An Ontology-driven Approach for Semantic Annotation of Documents with Specific Concepts

Céline Alec   Chantal Reynaud-Delaître   Brigitte Safar

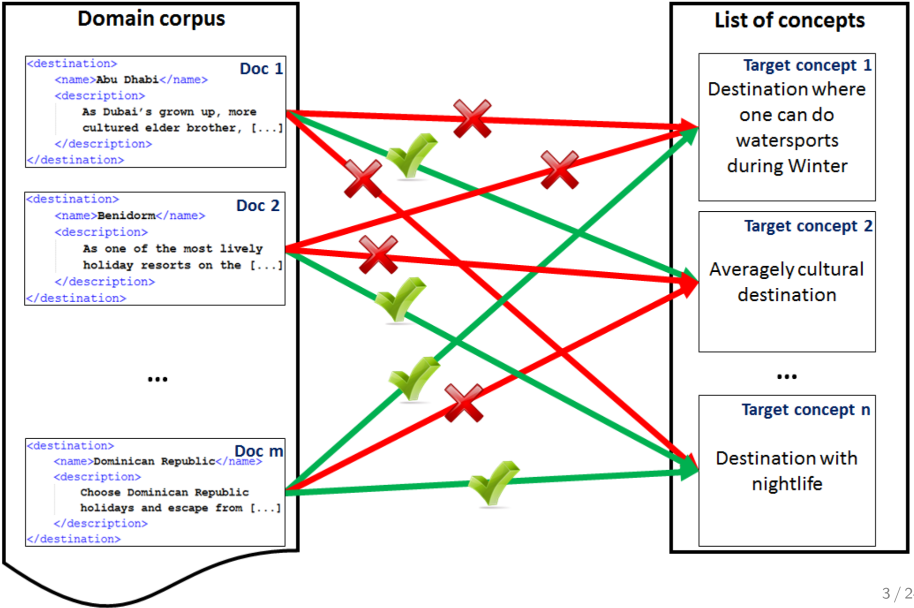
LRI, Univ. Paris-Sud, CNRS, Université Paris-Saclay, Orsay, France
{celine.alec,chantal.reynaud,brigitte.safar}@lri.fr

May 31, 2016

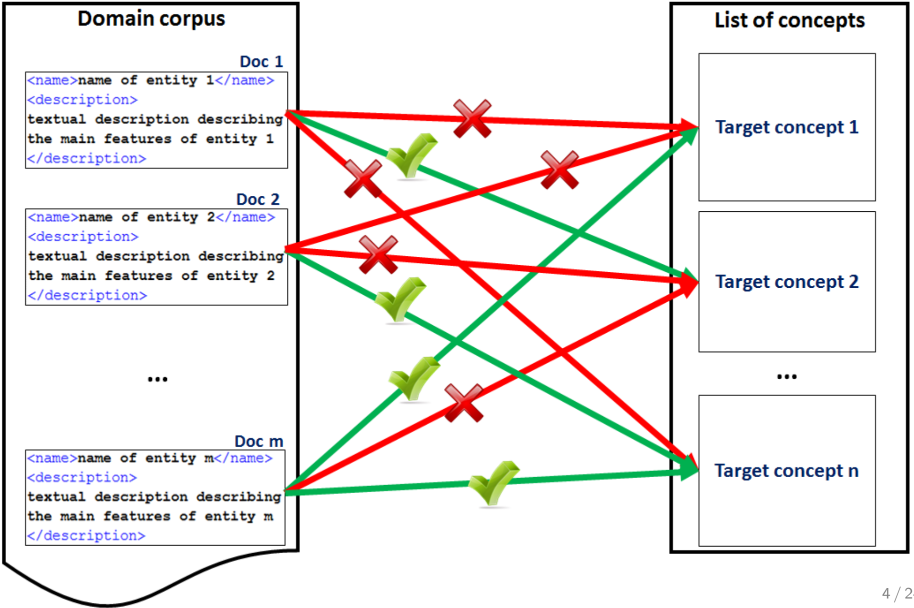
ESWC

Outline

Automatic semantic annotation of documents



Generic method: has to work for different domains



Target concepts = specific concepts

Only names of concepts

1. they are not explicitly mentioned in the documents

Example

"Destination where one can do watersports during Winter"

1. Not said in the document because user point of view

Target concepts = specific concepts

Only names of concepts

1. they are not explicitly mentioned in the documents
2. they are not defined, even if a domain expert knows their meaning
   $\Rightarrow$ need to learn the definitions

Example

"Destination where one can do watersports during Winter"

1. Not said in the document because user point of view
2. Watersports feasable in winter
   Weather good enough in winter?

