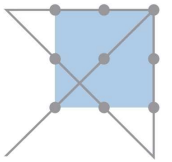
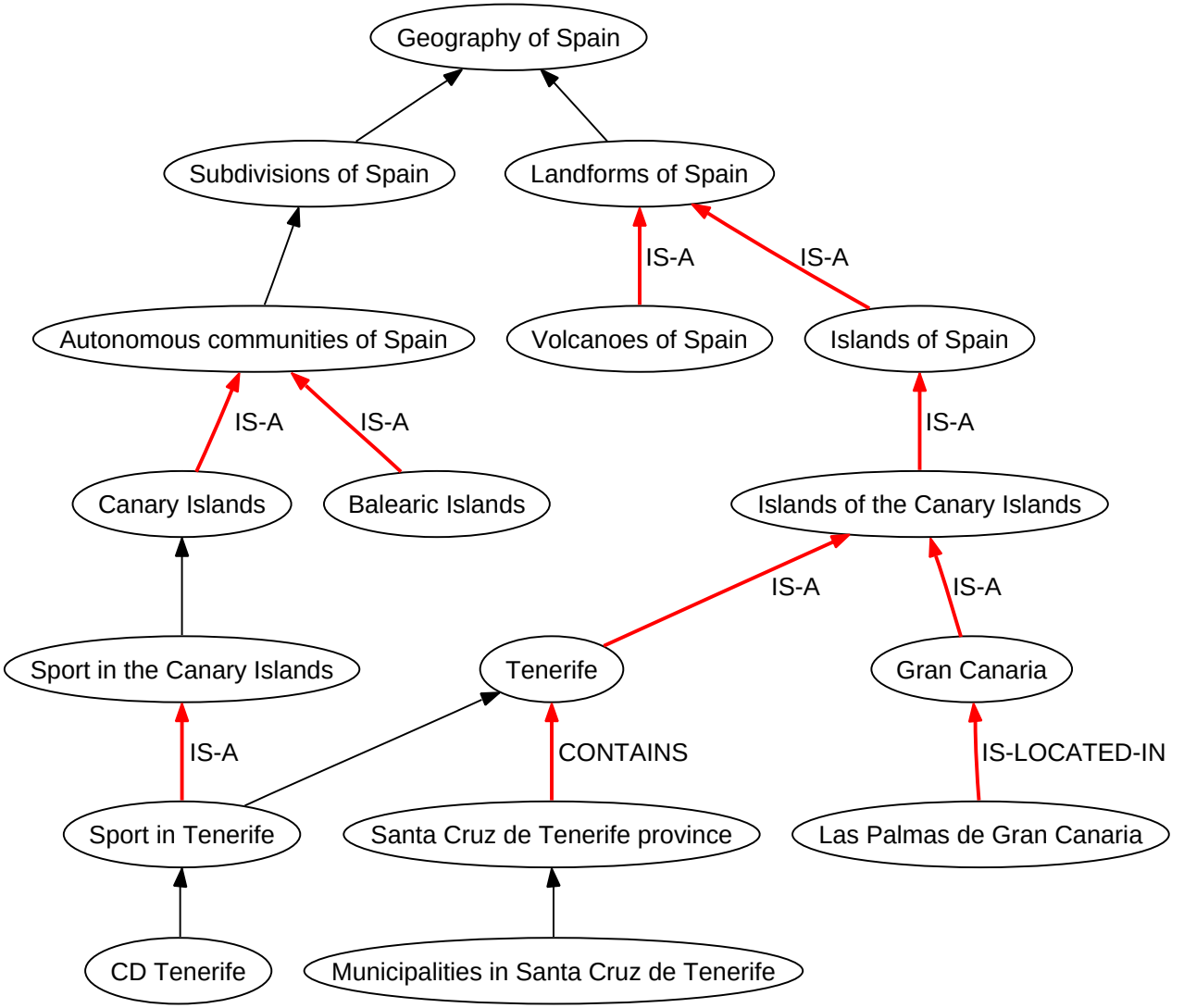


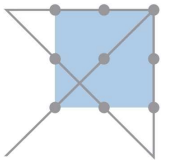
Distinguishing between Instances and Classes in the Wikipedia Taxonomy

Cäcilia Zirn

A Wikipedia Ontology?



Wikipedia Ontology



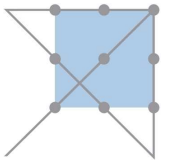
The big goal:

> Deriving an ontology from Wikipedia automatically

Necessary steps:

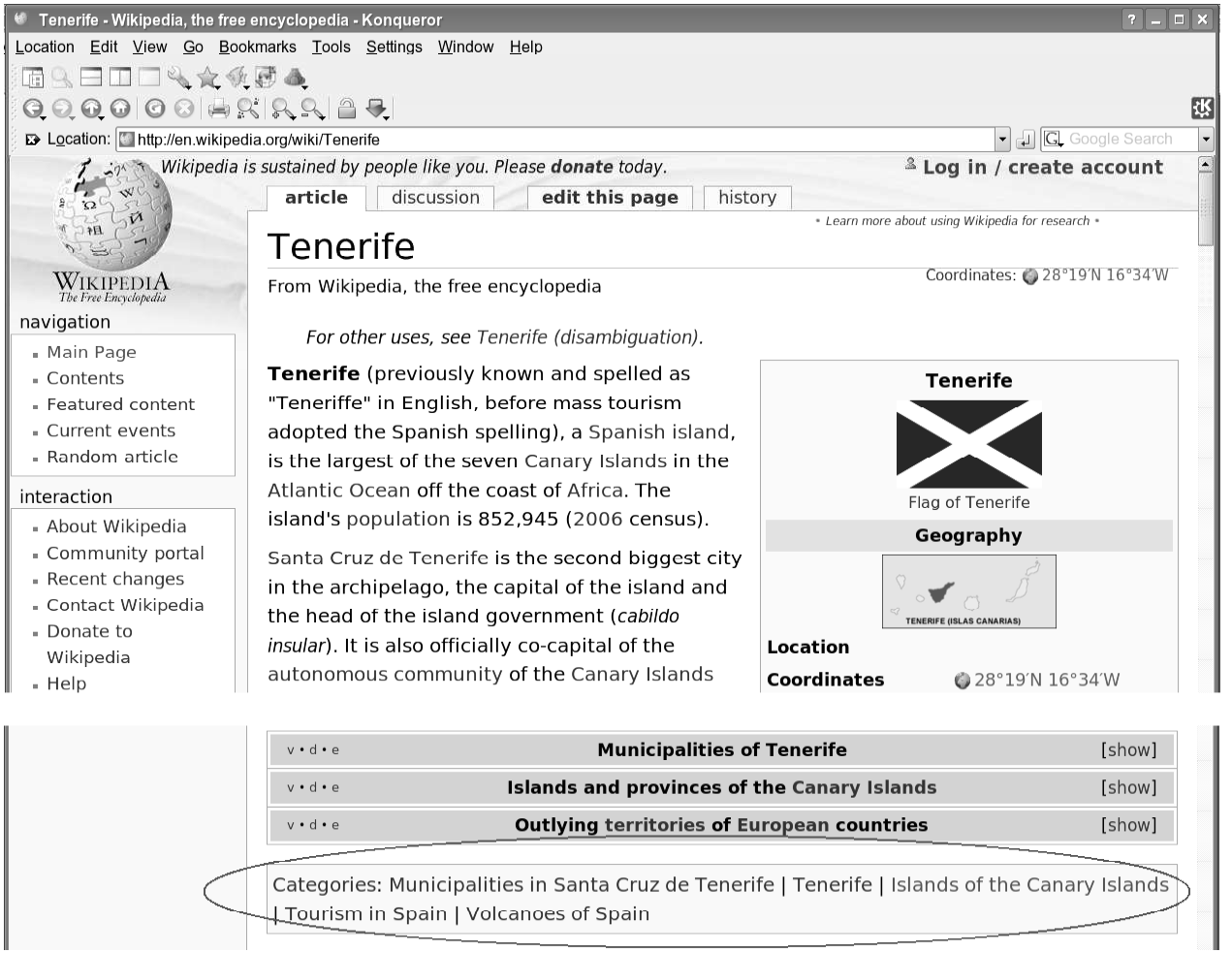
1. derive a **taxonomy** from Wikipedia (identify ISA relations), Ponzetto & Strube (AAAI 2007)
2. distinguish between **instances** and **classes** (work presented now)
3. interpret remaining **relations**, Nastase & Strube (AAAI 2008)

