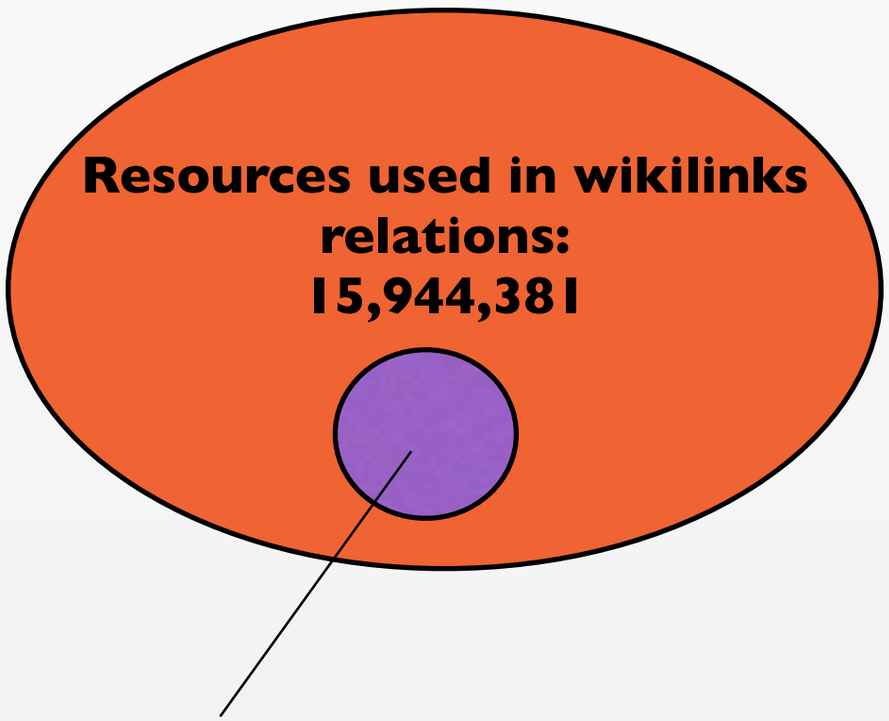
Type inference through the analysis of Wikipedia links

Andrea Giovanni Nuzzolese
nuzzoles@cs.unibo.it
Aldo Gangemi
aldo.gangemi@cnr.it
Valentina Presutti
valentina.presutti@cnr.it
Paolo Ciancarini
ciancarini@cs.unibo.it

Outline

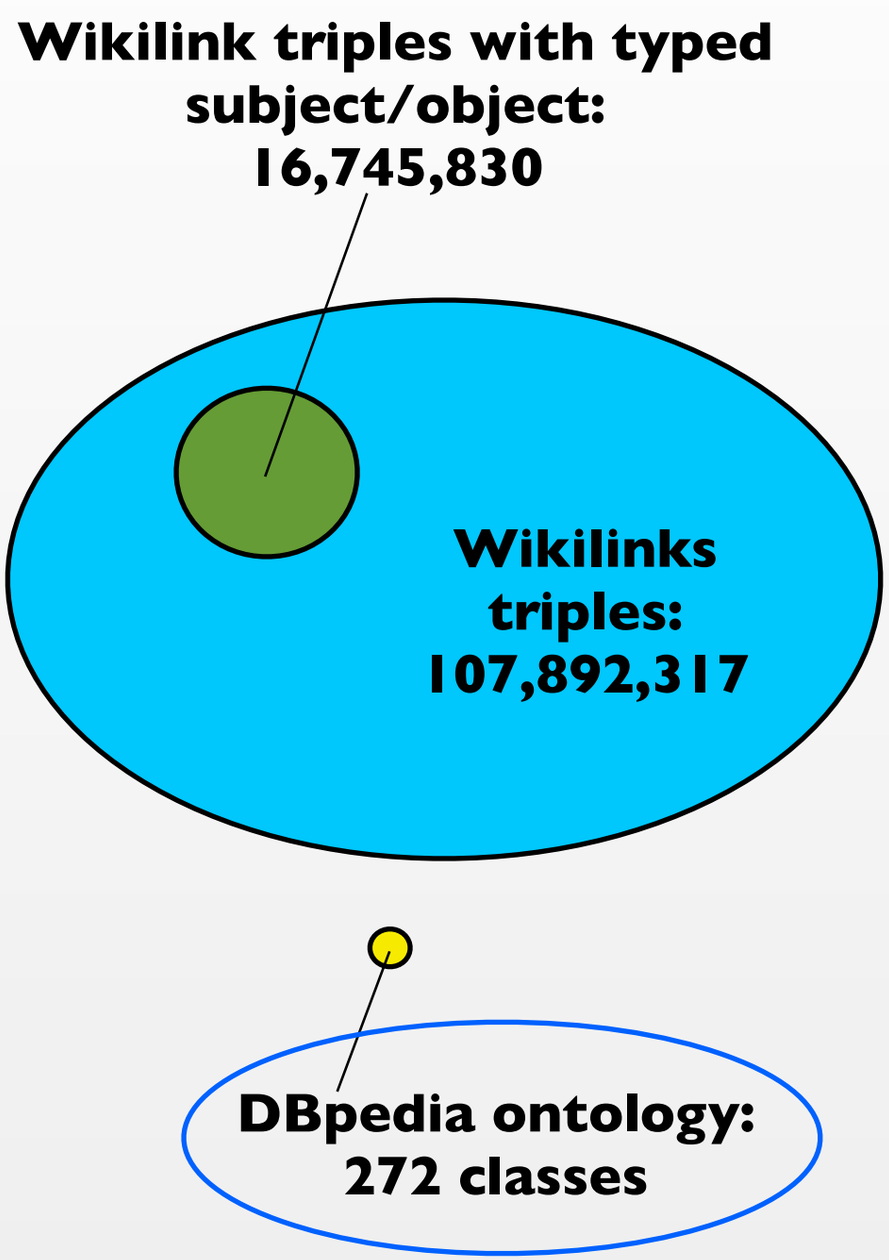
- **Motivations**
- **Materials**
- **Applied methods**
- **Results**
- **Conclusions**

Motivations



- Only a subset of the DBpedia resources is typed with the DBpedia ontology (DBPO)
- The typing procedure is top-down.
- Is the DBPO complete with respect to the DBpedia domain?
- How good and homogeneous is the granularity of DBPO types?

Materials

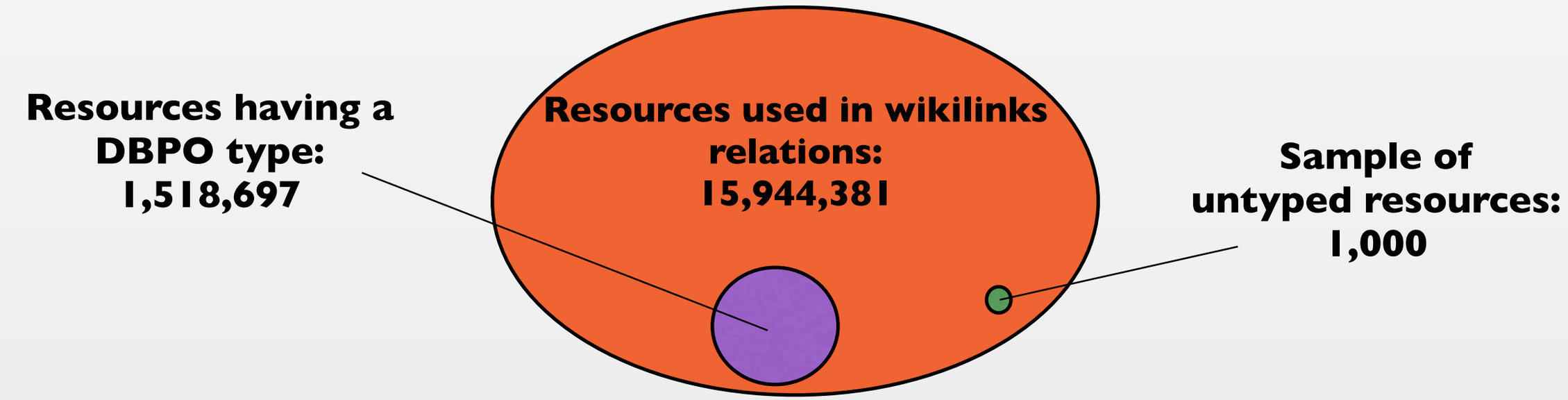


DBpedia 3.6

Dataset	# of triples
wikilink triples	107,892,317
infobox mapping-based "data" triples	9,357,273
rdfs:label triples	7,972,225
rdf:type triples	6,173,940
infobox mapping-based "object" triples	4,251,239