Target concepts = specific concepts

Only names of concepts

1. they are not explicitly mentioned in the documents
2. they are not defined, even if a domain expert knows their meaning
   $\Rightarrow$ need to learn the definitions
3. data from the documents insufficient to automatically annotate
   $\Rightarrow$ need to extract data from both documents and external resources

Example

"Destination where one can do watersports during Winter"

1. Not said in the document because user point of view
2. Watersports feasable in winter
   Weather good enough in winter?
3. Watersports OK
   Weather information KO

Related work: no solution in the state of the art

Two close works [Petasis et al., 2013, Yelagina and Panteleyev, 2014]: aim to deduce facts not explicitly present in the texts

- both use ontologies
- two-time processes:
  1. extraction of information from the documents
  2. reasoning: deduction of new facts from step 1 and given definitions

Our work

- uses an ontology (central role)
- same two-time process but two more problems:
  - all the necessary information to make the annotations is not mentioned
  - definitions are not given

Related work: no solution in the state of the art

Two close works [Petasis et al., 2013, Yelagina and Panteleyev, 2014]: aim to deduce facts not explicitly present in the texts

- both use ontologies
- two-time processes:
  1. extraction of information from the documents
  2. reasoning: deduction of new facts from step 1 and given definitions

Our work

- uses an ontology (central role)
- same two-time process but two more problems:
  - all the necessary information to make the annotations is not mentioned
    $\Rightarrow$ need to use external resources (Linked Open Data)
  - definitions are not given
    $\Rightarrow$ need to learn the definitions (machine learning)

Our approach: 4 inputs (provided)

1. corpus of XML documents (little structure)
   - description: hardly any negative expressions

```
<name>name of the entity</name>
<description>
textual description describing
the main features of the entity
</description>
```

2. list of target concepts

Our approach: 4 inputs (provided)

1. corpus of XML documents (little structure)
   - description: hardly any negative expressions
   - use of machine learning: some documents have to be manually annotated for each target concept (positive/negative examples)
2. list of target concepts

```
<name>name of the entity</name>
<description>
textual description describing
the main features of the entity
</description>
```

Our approach: 4 inputs (provided)

1. corpus of XML documents (little structure)
   - description: hardly any negative expressions
   - use of machine learning: some documents have to be manually annotated for each target concept (positive/negative examples)
2. list of target concepts
3. domain ontology

```
<name>name of the entity</name>
<description>
textual description describing
the main features of the entity
</description>
```

Our approach: 4 inputs (provided)

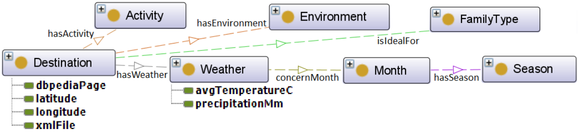
1. corpus of XML documents (little structure)
   - description: hardly any negative expressions
   - use of machine learning: some documents have to be manually annotated for each target concept (positive/negative examples)

```
<name>name of the entity</name>
<description>
textual description describing
the main features of the entity
</description>
```

2. list of target concepts
3. domain ontology
4. correspondences between properties: ontology $\leftrightarrow$ external resources (LOD)

Our approach: 4 inputs (provided)

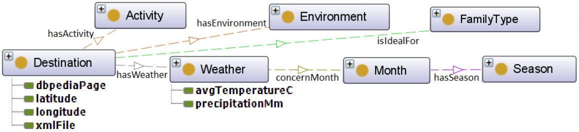
3. domain ontology (OWL)
   - classes

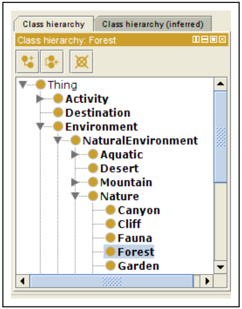


   - properties
   - individuals
   - axioms

Our approach: 4 inputs (provided)