Outline

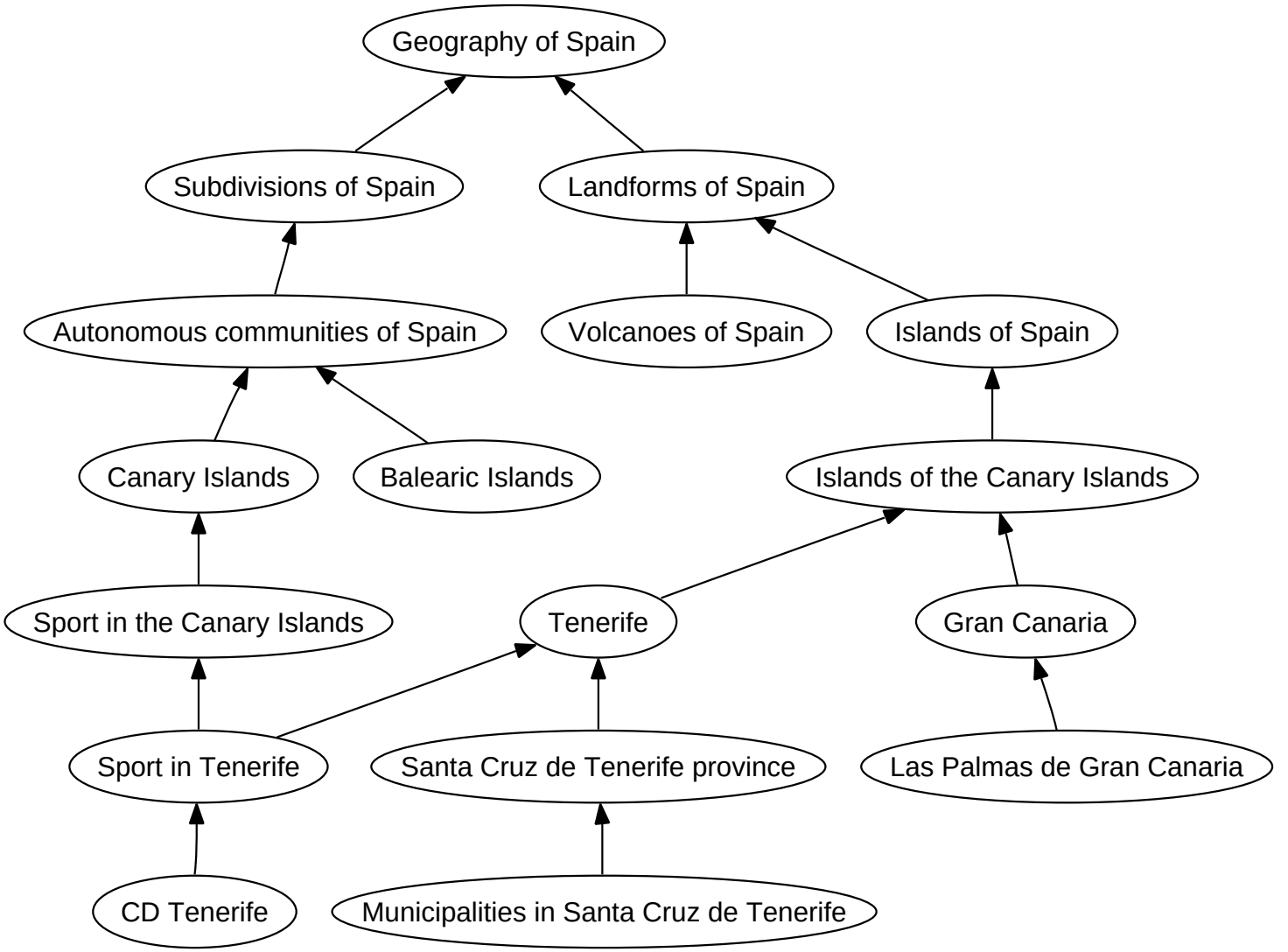


1. Deriving a taxonomy from Wikipedia
2. Instances and classes
3. Methods
4. Evaluation
5. Conclusions

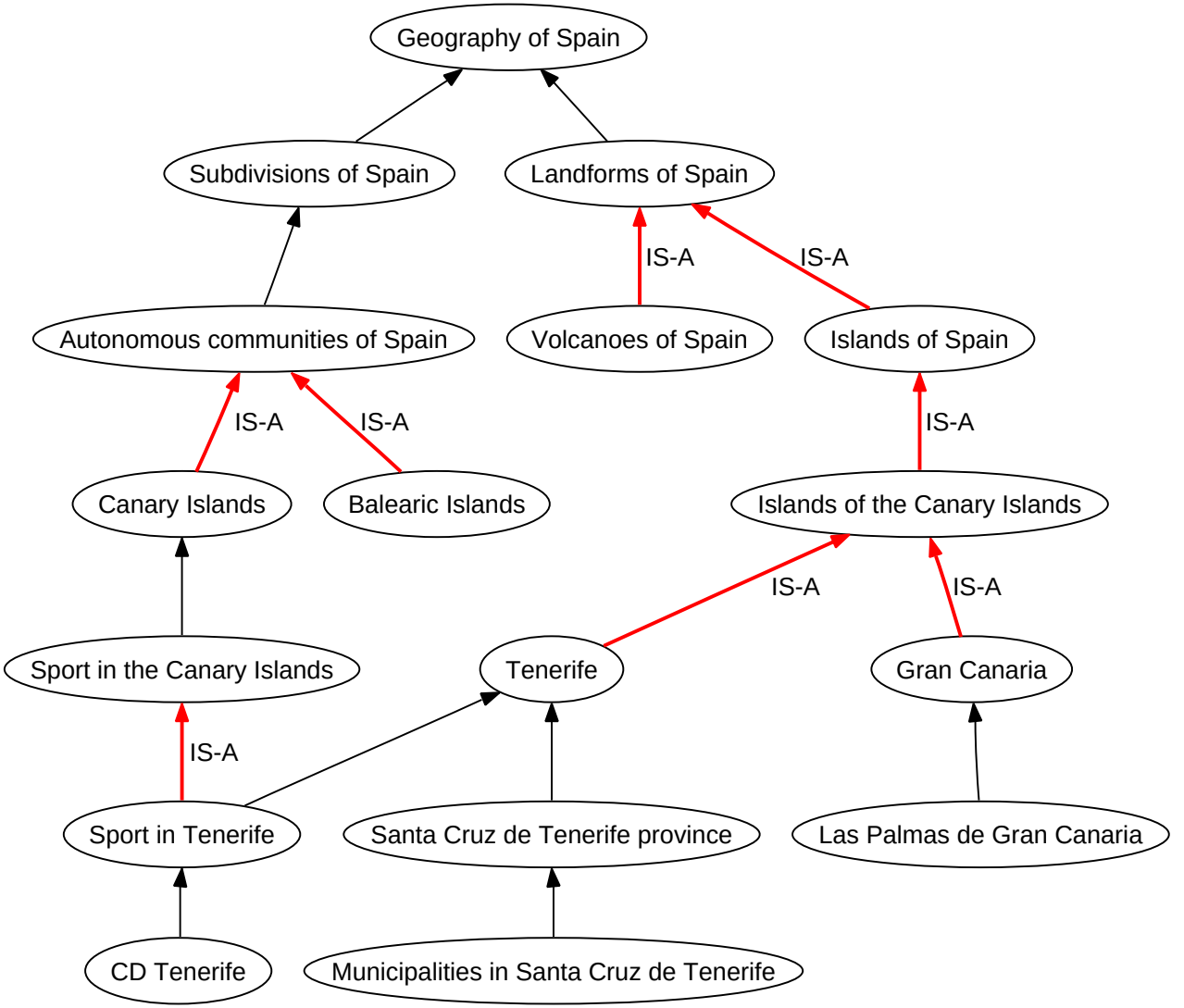
Prerequisites: Category Network



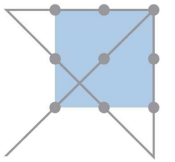
Deriving a taxonomy



Deriving a taxonomy



Outline



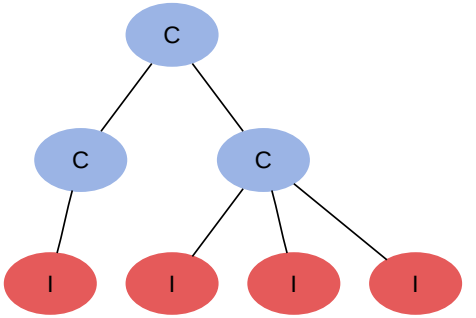
1. Deriving a taxonomy from Wikipedia
2. Instances and classes
3. Methods
4. Evaluation
5. Conclusions