What we did

- **Wikilinks of a DBpedia resource convey knowledge that can be used for classifying it.**
- **Classification methods**
  - Inductive learning: k-Nearest Neighbor algorithm
  - Abductive classification based on *EKPs* [1] and *homotypes* used as background knowledge
- **The methods were performed on**

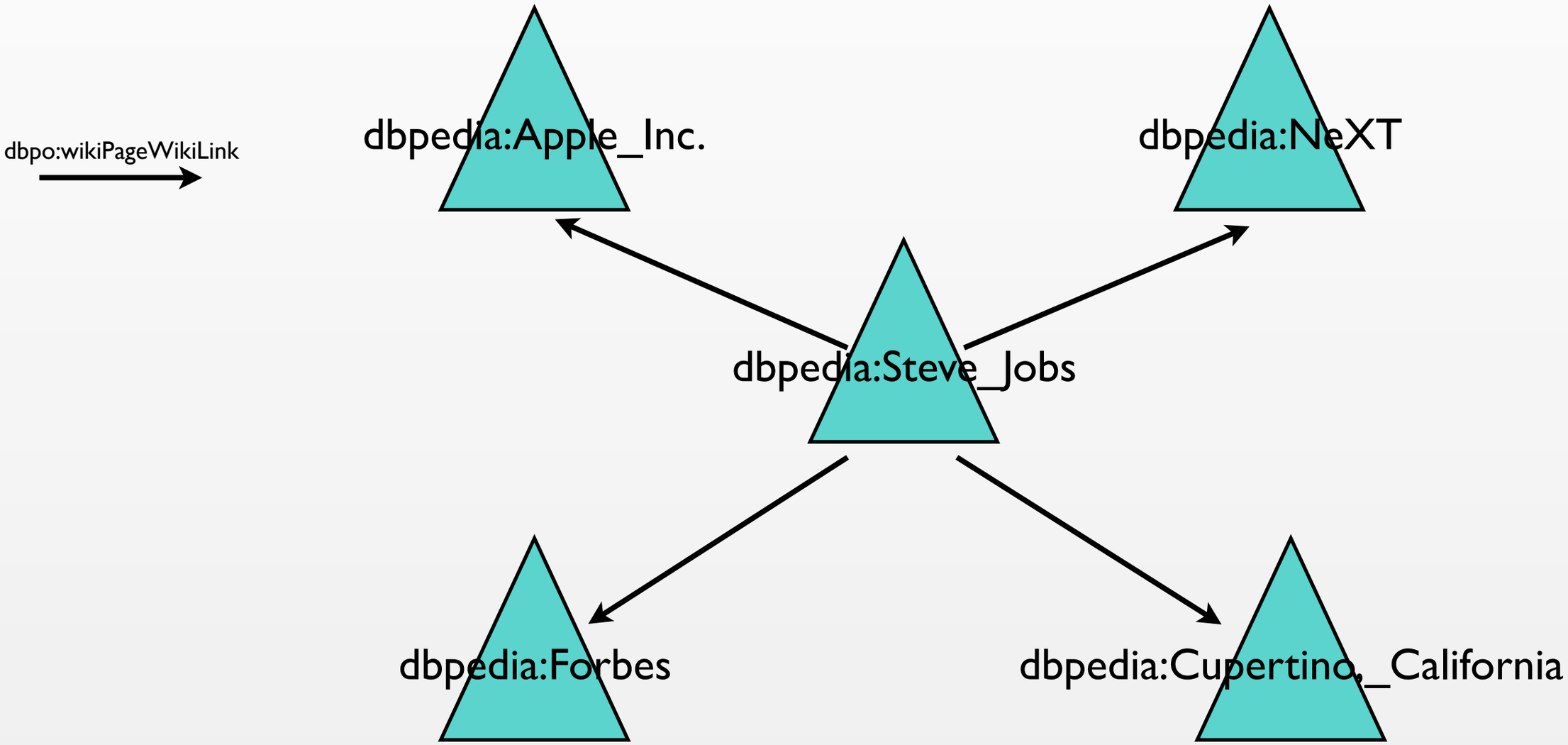


[1] A. G. Nuzzolese, A. Gangemi, V. Presutti, and P. Ciancarini. Encyclopedic Knowledge Patterns from Wikipedia Links. In L. Aroyo, N. Noy, and C. Welty, editors, Proceedings of the 10th International Semantic Web Conference (ISWC2011), pages 520-536. Springer, 2011.

Inductive classification

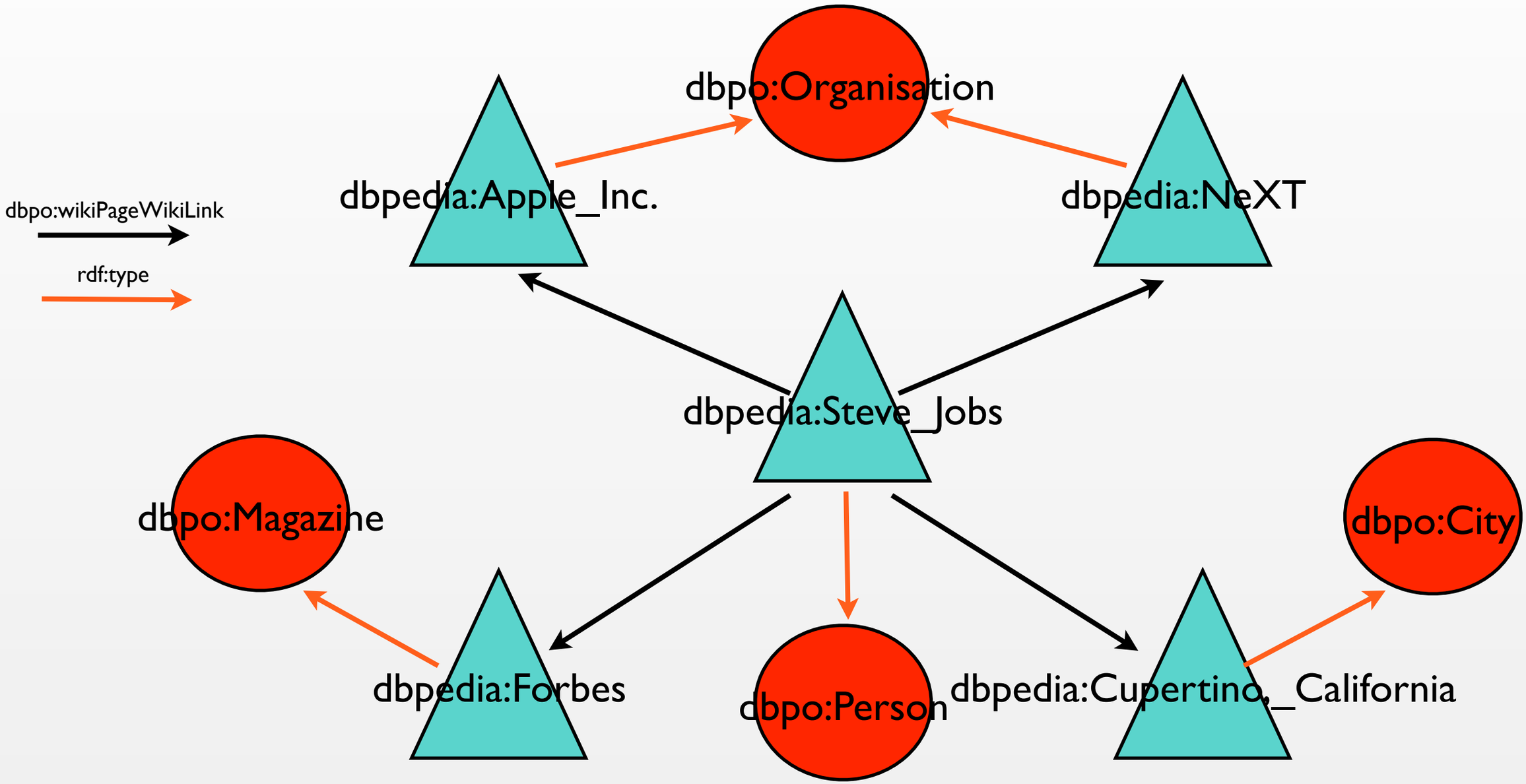
- **We designed two inductive classification experiments based on the *k*-NN algorithm**
  - on 272 features, i.e., all the classes in the DBPO
  - on 27 features, i.e., the top-level classes in the DBPO hierarchy
- **For each experiment we built a labeled feature space model as training set by using a randomly sampled 20% of typed resources**
  - the algorithms were tested on the remaining 80% of typed resources