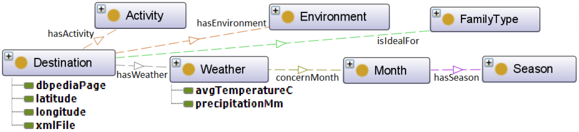
3. domain ontology (OWL)
   - classes
     - *1 main class* (e.g., Destination)
     - *descriptive classes* (e.g., Activity, etc.)
   - properties
   - individuals
   - axioms

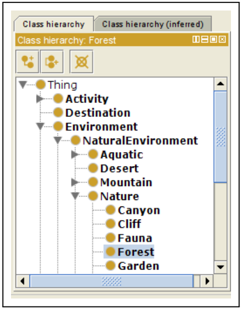




Our approach: 4 inputs (provided)

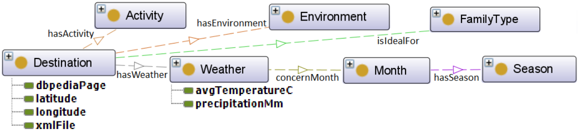
3. domain ontology (OWL)
   - classes
   - properties (object, datatype, annotation)
   - individuals
   - axioms

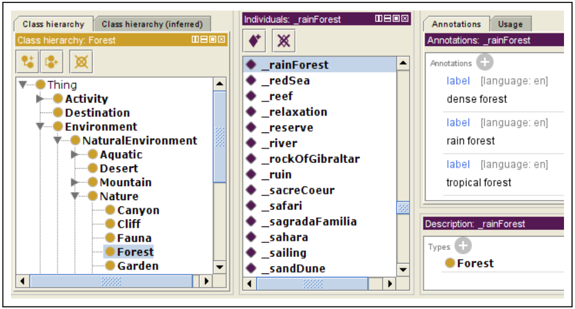




Our approach: 4 inputs (provided)

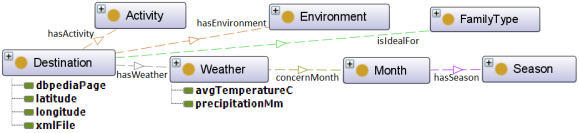
3. domain ontology (OWL)
   - classes
   - properties
   - individuals: instances of some descriptive classes ⇒ have terminology
   - axioms



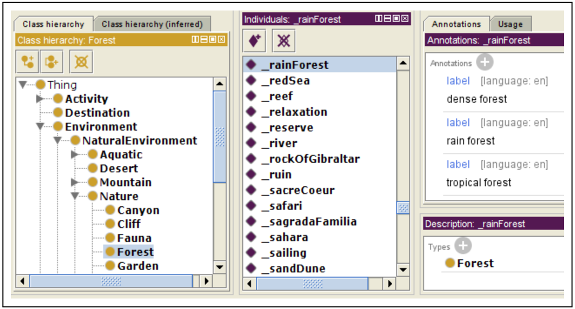


Our approach: 4 inputs (provided)

3. domain ontology (OWL)
   - classes



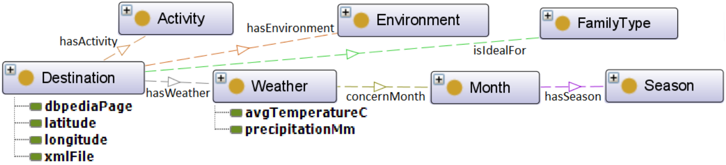
   - properties
   - individuals



   - axioms
     - terminological (domain, range, subsumption, etc.)
     - assertional (typing, property assertions)

Our approach: 4 inputs (provided)

4. correspondences between properties: ontology ↔ external resources

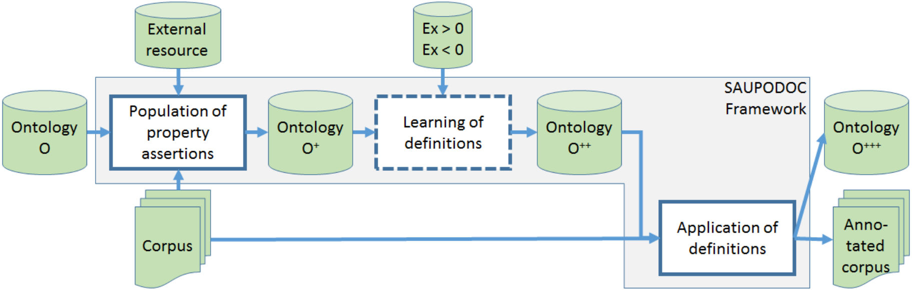


- *document properties*: documents are complete w.r.t. these properties (e.g. hasActivity, hasEnvironment, etc.)
- *external properties*: not mentioned at all in the documents (e.g. avgTemperatureC, precipitationMm, etc.)
  ⇒ external resources needed (Linked Open Data)