Instances and Classes

Instances

TENERIFE, TEIDE, 2008

- are unique entities in the world
- in reasoning, they are mapped to objects

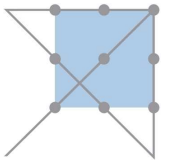


Classes

MUNICIPALITIES IN SANTA CRUZ DE TENERIFE, VOLCANOES OF SPAIN

- concepts that subsume classes or individuals
- in reasoning, they are mapped to predicates

Instances and Classes

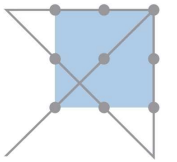


Distinction between instances and classes...

- can be found in WordNet and Cyc
- was done manually there
- agreement coefficient on this task
  on WordNet data κ = 0.75
  (Miller & Hristea, Computational Linguistics 2006)

➠ high cost!

Instances and Classes



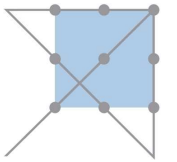
Distinction between instances and classes...

- can be found in WordNet and Cyc
- was done manually there
- agreement coefficient on this task
  on WordNet data κ = 0.75
  (Miller & Hristea, Computational Linguistics 2006)

➠ high cost!

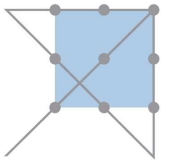
develop heuristics to distinguish between
instances and classes **fully automatically**

Instances and Classes



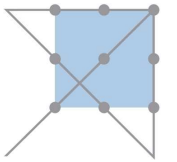
- intuitively all Wikipedia categories seem to be classes
- not the case
- instances also used as categories to organize related concepts

Outline



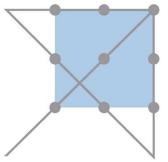
1. Deriving a taxonomy from Wikipedia
2. Instances and classes
3. Methods
4. Evaluation
5. Conclusions

Methods



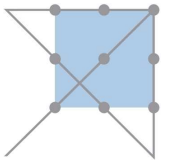
- development of 5 methods
  - Structure-based method
  - NER (Named entity recognition)
  - Capitalization
  - Plural
  - Page
- all are **heuristics**
- use NLP techniques
- based on category network

Methods



- development of 5 methods
  - Structure-based method
  - NER (Named entity recognition)
  - Capitalization
  - Plural
  - Page

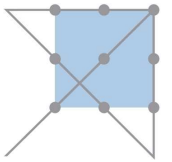
Structure-based method (1)



Only classes can have instances and classes.

TENERIFE, TENERIFE NORTH AIRPORT

Structure-based method (1)



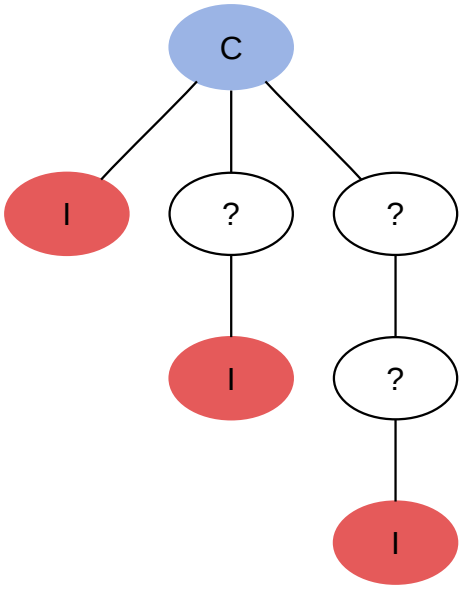
Only classes can have instances and classes.

TENERIFE, TENERIFE NORTH AIRPORT

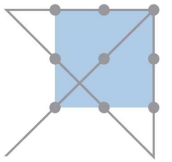
- if a category has hyponyms, it has to be a class
- count hyponyms (incoming ISA-links)

Structure-based method (2)

- if a category has **more than one** hyponym:
  - the category is labeled as **Class**
- if a category has **no** hyponym:
  - the category is labeled as **Instance**



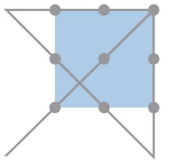
Structure-based method (3)



Only classes can have instances and classes.

TENERIFE, TENERIFE NORTH AIRPORT

Structure-based method (3)



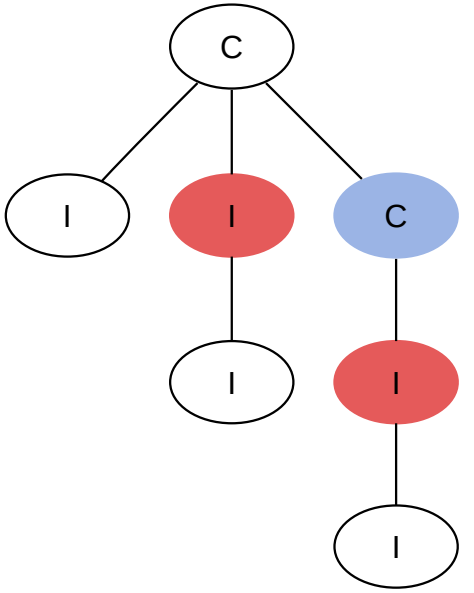
Only classes can have instances and classes.