Building the training set for K-Nearest Neighbor algorithm



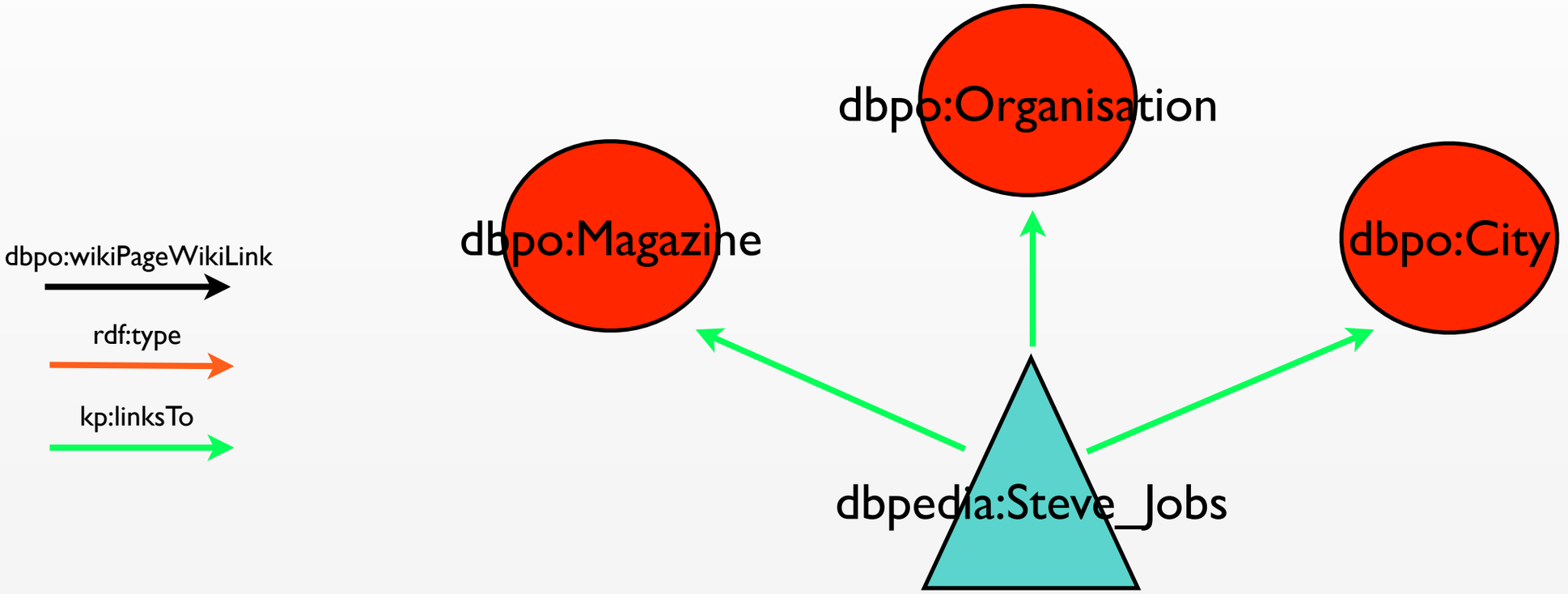
	Mammal	Scientist	Company	Drug	City	Magazine	*Class*
dbpedia:Steve_Jobs							
							...

Building the training set for K-Nearest Neighbor algorithm



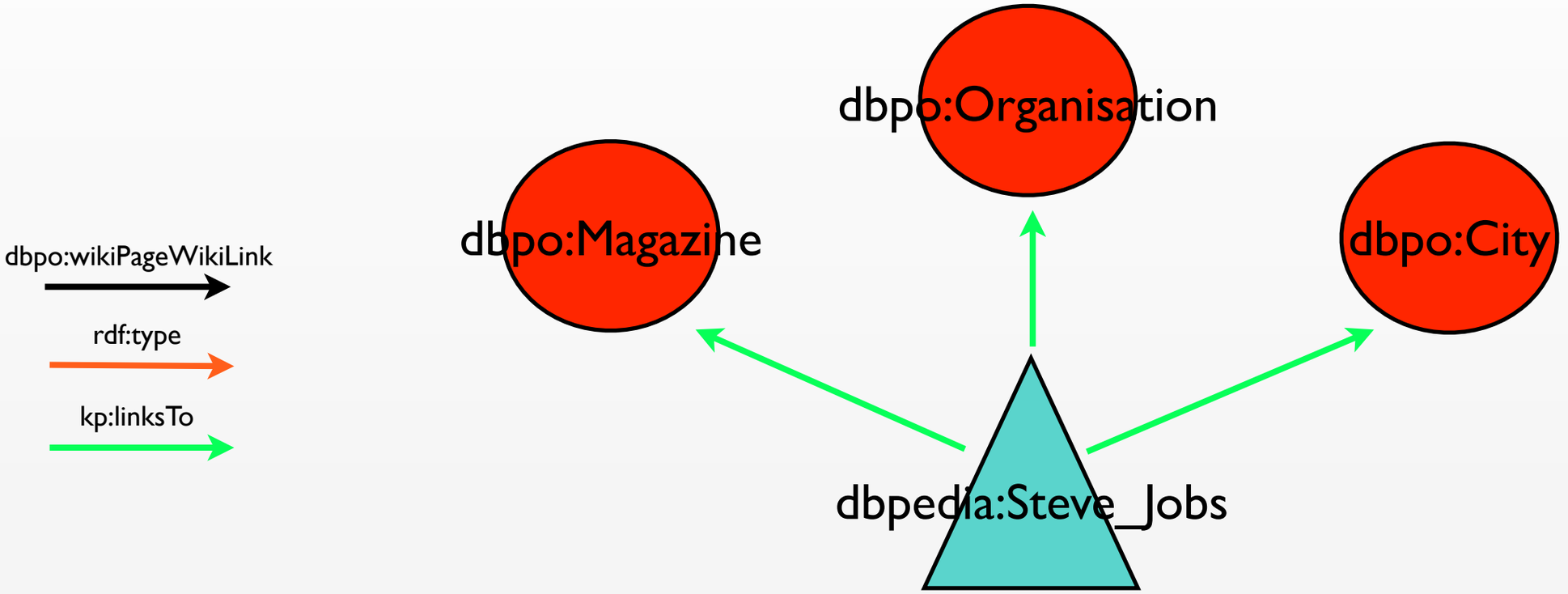
	Mammal	Scientist	Company	Drug	City	Magazine	*Class*
dbpedia:Steve_Jobs							dbpo:Person
							...