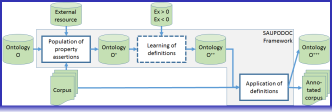
The SAUPODOC approach

= Semantic Annotation Using Population of Ontology and Definitions of Classes

1. corpus of documents
   - one part to be annotated
   - one part annotated for each target concept: positive/negative examples
2. list of target concepts
3. domain ontology
4. correspondences between properties: ontology $\leftrightarrow$ external resources (LOD)



Preliminary task



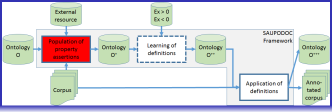
For each document, creation of an instance of the *main class* representing the entity described in the document

Example

```
<destination>
    <name>Dominican Republic</name>
    <description>
        Choose Dominican Republic holidays and escape from it all on a
        Caribbean island with a distinct Latin flavour. The Dominican
        coast offers postcard-perfect sceneries, from white sand beaches to
        jutting mountains and thick rainforests further inland. Influenced
        by its closest island neighbours, Cuba and Puerto Rico, the
        Dominican Republic is a feast of colour. Marvel at the merging of
        tropical blues where the sky touches the water as well as its
        colourful rainbow of traditional painted houses and huts. [...]
    </description>
</destination>
```

$\Rightarrow$ individual Dominican_Republic such as $<$Dominican_Republic isA Destination$>$

Task 1: data extraction from texts

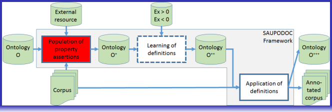


The task adds assertions of *document properties* (ontology population)

Reminder

Documents are complete w.r.t. *document properties*
(e.g. hasActivity, hasEnvironment, etc.)

Task 1: data extraction from texts



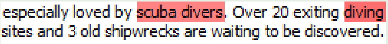
The task adds assertions of *document properties* (ontology population)

Reminder

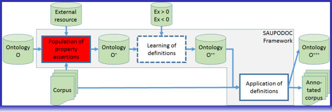
Documents are complete w.r.t. *document properties* (e.g. hasActivity, hasEnvironment, etc.)

Example

- Dominican Republic description:


especially loved by scuba divers. Over 20 exiting diving sites and 3 old shipwrecks are waiting to be discovered.

Task 1: data extraction from texts



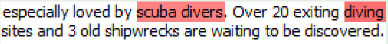
The task adds assertions of *document properties* (ontology population)

- extraction guided by the ontology

☞ GATE - OntoRoot Gazeteer - JAPE transducer (JAPE generic pattern) [Cunningham et al., 2011, Bontcheva et al., 2004]

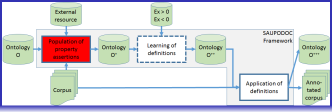
Example

- Dominican Republic description:


especially loved by scuba divers. Over 20 exiting diving sites and 3 old shipwrecks are waiting to be discovered.

- Ontology:

Task 1: data extraction from texts



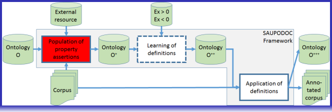
The task adds assertions of *document properties* (ontology population)

- extraction guided by the ontology
    - **by the terms (labels)** related to instances of *descriptive classes*

Example

- Dominican Republic description: especially loved by scuba divers. Over 20 exiting diving sites and 3 old shipwrecks are waiting to be discovered.
- Ontology:
    - "scuba diver" "diving" } **terms** related to the individual **_diving** from the ontology, instance of a subclass of **Activity**

Task 1: data extraction from texts



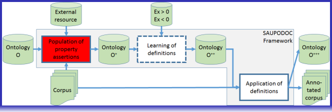
The task adds assertions of *document properties* (ontology population)