TENERIFE, TENERIFE NORTH AIRPORT

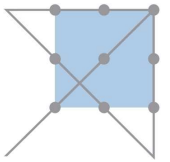
- labeling of the ISA-links has been done automatically
- possible that links are classified erroneously
- ➠ tolerate one erroneous link

Structure-based method (4)

- if a category has **exactly one** hyponym:
  - if the hyponym **has a hyponym** itself:
    - the category is labeled as **Class**
  - if the hyponym **has no hyponym**:
    - the category is labeled as **Instance**

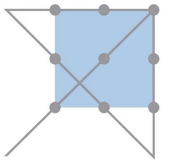


Methods



- development of 5 methods
  - Structure-based method
  - NER (Named entity recognition)
  - Capitalization
  - Plural
  - Page

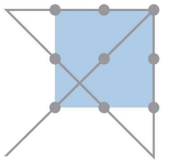
Method: NER (1)



Instances correspond to unique entities in the world and are therefore named entities.

TENERIFE, CD TENERIFE

Method: NER (1)

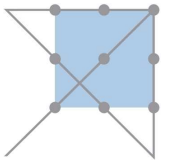


Instances correspond to unique entities in the world and are therefore named entities.

TENERIFE, CD TENERIFE

- idea: use a named entity recognizer
- we use CRFClassifier (Stanford)
  - *Person, Location, Organization* for named entities
  - *Other* for the rest

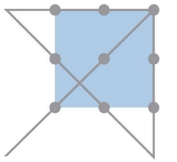
Method: NER (2)



Instances correspond to unique entities in the world and are therefore named entities.

TENERIFE, CD TENERIFE

Method: NER (2)

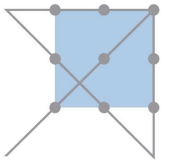


Instances correspond to unique entities in the world and are therefore named entities.

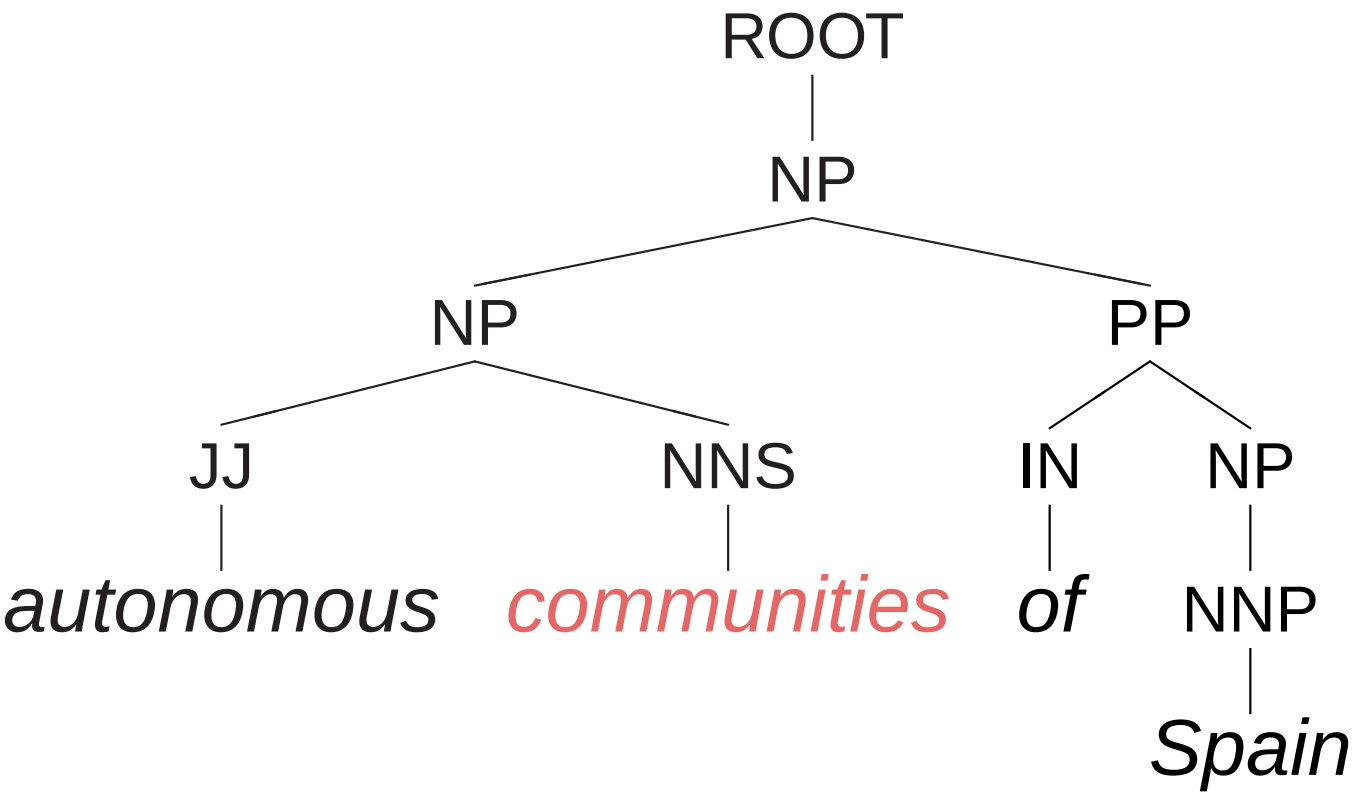
TENERIFE, CD TENERIFE

- some names consist of complex noun structures: AUTONOMOUS COMMUNITIES OF SPAIN
  - only lexical heads are passed to named entity recognizer
  - lexical heads are extracted using Stanford Parser

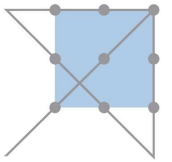
Utility: Lexical head finder



- lexical heads: determine the syntactic properties of a phrase
- in a noun phrase: the noun