Building the training set for K-Nearest Neighbor algorithm



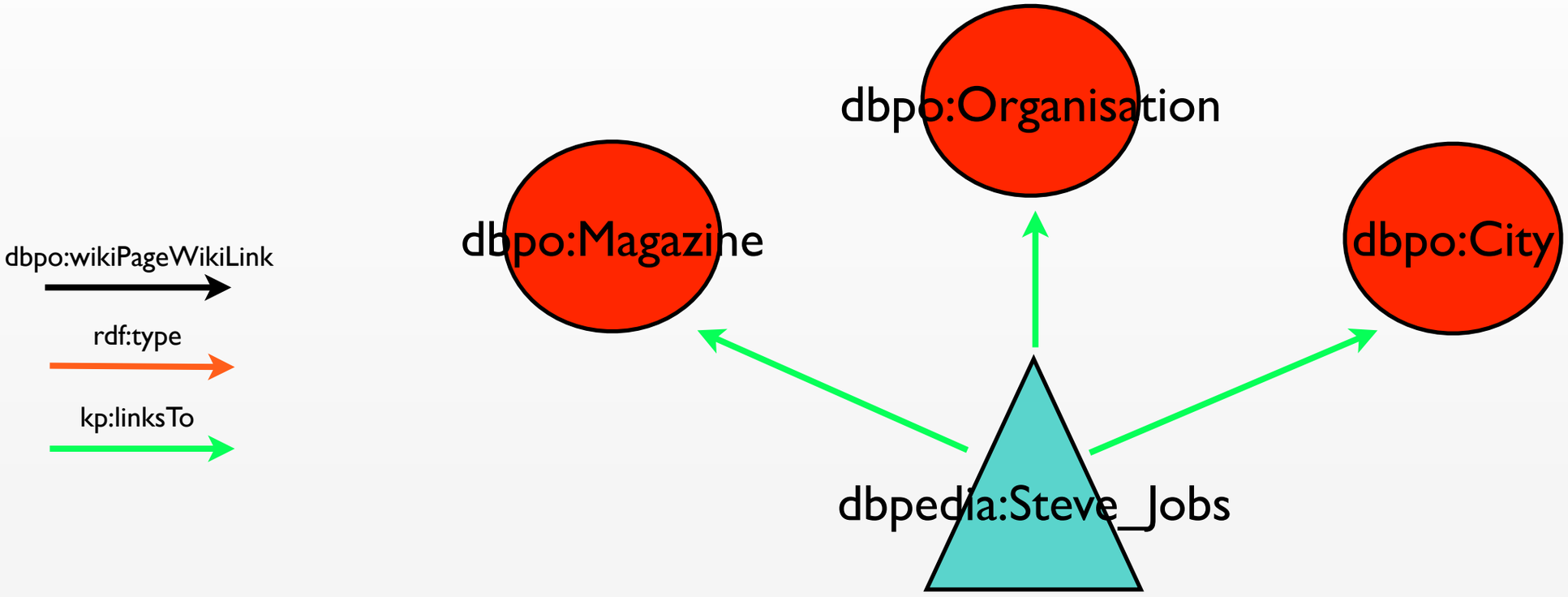
	Mammal	Scientist	Company	Drug	City	Magazine	*Class*
dbpedia:Steve_Jobs	0	0	1	0	1	1	dbpo:Person
							...

Building the training set for K-Nearest Neighbor algorithm



	Mammal	Scientist	Company	Drug	City	Magazine	*Class*
dbpedia:Steve_Jobs	0	0	1	0	1	1	dbpo:Person
...	...	...	...	...	...	...	...

Building the training set for K-Nearest Neighbor algorithm



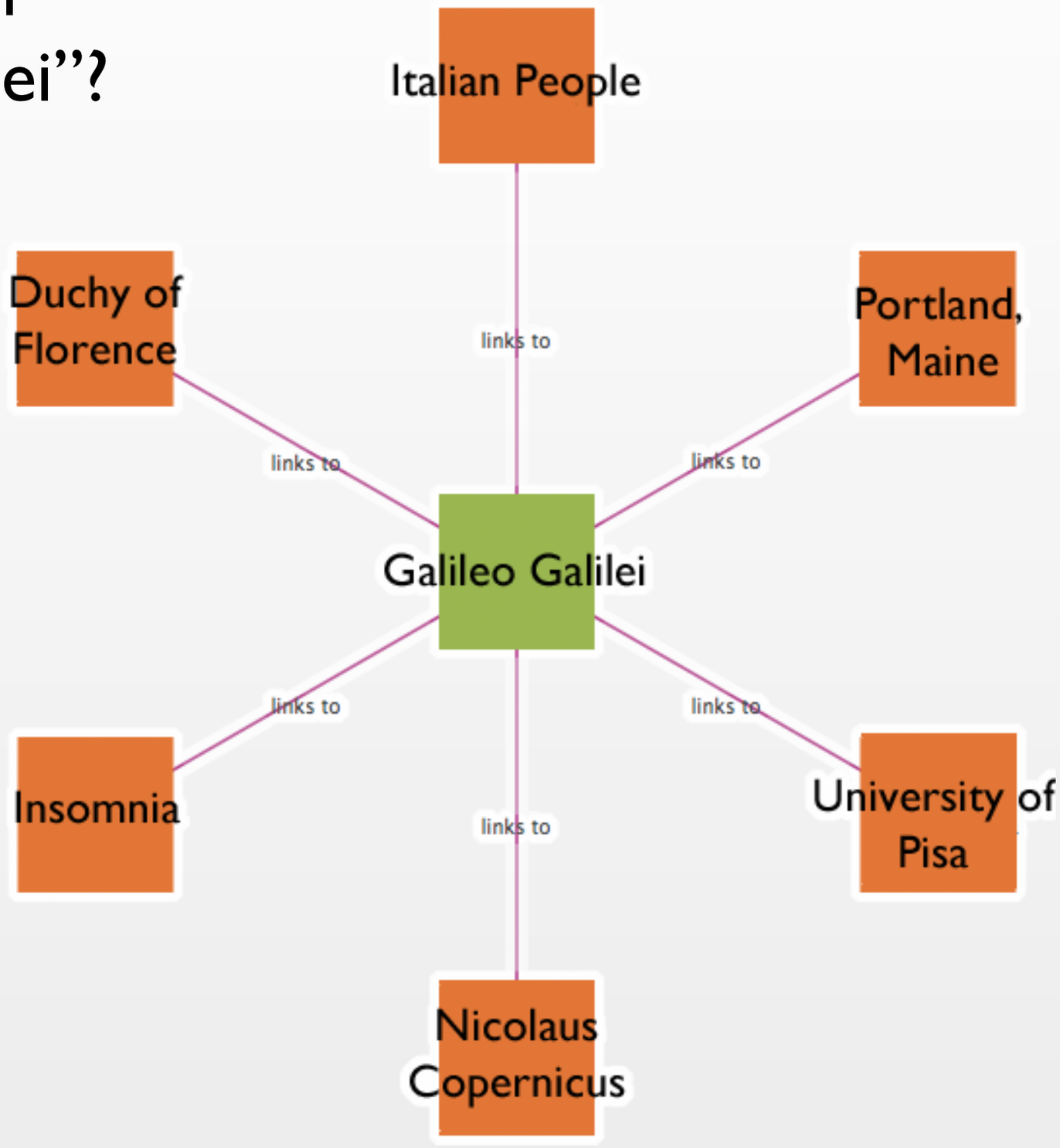
- Precision using all DBPO types as features: 31.65%
- Precision using the top-level of DBPO as features: 40.27%

Abductive classification with EKPs

- **EKPs**
  - A EKP of a certain entity type is a small vocabulary that captures the core types used for describing such entity type as it emerges from the Wikipedia crowds

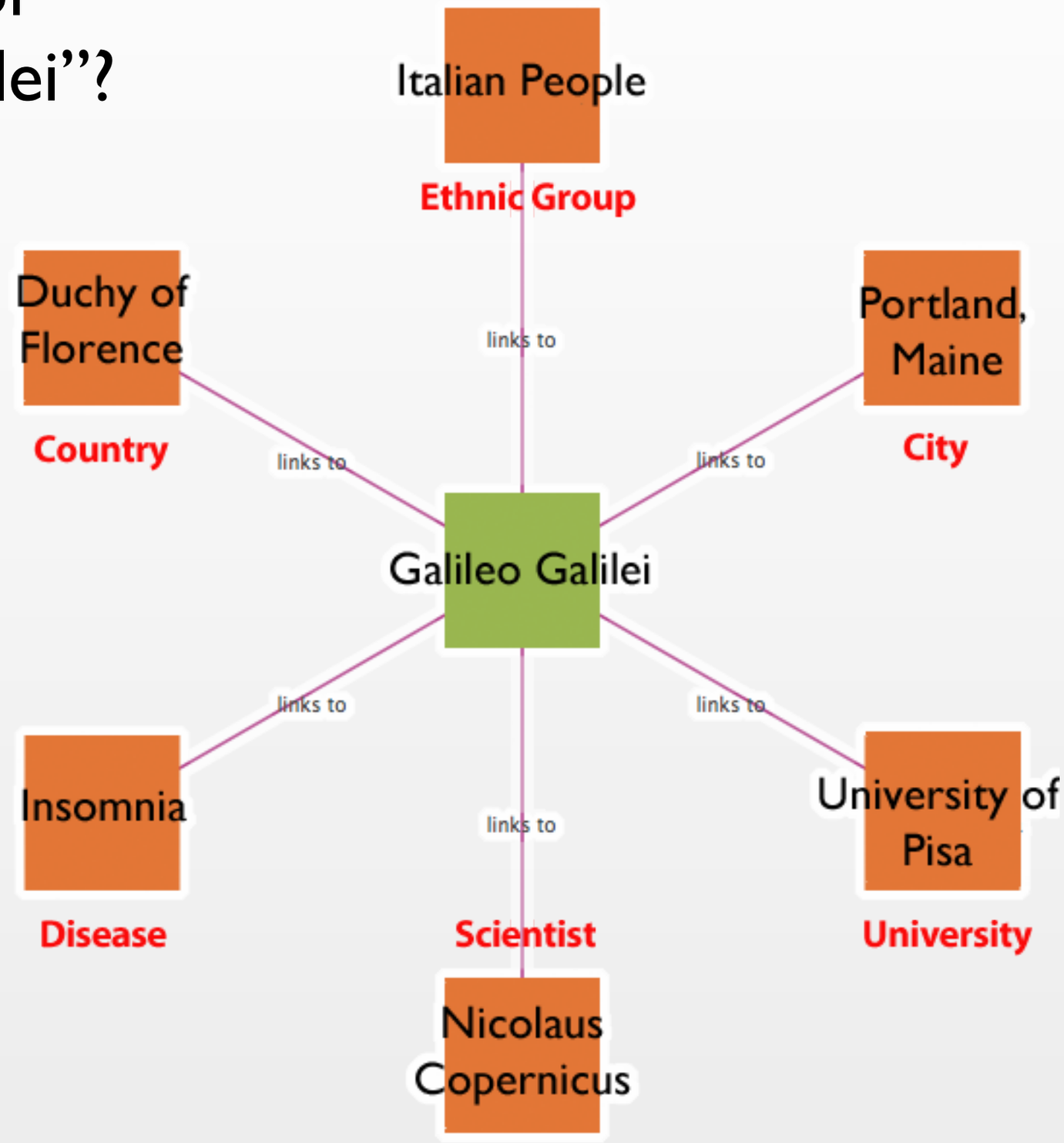
*visit **aemoo.org** for an exploratory tool based on EKPs*