- extraction guided by the ontology
  - **by the terms (labels)** related to instances of *descriptive classes*
  - **by the range constraints** of the *document properties*

Example

- Dominican Republic description: especially loved by scuba divers. Over 20 exiting diving sites and 3 old shipwrecks are waiting to be discovered.
- Ontology:
  - "scuba diver" "diving" } **terms** related to the individual **_diving** from the ontology, instance of a subclass of **Activity**
  - <Destination, **hasActivity**, **Activity**>

Task 1: data extraction from texts



The task adds assertions of *document properties* (ontology population)

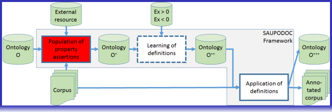
- extraction guided by the ontology
  - **by the terms (labels)** related to instances of *descriptive classes*
  - **by the range constraints** of the *document properties*

Example

- Dominican Republic description: especially loved by scuba divers. Over 20 exiting diving sites and 3 old shipwrecks are waiting to be discovered.
- Ontology:
  - "scuba diver" "diving" } **terms** related to the individual **_diving** from the ontology, instance of a subclass of **Activity**
  - <Destination, **hasActivity**, **Activity**>

⇒ **<Dominican_Republic, hasActivity, _diving>** is built.

Task 2: data completion with LOD (implemented with DBpedia)

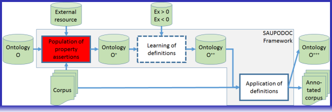


The task adds assertions of *external properties* (ontology population)

Reminder

External properties are not mentioned at all in the documents
(e.g. avgTemperatureC, precipitationMm, etc.)

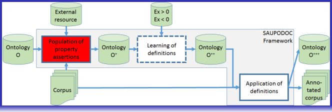
Task 2: data completion with LOD (implemented with DBpedia)



The task adds assertions of *external properties* (ontology population)

1. get DBpedia page (e.g., `http://dbpedia.org/resource/Dominican_Republic`)
   - ☞ DBpedia Spotlight [Mendes et al., 2011]

Task 2: data completion with LOD (implemented with DBpedia)



The task adds assertions of *external properties* (ontology population)

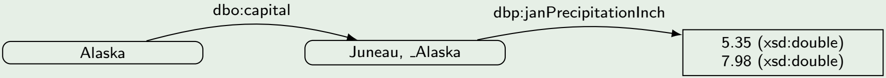
1. get DBpedia page (e.g., `http://dbpedia.org/resource/Dominican_Republic`)
2. automatic generation of SPARQL queries (*CONSTRUCT*) from a model of acquisition [Alec et al., 2016] expressing:
   - correspondences with LOD: complex correspondences

Example

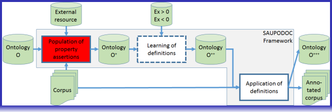
$$\text{precipitation\_in\_January}_{ontology} \equiv \{\text{janPrecipitationMm}, \text{janRainMm}, \text{janPrecipitationInch}, \text{janRainInch}, \text{janPrecipitationIn}, \text{janRainIn}\}_{DBpedia}$$

   - access paths (dealing with incompleteness)

Example



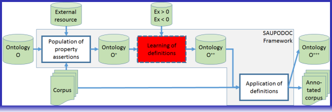
Task 2: data completion with LOD (implemented with DBpedia)



The task adds assertions of *external properties* (ontology population)

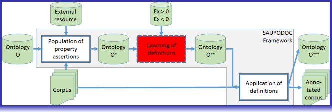
1. get DBpedia page (e.g., `http://dbpedia.org/resource/Dominican_Republic`)
2. automatic generation of SPARQL queries (*CONSTRUCT*) from a model of acquisition [Alec et al., 2016]
3. run queries
   - ☞ DBpedia SPARQL endpoint

Task 3: learning the definitions of target concepts



The task adds definitions of target concepts (ontology enrichment)

Task 3: learning the definitions of target concepts



The task adds definitions of target concepts (ontology enrichment)

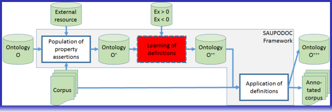
1. learn the definition of each target concept based on
   - manual annotations given by a domain expert
   - the populated ontology

☞ DL-Learner - CELOE algorithm (Inductive Logic Programming) [Lehmann, 2009]

Example

Destination where one can do watersports during Winter ≡
(Destination and (hasActivity some Watersport)
and (hasWeather min 2 ((concernMonth some (hasSeason some MidWinter))
and (avgTemperatureC some double[>= 23.0])
and (precipitationMm some double[<= 70.0])))).