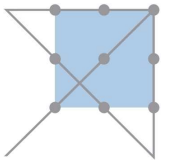


Method: NER (3)



- if the named entity recognizer returns one of the labels: *Person, Location, Organization*:
  - the category is labeled as **Instance**
- if the named entity recognizer returns the label *Other*:
  - the category is labeled as **Class**

Method: NER (3)

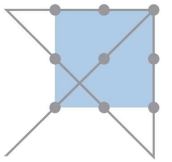


- if the named entity recognizer returns one of the labels: *Person, Location, Organization*:
  - the category is labeled as **Instance**
- if the named entity recognizer returns the label *Other*:
  - the category is labeled as **Class**

the parser sometimes returns several heads

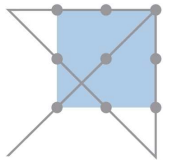
- if the majority of returned labels is *Other*:
  - the category is labeled as **Class**
- otherwise:
  - the category is labeled as **Instance**

Methods



- development of 5 methods
  - Structure-based method
  - NER (Named entity recognition)
  - Capitalization
  - Plural
  - Page

Method: Capitalization (1)



Content words belonging to a named entity are capitalized.

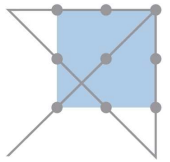
Convention for Wikipedia titles.

TENERIFE LADIES OPEN

and

AUTONOMOUS COMMUNITIES OF SPAIN

Method: Capitalization (1)



Content words belonging to a named entity are capitalized.

Convention for Wikipedia titles.

TENERIFE LADIES OPEN

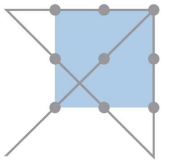
and

AUTONOMOUS COMMUNITIES OF SPAIN

- Bunescu & Paşca (2006) developed a heuristic to process Wikipedia **page** titles:
  "If all content words of a page title are capitalized, it corresponds to a named entity"

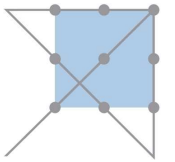
➠ we apply this heuristic to **category** titles

Method: Capitalization (2)



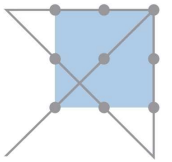
1. preprocess first word
   - first word is always capitalized
   - ➠ pass it to CRFClassifier
   - if it is not recognized as a named entity: lowercase the word
2. filter out function words
3. analyze remaining words
   - if all words are capitalized:
     - ➠ the category is labeled as **Instance**
   - otherwise:
     - ➠ the category is labeled as **Class**

Methods



- development of 5 methods
  - Structure-based method
  - NER (Named entity recognition)
  - Capitalization
  - Plural
  - Page

Method: Plural (1)



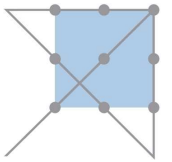
Instances are unique ➠ generally used in singular form.

Tenerife, Spain

and

Autonomous communities of Spain

Method: Plural (1)



Instances are unique ➠ generally used in singular form.

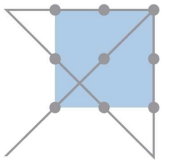
TENERIFE, SPAIN

and

AUTONOMOUS COMMUNITIES OF SPAIN

- exceptions: *“The Millers are coming to our party”* not to be expected in Wikipedia category titles

Method: Plural (2)



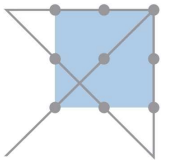
Instances are unique ➠ generally used in singular form.

Tenerife, Spain

and

Autonomous communities of Spain

Method: Plural (2)



Instances are unique ➠ generally used in singular form.

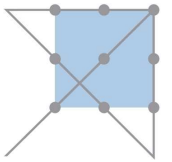
TENERIFE, SPAIN

and

AUTONOMOUS COMMUNITIES OF SPAIN

- the grammatical number of the lexical head is the same as the number of the category title
- ➠ we parse the category title with the Stanford Parser, obtaining:
  - the lexical head(s)
  - the part-of-speech tags

Method: Plural (3)



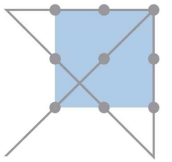
Instances are unique ➠ generally used in singular form.

TENERIFE, SPAIN

and

AUTONOMOUS COMMUNITIES OF SPAIN

Method: Plural (3)



Instances are unique ➠ generally used in singular form.

TENERIFE, SPAIN

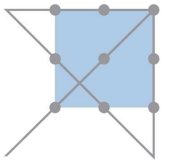
and

AUTONOMOUS COMMUNITIES OF SPAIN

- tags of interest:
  - NNPS = noun, proper, plural
  - NNS = noun, common, plural
  - NNP = noun, proper, singular

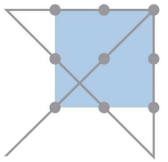
Autonomous/JJ communities/NNS of/IN Spain/NNP

Method: Plural (3)



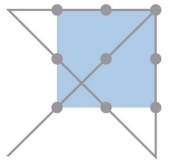
- if the lexical head of a phrase is tagged as plural noun (NNS, NNPS):
  - the category is labeled as **Class**
- otherwise:
  - the category is labeled as **Instance**

Methods



- development of 5 methods
  - Structure-based method
  - NER (Named entity recognition)
  - Capitalization
  - Plural
  - Page

Method: Page (1)

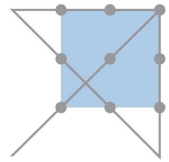


Articles should be placed in categories
with the same name.

Advice for authors in Wikipedia.