How can we infer the type of "Galileo Galilei"?



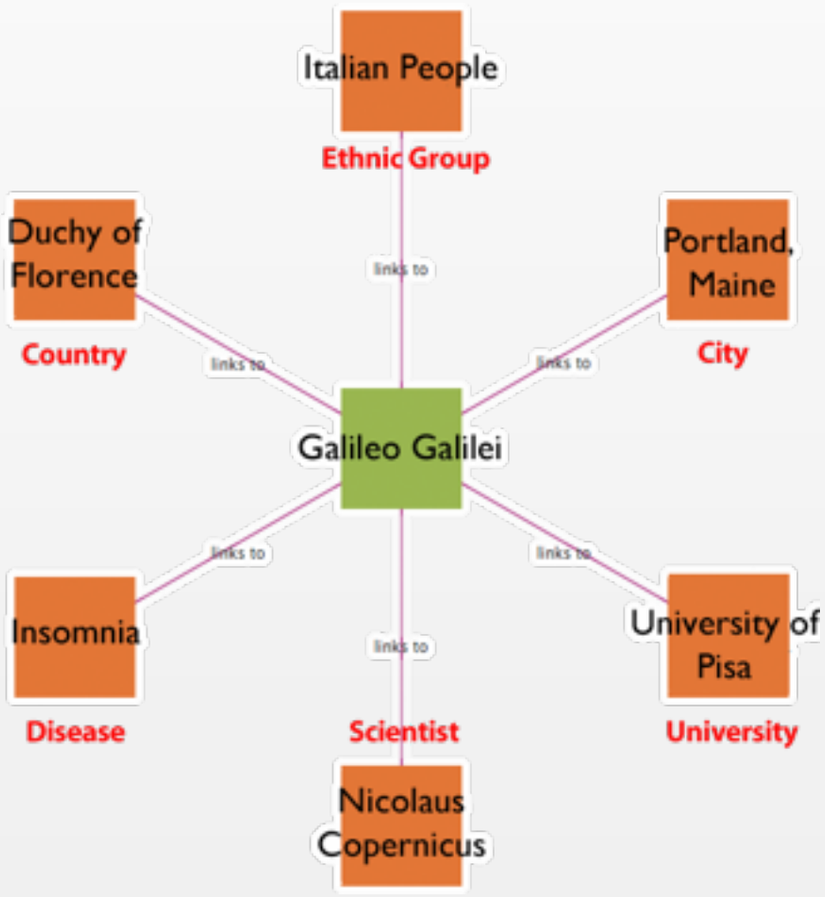
How can we infer the type of "Galileo Galilei"?