Task 3: learning the definitions of target concepts



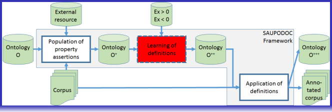
The task adds definitions of target concepts (ontology enrichment)

1. learn the definition of each target concept
2. add target concepts as classes in the ontology
   - as subclasses of the *main class*

Example

<DestinationWithWatersportsDuringWinter, subClassOf, Destination>

Task 3: learning the definitions of target concepts



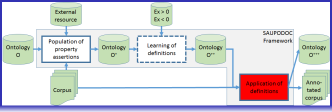
The task adds definitions of target concepts (ontology enrichment)

1. learn the definition of each target concept
2. add target concepts as classes in the ontology
3. add axioms of equivalence between a target concept and its definition

Example

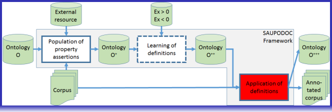
<DestinationWithWatersportsDuringWinter,
owl:equivalentClass,
(Destination and (hasActivity some Watersport)
and (hasWeather min 2 ((concernMonth some (hasSeason some MidWinter))
and (avgTemperatureC some double[>= 23.0])
and (precipitationMm some double[<= 70.0]))))>

Task 4: reasoning to annotate the documents



The task populates the target concepts (ontology population) and annotates documents (semantic annotation of documents)

Task 4: reasoning to annotate the documents



The task populates the target concepts (ontology population) and annotates documents (semantic annotation of documents)

1. apply the definitions on the documents that need to be annotated
   ⇒ target concepts are instanciated
   ☞ FaCT++ [Tsarkov and Horrocks, 2006]

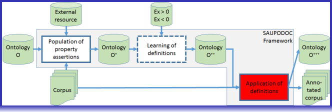
Example

<Dominican_Republic, isA, Destination>
<Dominican_Republic, hasActivity, _diving>
...

DestinationWithWatersportsDuringWinter ≡ (Destination and (hasActivity some Watersport) and ...)

⇒ <Dominican_Republic, isA, DestinationWithWatersportsDuringWinter>

Task 4: reasoning to annotate the documents



The task populates the target concepts (ontology population) and annotates documents (semantic annotation of documents)

1. apply the definitions on the documents that need to be annotated
   $\Rightarrow$ target concepts are instanciated
   ☞ FaCT++ [Tsarkov and Horrocks, 2006]

Example

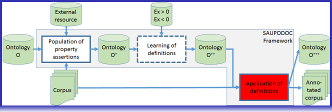
<Dominican_Republic, isA, Destination>
<Dominican_Republic, hasActivity, _diving>
...

DestinationWithWatersportsDuringWinter $\equiv$ (Destination and (hasActivity some Watersport) and ...)

$\Rightarrow$ <Dominican_Republic, isA, DestinationWithWatersportsDuringWinter>

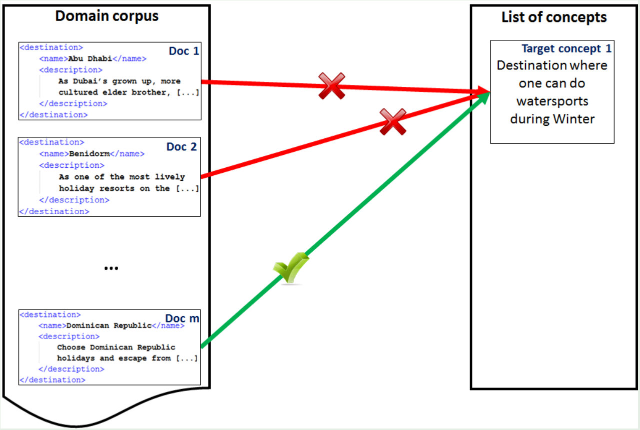
2. get the annotations:
   - document instance of a target concept $\Rightarrow$ positive annotation
   - document not instance of a target concept $\Rightarrow$ negative annotation