SPAIN, TENERIFE

Method: Page (1)

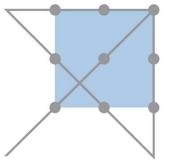


> Articles should be placed in categories with the same name.
>
> *Advice for authors in Wikipedia.*

SPAIN, TENERIFE

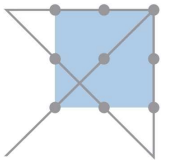
- a number of articles have homonymous categories
- most articles refer to unique entities
- heuristic: a category containing a page with the same name is an instance

Method: Page (2)



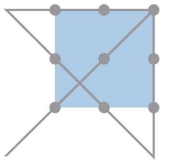
- if a page with homonymous title exists:
  - the category is labeled as **Instance**
- otherwise:
  - the category is labeled as **Class**

Implementation



- working on Wikipedia derived taxonomy
- containing 127,124 nodes and 106,258 (ISA) links
- data accessed via Wikipedia dump
- own Wikipedia-API (in Java)

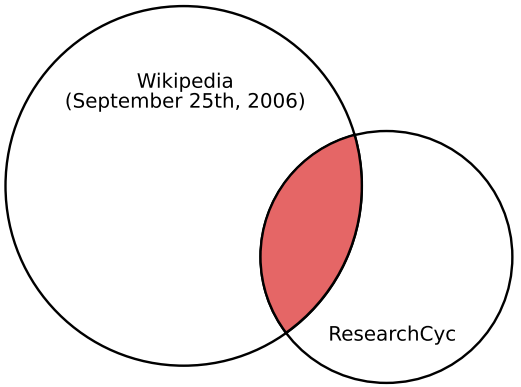
Outline



1. Deriving a taxonomy from Wikipedia
2. Instances and classes
3. Methods
4. Evaluation
5. Conclusions

Data (1)

Use ResearchCyc as **gold standard**.



ResearchCyc

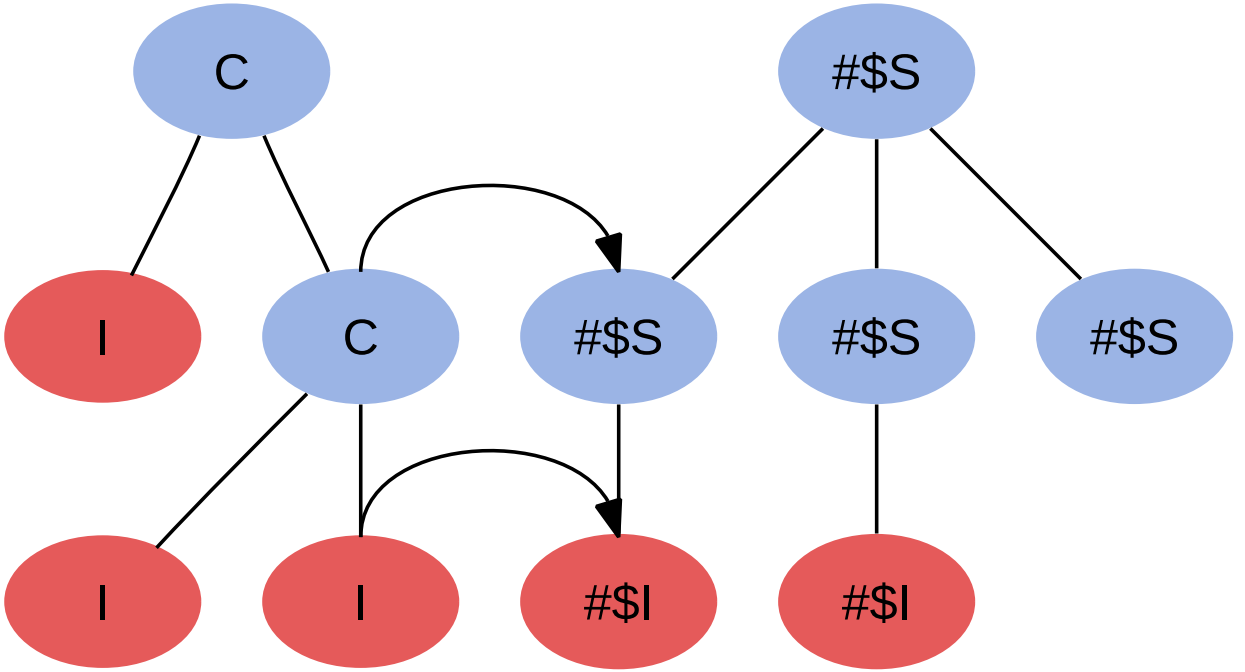
- distinguishes between #$Individual and #$SetOrCollection
- distinction is done manually

overlap Wikipedia / ResearchCyc:

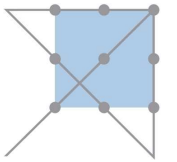
- 7860 concepts
  - 44.35%(3486)#$Individual
  - 55.65%(4374)#$SetOrCollection

Data (2)

Use ResearchCyc as **gold standard**.