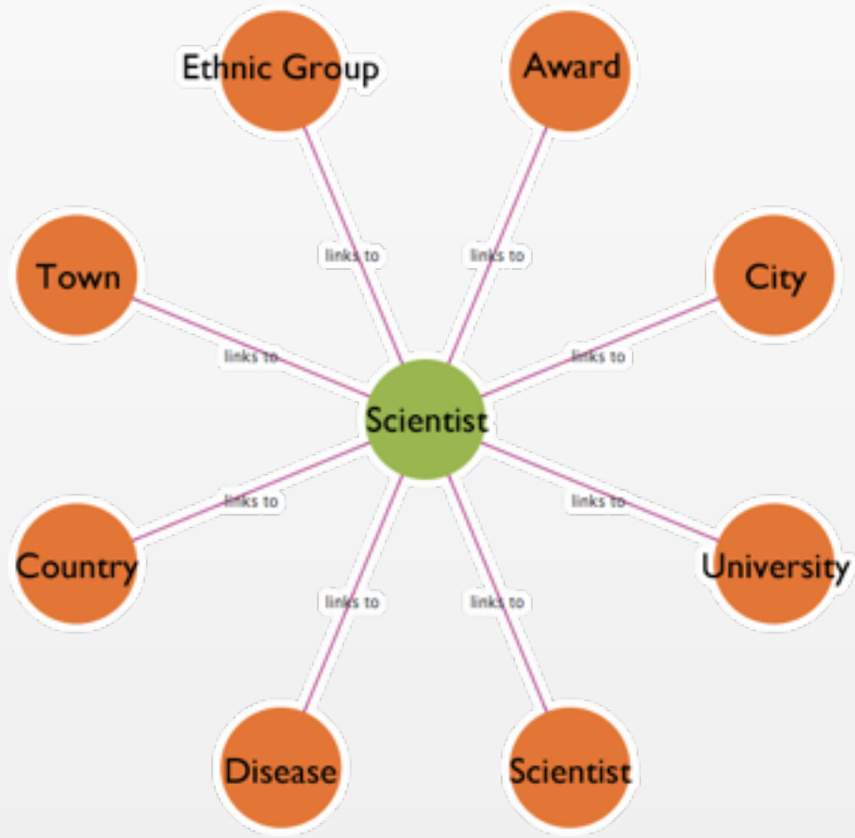
We know its path types



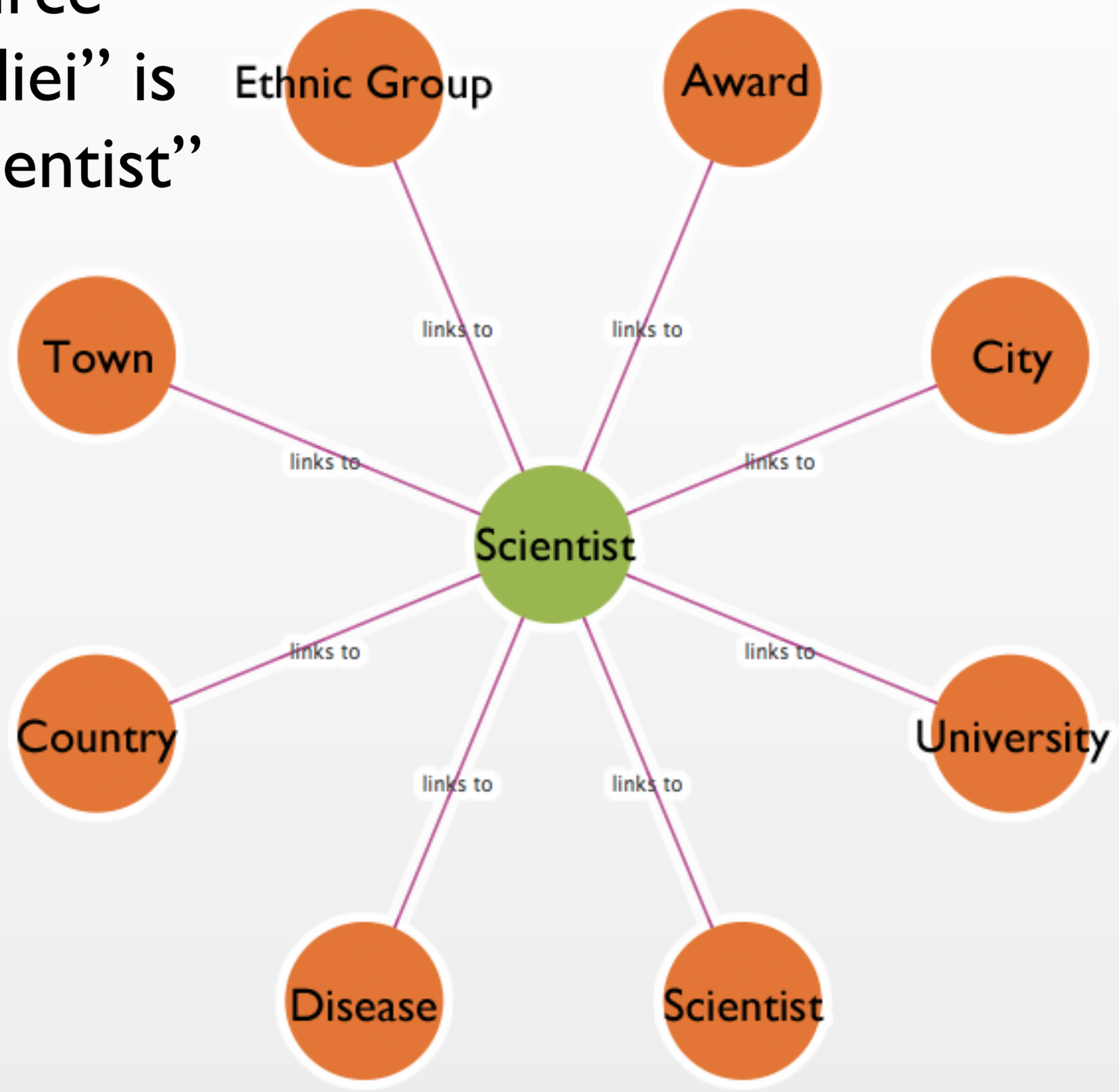
We have 231 EKPs

We compare the path types involving "Galileo Galilei" as subject with EKPs in order to identify the most similar, which is the "Scientist" EKP.

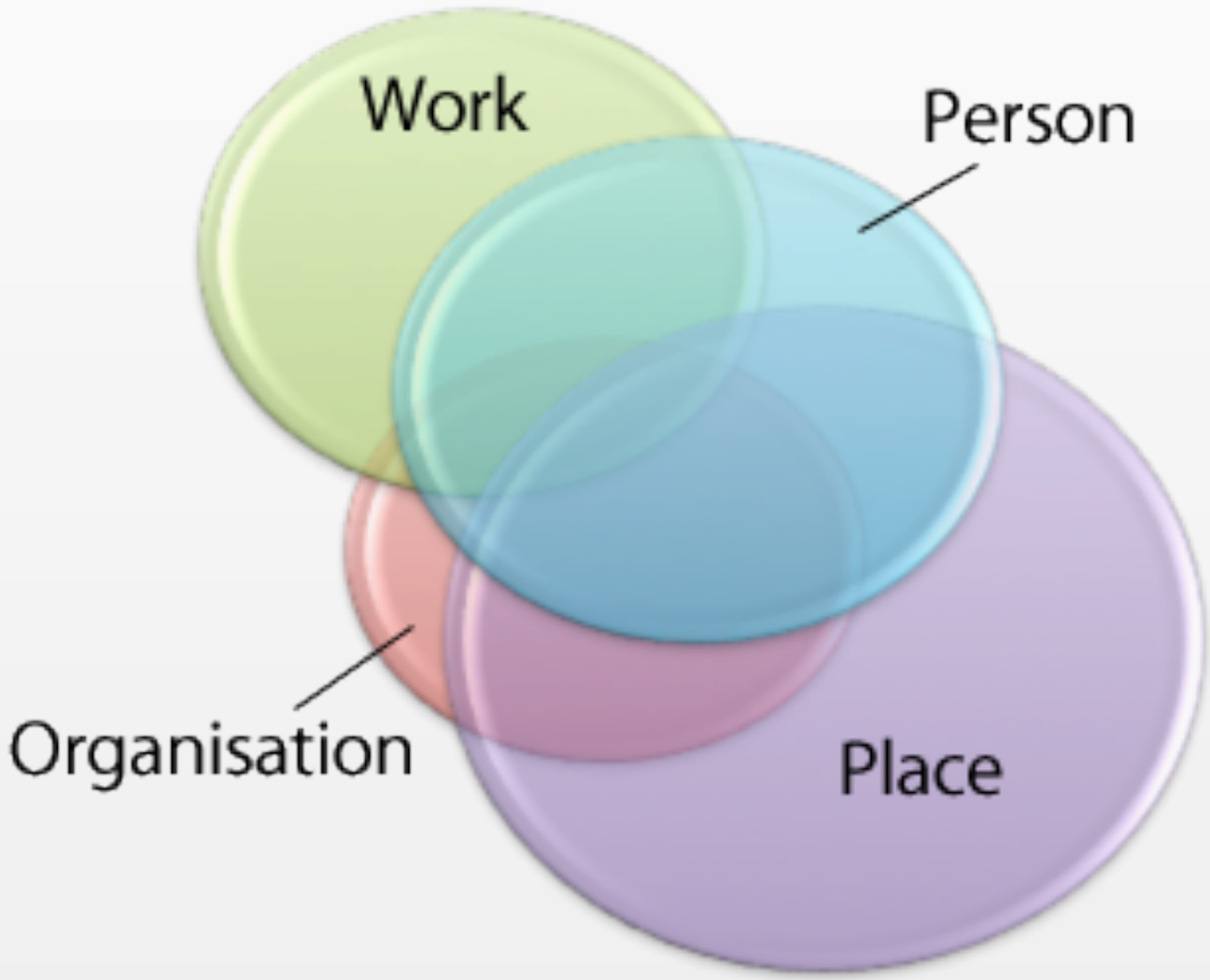




The inferred type for the resource "Galileo Galiei" is the class "Scientist"



Distinctive weakness of some EKPs



- The distinctive weakness seems due to wide overlaps among some EKPs
- Systematic ambiguity of the 4 largest classes

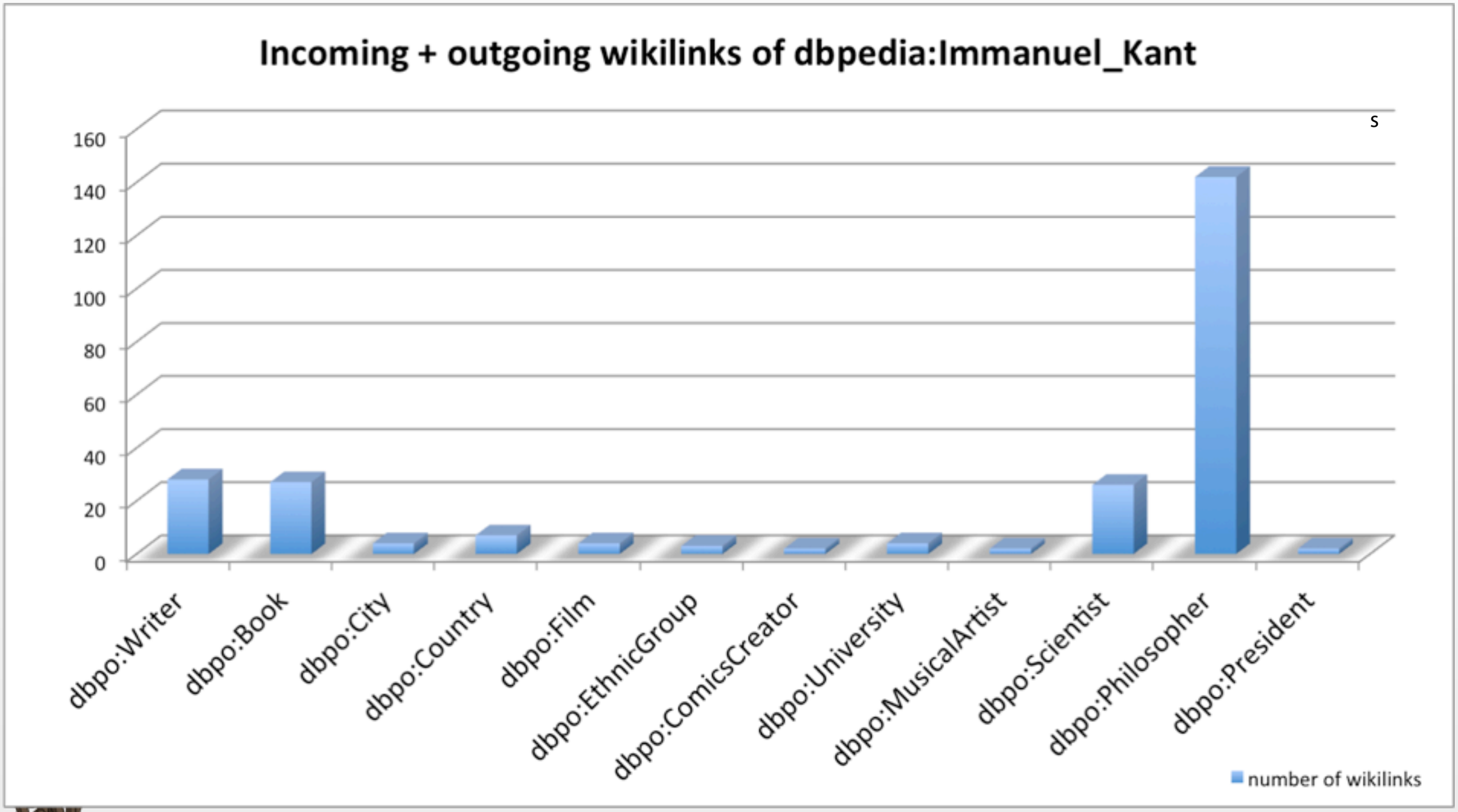
- Precision and recall on all DBPO types both 44.4%
- Precision and recall on the top-level of DBPO hierarchy: 36.5% and 79.5%

