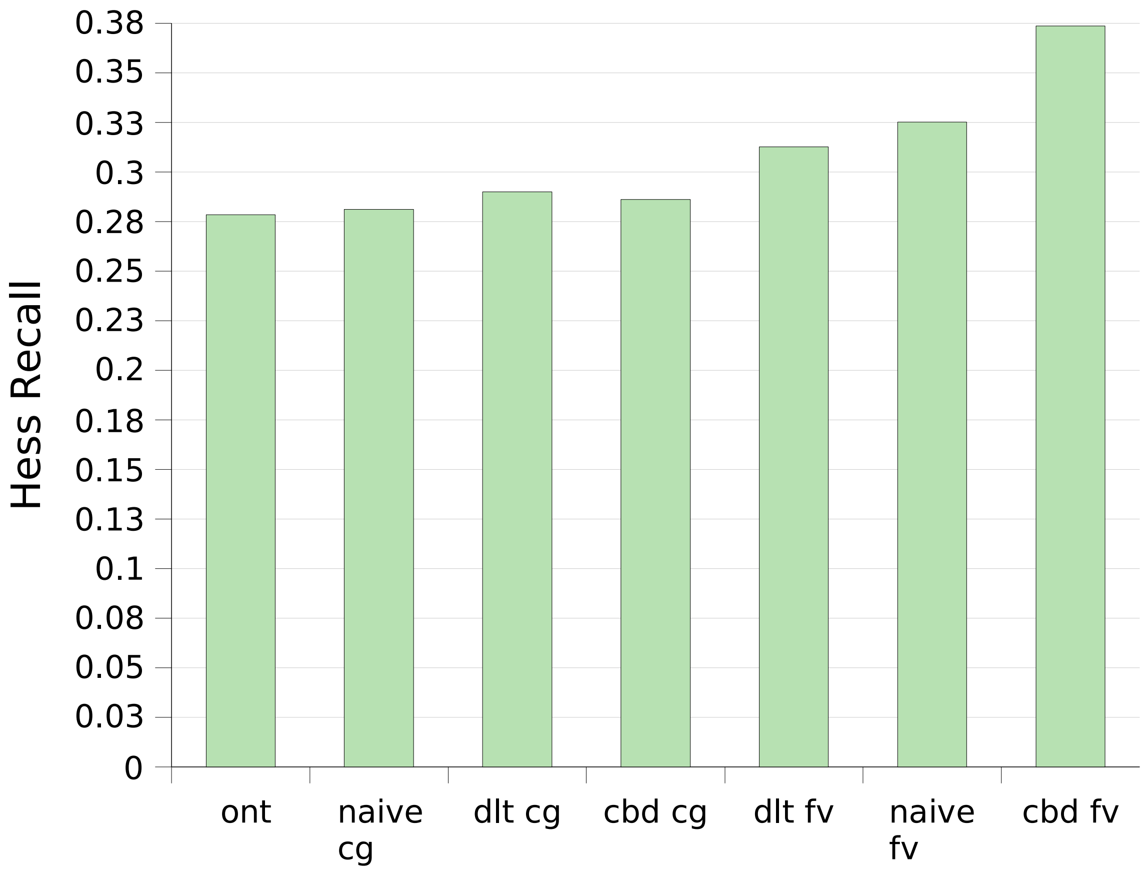


Supervised Results II

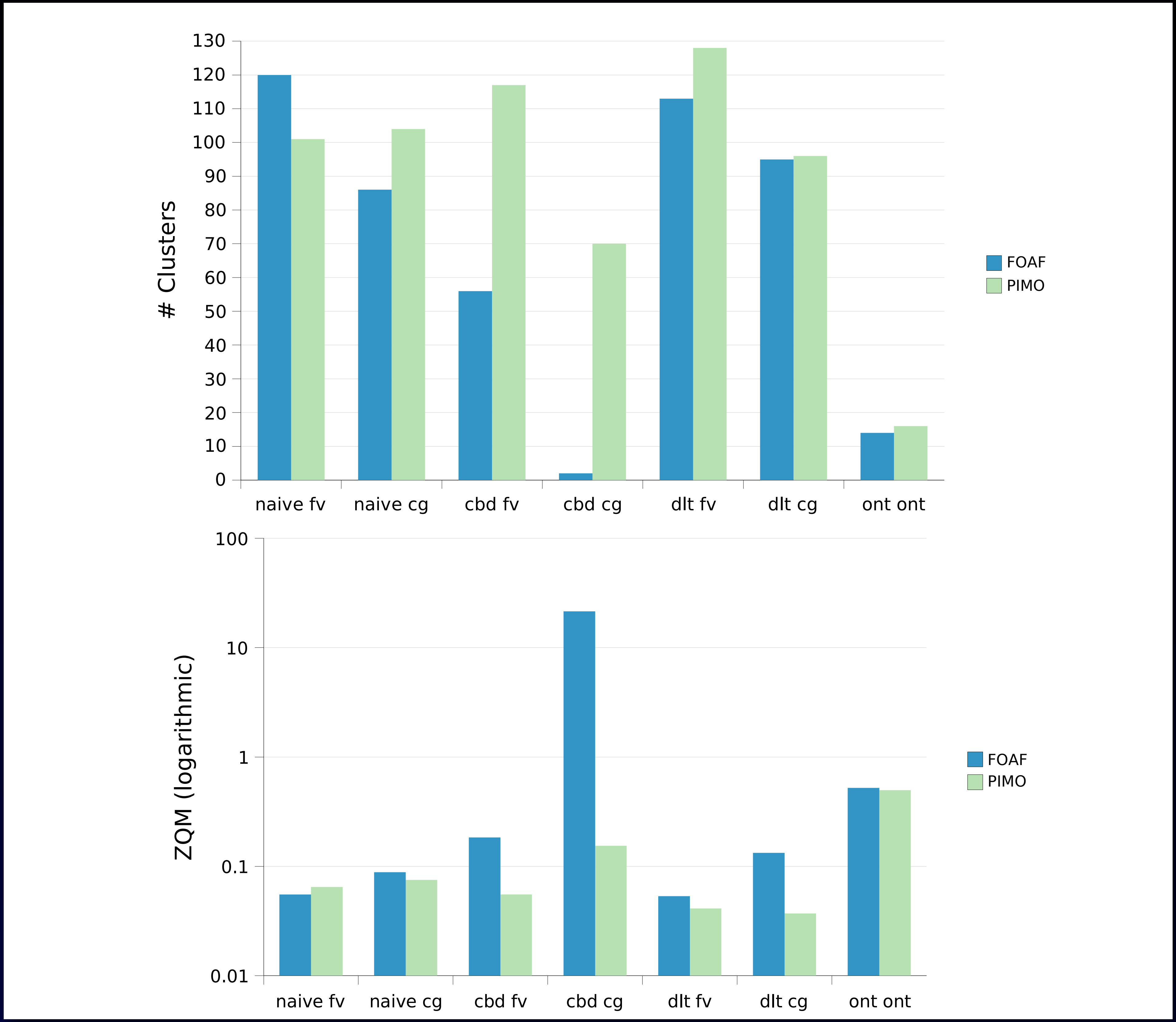
Citeseer



PIMO



Unsupervised Results



Conclusions

- Ultimately “it depends” :)
  - On the features of the data-set
  - Mainly on the (here non-existent) *application!*
- Ontological distance measure is quite slow to compute – and does not perform significantly better – but it may for data with better ontologies

Future Work

- Hybrid instance extraction method
  – combine node-type and depth limit ...
- ... or a frequency based instance extraction method
- Find a specific application!
  - Which hopefully could also give us more natural data

Thanks for you attention!

Questions?