Measures (1)



$T_{instances}$	$F_{classes}$
$F_{instances}$	$T_{classes}$

$$\text{Prec}_{instances} = \frac{T_{instances}}{T_{instances} + F_{instances}}$$

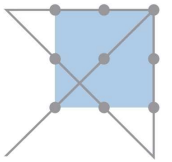
$T_{instances}$: Instance in Wiki & Individual in Cyc

$F_{instances}$: Instance in Wiki but **not** Individual in Cyc

$T_{classes}$: Class in Wiki & SetOrCollection in Cyc

$F_{classes}$: Class in Wiki but **not** SetOrCollection in Cyc

Measures (2)



$T_{instances}$	$F_{classes}$
$F_{instances}$	$T_{classes}$

$$\text{Prec}_{classes} = \frac{T_{classes}}{T_{classes}+F_{classes}}$$

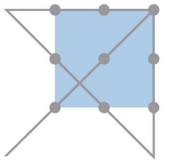
$T_{instances}$: Instance in Wiki & Individual in Cyc

$F_{instances}$: Instance in Wiki but **not** Individual in Cyc

$T_{classes}$: Class in Wiki & SetOrCollection in Cyc

$F_{classes}$: Class in Wiki but **not** SetOrCollection in Cyc

Measures (3)



$T_{instances}$	$F_{classes}$
$F_{instances}$	$T_{classes}$

$$\text{Accuracy} = \frac{T_{instances}+T_{classes}}{T_{instances}+F_{instances}+T_{classes}+F_{classes}}$$

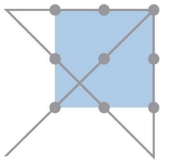
$T_{instances}$: Instance in Wiki & Individual in Cyc

$F_{instances}$: Instance in Wiki but **not** Individual in Cyc

$T_{classes}$: Class in Wiki & SetOrCollection in Cyc

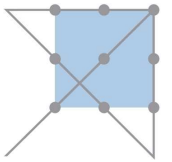
$F_{classes}$: Class in Wiki but **not** SetOrCollection in Cyc

Evaluate every method separately



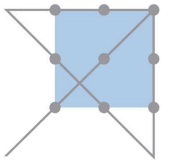
Method	$\text{Prec}_{instances}$	$\text{Prec}_{classes}$
NER	**85.23**	76.84
page	66.1	**91.5**
capitalization	**85.99**	82.44
plural	66.44	**87.99**
structure	56.17	**87.21**

Evaluate every method separately



Method	$\text{Prec}_{instances}$	$\text{Prec}_{classes}$	Accuracy
NER	**85.23**	76.84	79.69
page	66.1	**91.5**	75.74
capitalization	**85.99**	82.44	83.82
plural	66.44	**87.99**	75.24
structure	56.17	**87.21**	64.71

Final setting

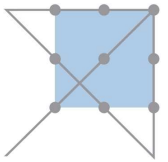


Classification schemes

A) Accuracy scheme

- method with best accuracy: **capitalization**
- (regard scheme as baseline)

Final setting



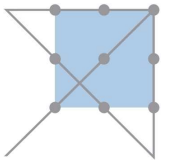
Classification schemes

B) Precision scheme

- order methods according to their precision ($\text{Prec}_{instances}$ or $\text{Prec}_{classes}$)

1. page ➪ class
2. plural ➪ class
3. structure ➪ class
4. capitalization ➪ instance
5. remaining categories ➪ class

Final setting

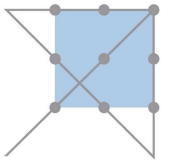


Classification schemes

C) Voting scheme

1. page & plural ➨ class
2. capitalization & NER ➨ instance
3. remaining categories ➨ precision scheme

Final setting



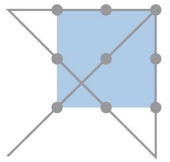
Classification schemes

A) Accuracy scheme
B) Precision scheme
C) Voting scheme

special form of cross-validation:

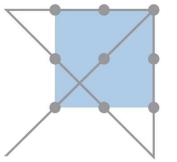
- 5 rounds of binary random splits
- maintain the #$Individual / #$SetOrCollection distribution
- evaluate on the resulting 10 data sets

Final results



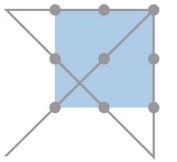
Method	$\text{Precision}_{instances}$	$\text{Precision}_{classes}$	Accuracy
A) *Accuracy sc.*	85.99±0.54	**82.44±0.63**	82.82±0.5
B) *Precision sc.*	**90.92±0.41**	77.36±0.52	81.64±0.42
C) *Voting sc.*	89.21±0.46	81.82±0.52	**84.52±0.34**

Discussion



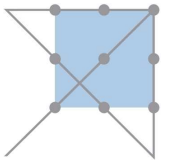
- preprocessing errors, e.g. wrong parsing results
  (…AND YOU WILL KNOW US BY THE TRAIL OF DEAD ALBUMS)
- recognizing named entities:
  BEE TRAIN
  If components of a named entity are not named entities, it is not recognized
- classification in gold standard:
  Inter-agreement between judges is not 100%
  - ➠ different possible judgements

Outline



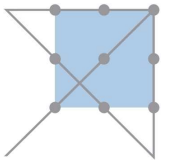
1. Deriving a taxonomy from Wikipedia
2. Instances and classes
3. Methods
4. Evaluation
5. Conclusions

Conclusions



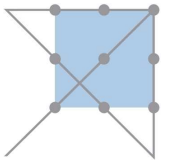
- automatic distinction between instances and classes is possible with a high accuracy (84.52%)
- combining the methods with machine learning could improve performance even more
- next step: introducing distinction between instances and classes to Wikipedia articles
- methods can easily be applied to other languages

The resulting resource



- we applied classification scheme C
  to the 127,124 nodes in the Wikipedia taxonomy
  - **15,472** nodes were classified as **instance**
  - **111,652** nodes were classified as **class**
- we converted the data into **RDF Schema** file
  (using the Jena Semantic Web Framework)
- the result is freely available

Thanks!



Acknowledgements

- Michael Strube and Vivi Nastase
- Anette Frank for supervising my bachelor thesis
- Simone Ponzetto for his work in deriving the taxonomy
- Klaus Tschira Foundation

Check out

... the results (RDF Schema)

`www.eml-research.de/nlp/download/wikitaxonomy.php`

... more papers on Wikipedia

`www.eml-research.de/~strube`