Homotype-based abductive classification

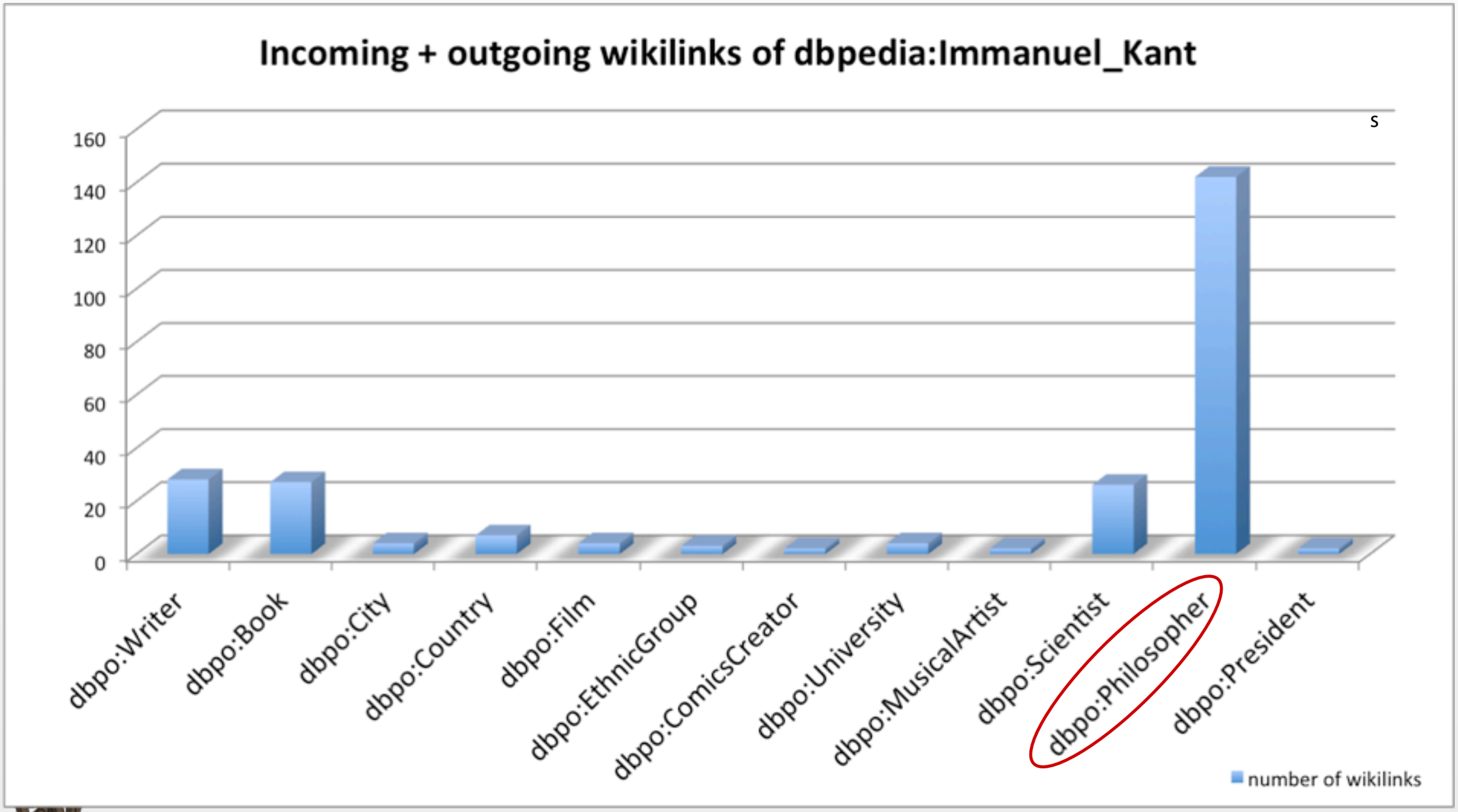
- **Homotypes are wikilinks that have the same type on both the subject and the object of the triple**

- **We have observed how the homotype is usually the most frequent (or in the top 3) wikilink type**
- **Given an untyped entity, we hypothesize that the most frequent type involved in its ingoing/ outgoing wikilinks detects its homotype, hence it indicates its type**