Task 4: reasoning to annotate the documents



Example

<Dominican_Republic, isA, DestinationWithWatersportsDuringWinter>



Experimental evaluation: procedure

- For each domain, the set of annotated examples is split:
  - 2/3 training set - 1/3 test set

Experimental evaluation: procedure

- 2/3 training set - 1/3 test set

- Comparison of SAUPODOC with 2 classification approaches
  1. SVM
  2. Decision trees

  all 3 tested with several parameters ⇒ we keep the best results on the test set

Experimental evaluation: procedure

- 2/3 training set - 1/3 test set

1. SVM
2. Decision trees

- For classification approaches:
  - lemmatized bag-of-words TF-IDF
  - dictionary = ontology terminology (labels of individuals)

Example

- SAUPODOC individual of the ontology: "_rainForest" (labels: rain forest, dense forest, tropical forest)
- Classifiers vector component: "_rainForest" (union of words: rain forest, dense forest, tropical forest)

Experimental evaluation: the two tested domains

Destination domain

- 80 documents
- main class = Destination
- 161 descriptive classes
- 39 target concepts

Film domain

- 10,000 documents
- main class = Film
- 5 descriptive classes
- 12 target concepts

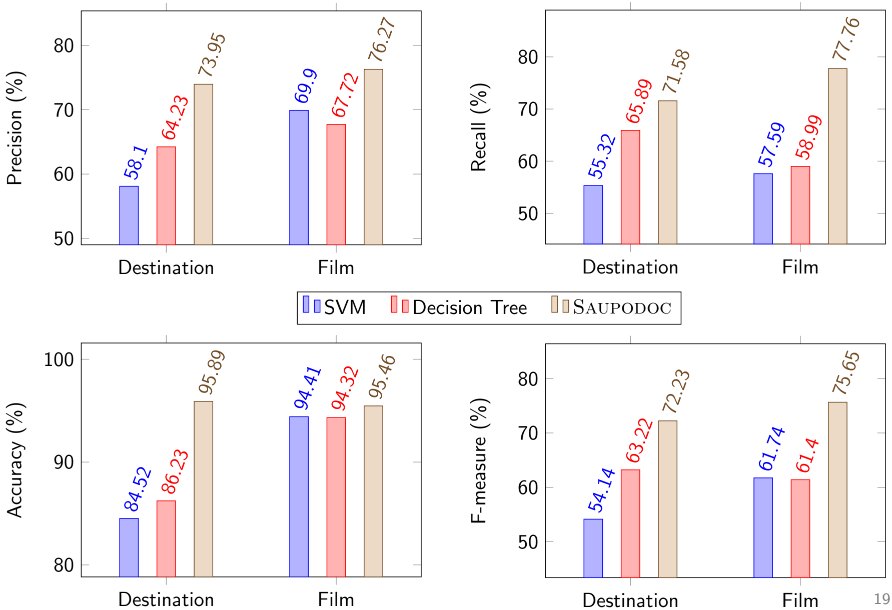
$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Experimental evaluation: average results on the test set



Experimental evaluation: explicit definitions

Collaboration with the Wepingo company

Wepingo recommends entities w.r.t. user needs (target concepts)

- need to have some positive annotations to make recommendations
- need to have intelligible definitions: if 0 positive annotations for a user need ⇒ definition refinement to get "almost positive" annotations

Experimental evaluation: explicit definitions

Collaboration with the Wepingo company

Wepingo recommends entities w.r.t. user needs (target concepts)

- need to have some positive annotations to make recommendations
- need to have intelligible definitions: if 0 positive annotations for a user need $\Rightarrow$ definition refinement to get "almost positive" annotations

- SVM: unintelligible ✘

Experimental evaluation: explicit definitions

Collaboration with the Wepingo company

Wepingo recommends entities w.r.t. user needs (target concepts)

- need to have some positive annotations to make recommendations
- need to have intelligible definitions: if 0 positive annotations for a user need ⇒ definition refinement to get "almost positive" annotations