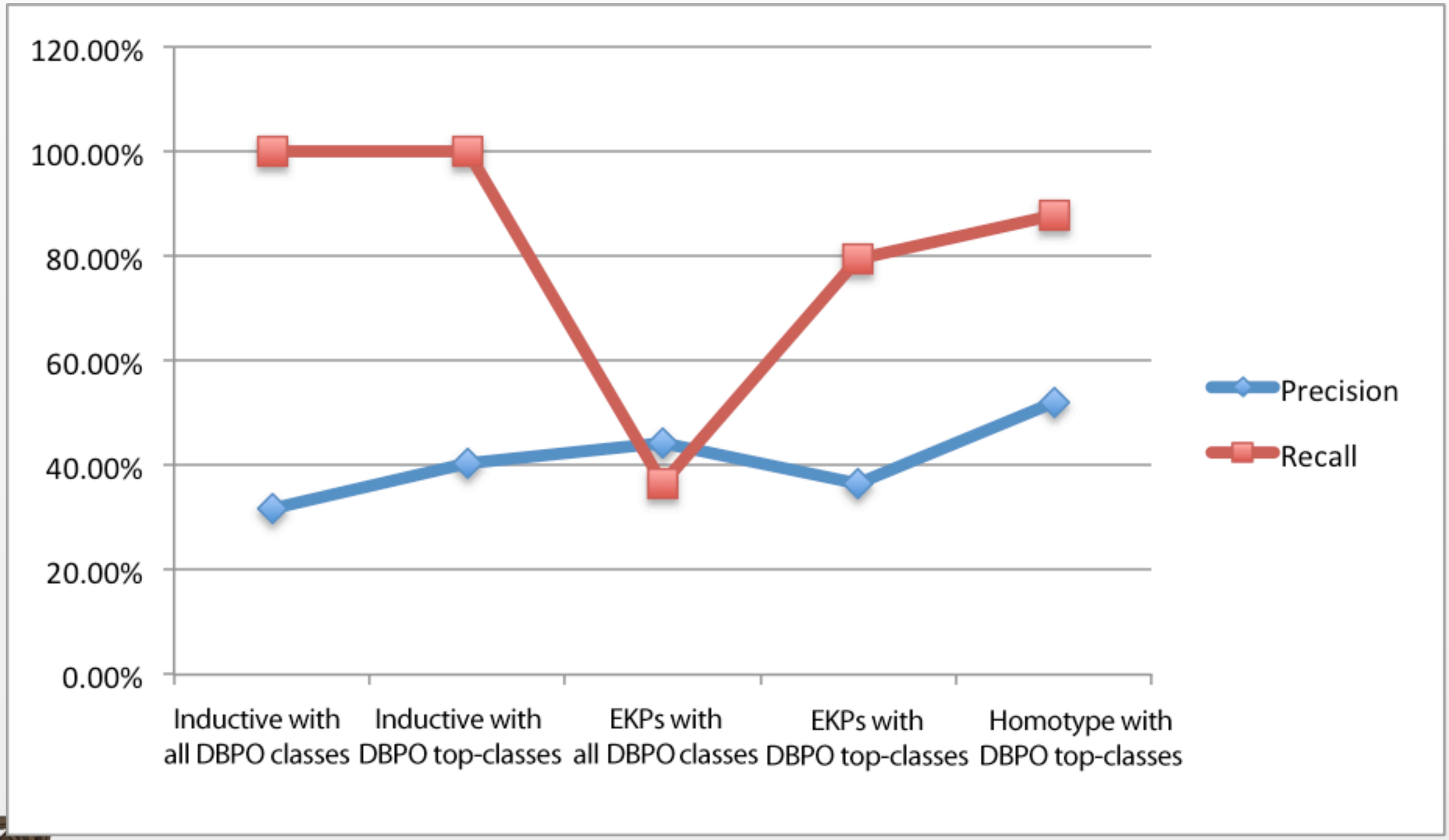
Homotype-based abductive classification



Homotype-based abductive classification



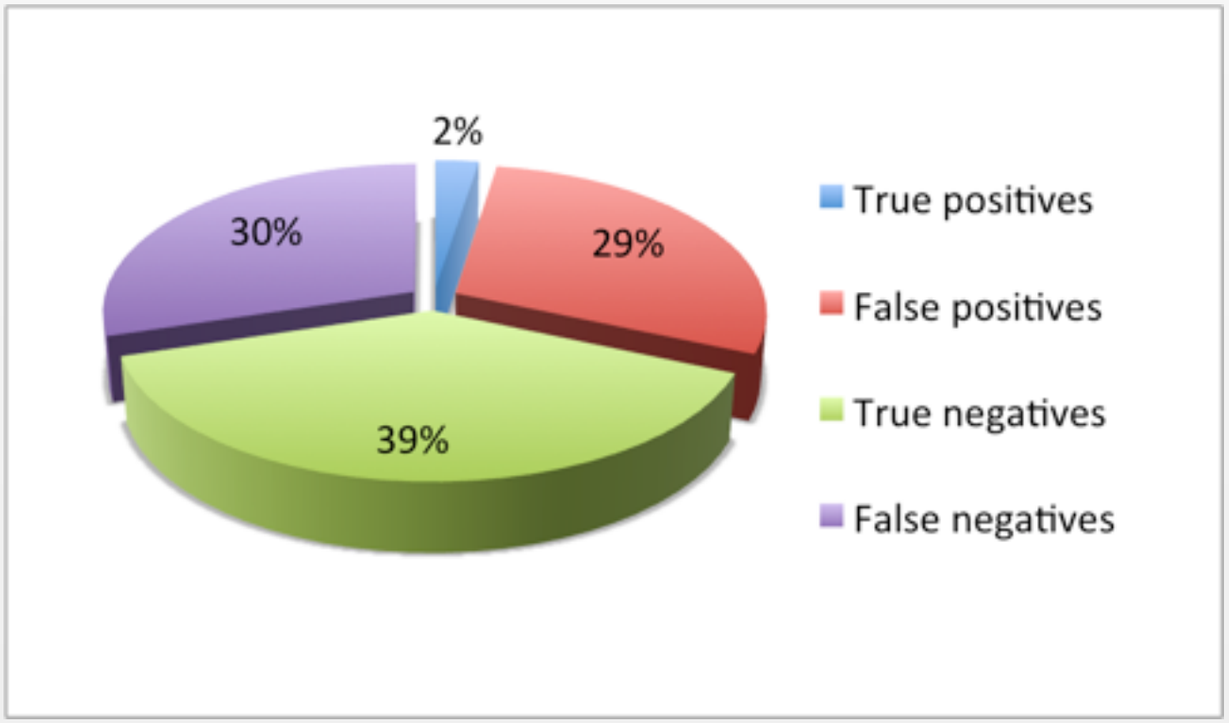
Results on classifying already typed resources



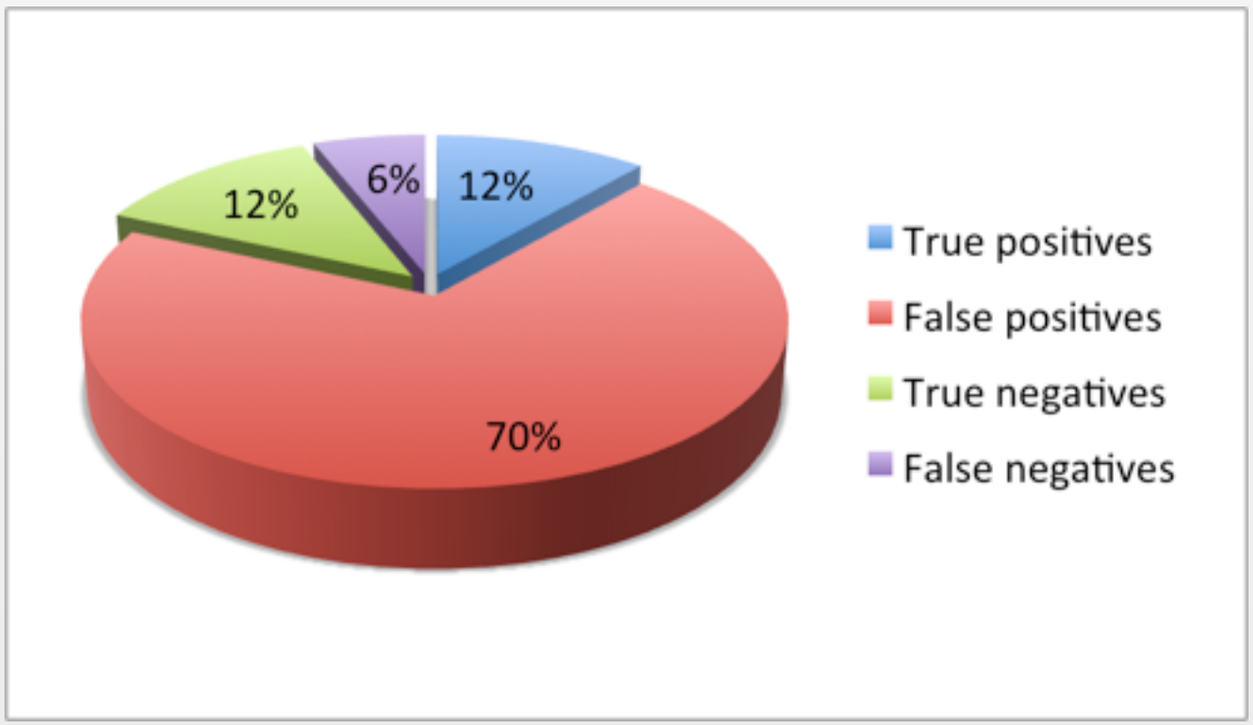
Results on untyped resources

- **Results on a sample of 1,000 untyped resources are much less satisfactory**

With EKPs



With Homotypes



Why? [1]

- **Typed entities: 2:3 typed wikilinks ratio**
- **Untyped entities: 1:3 typed wikilinks ratio**
- ***Link structure for untyped entities is not rich enough***

Why? [2]

- **DBPO does not provide a complete set of classes for correctly typing DBpedia resources**

dbpedia:List_of_FIFA_World_Cup_finals ⟶ Collection

dbpedia:Computer_Science ⟶ ScientificDiscipline

dbpedia:Counterattack ⟶ Plan

dbpedia:Eros(concept) ⟶ Concept

dbpedia:Gentlemen's_agreement ⟶ Agreement

Conclusions

- **We have investigated different approaches for typing DBpedia resources based on the data set of wikilinks**

- **Results are acceptable in the test set, but extensive untypedness in output links, and poor DBPO coverage severely compromise automatic typing for untyped resources**

- **We have analyzed possible causes deriving from some bias in DBpedia**

Future work

- **Yago could be helpful but**
  - there is a lack of mapping between YAGO and DBPO
  - it has larger coverage and only an overlap with DBPO
  - the granularity of its categories is finer, and not easily reusable, because the top level is very large

Thank you

Andrea Nuzzolese

-

STLab, ISTC-CNR

&

Dipartimento di Scienze dell'Informazione

University of Bologna

Italy