- SVM: unintelligible ✘
- Decision tree: rules about TF-IDF values ⇒ hard to be adjusted ✘

```
_urban <= 0.018893
|   _beach <= 0
|   |   _sea <= 0.005502: 0
|   |   _sea > 0.005502: 1
|   _beach > 0: 1
_urban > 0.018893: 0
```

Experimental evaluation: explicit definitions

Collaboration with the Wepingo company

Wepingo recommends entities w.r.t. user needs (target concepts)

- need to have some positive annotations to make recommendations
- need to have intelligible definitions: if 0 positive annotations for a user need ⇒ definition refinement to get "almost positive" annotations

- SVM: unintelligible ✘
- Decision tree: rules about TF-IDF values ⇒ hard to be adjusted ✘
- SAUPODOC: explicit definitions ⇒ can be adjusted ✔

Conclusion

- Challenge: annotate a document as a whole with concepts neither explicitly mentioned in the text, nor defined
- Acquisition of data from Linked Open Data (complex task because complex correspondences and incompleteness)
- Use of several tools: possible thanks to the ontology
  - makes the tasks cooperate
  - integrates knowledge
  - enables reasoning
- Experiments with classifiers (no other existing systems)

Perspective

Semi-automatic refinement of the definitions

1. Automatic refinement: make some replacements and keep the candidate definitions that make some "almost positive" annotations

Example

"(hasObjectProperty some A) and (hasDataProperty some double[>= 10.0]) and ..."

Some ideas:

- remove one *and* clause
- replace A by one of its ascendants
- replace 10.0 by a smaller number

2. Manual validation of the candidate definitions

References I

Alec, C., Reynaud-Delaître, C., and Safar, B. (2016).
A Model for Linked Open Data Acquisition and SPARQL Query Generation.
In *International Conference on Conceptual Structures, ICCS*.

Bontcheva, K., Tablan, V., Maynard, D., and Cunningham, H. (2004).
Evolving GATE to Meet New Challenges in Language Engineering.
Natural Language Engineering, 10(3/4):349–373.

Cunningham, H. et al. (2011).
Text Processing with GATE.
University of Sheffield Department of Computer Science.

Lehmann, J. (2009).
DL-Learner: Learning Concepts in Description Logics.
Journal of Machine Learning Research, 10:2639–2642.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011).
DBpedia Spotlight: Shedding Light on the Web of Documents.
In *I-Semantics*, pages 1–8, NY, USA. ACM.

Petasis, G., Möller, R., and Karkaletsis, V. (2013).
BOEMIE: Reasoning-based Information Extraction.
In *LPNMR*, pages 60–75, A Corunna, Spain. CEUR-WS.org.

Tsarkov, D. and Horrocks, I. (2006).
FaCT++ Description Logic Reasoner: System Description.
In *IJCAR*, pages 292–297, Berlin, Heidelberg. Springer.

Yelagina, N. and Panteleyev, M. (2014).
Deriving of Thematic Facts from Unstructured Texts and Background Knowledge.
In *KESW*, volume 468, pages 208–218. Springer.

Thank you for your attention

Questions?

Closed World Assumption

- document not instance of a target concept ⇒ negative annotation
  Closed World Assumption (CWA)
- simulation of CWA at each task
  - task 1: extraction of data from documents: documents are supposed to be complete for all *document properties*
  - task 2: extraction of data from LOD: access paths providing approximate values to overcome incompleteness
  - task 3: learning the definitions:
    - some operators are disabled (NOT, ONLY, etc.)
    - different individuals (*owl:AllDifferent*) ⇒ simulation of Unique Name Assumption (UNA)

    ⇒ same results under CWA and OWA
  - task 4: applying the definitions with a reasoner under OWA: no problem

Model of correspondences: the reasons

Linked Open Data

- Equivalent properties
  janPrecipitationMm, janRainMm, etc.
- Multi-valued properties
  <Juneau_Alaska janPrecipitationInch 5,35>
  <Juneau_Alaska janPrecipitationInch 7,98>
- Unity conversion
  janPrecipitationInch? Mm?
- Properties obtained by transformation
  (janHighC + janLowC) /2

Our ontology

- Functional property
  Aggregation of the values from DBpedia
- Domain constraints
  $\text{precipitationMm}_{ontology} \equiv \{\text{janPrecipitationMm, janRainMm, ...}\}_{DBpedia}$
  iff <domain concernMonth January>