An Unsupervised Method for Ontology Population from the Web

Master candidate: Hilário Tomaz

Supervisor: Fred Freitas

Co-supervisor: Rinaldo Lina

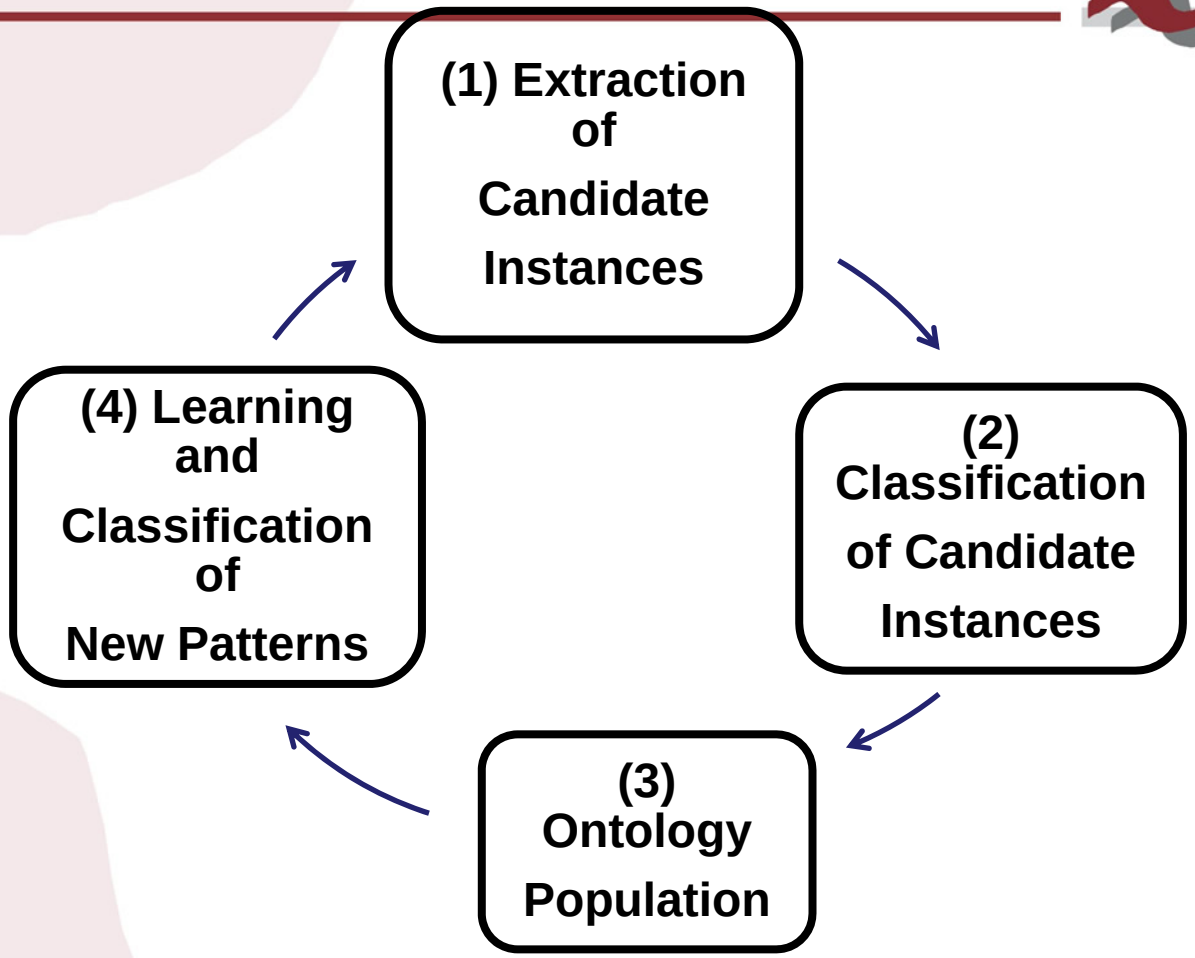
Introduction and Motivation

This work proposes the Unsupervised Method for Ontology Population from Web Texts (**UMOPOW**) that combines:

- web-scale statistics,
- semantic metrics
- other heuristics

for evaluating confidence scores to be applied to the specific task of *Ontology Population* (Oliveira et al., 2012)

The Proposed Approach



Steps of the UMOPOW.

(1) Extraction of Candidate Instances

- We use a set of domain-independent linguistic patterns proposed by Hearst **(Hearst, 1992)**

Ex.: Queries used to gather documents of the *Sport* class

Table 1. Patterns for retrieving relevant documents.

Extraction Pattern	Search Query
class such as **candidates**	"*sports* such as"
such **class** as **candidates**	"such *sports* as"
candidates and other **class**	"and other *sports*"
candidates or other **class**	"or other *sports*"
class especially **candidates**	"*sports* especially"
class including **candidates**	"*sports* including"

(2) Classification of Candidate Instances

- The Confidence Score (*ConfScore*) of a candidate instance as the weighted sum of all constituent scores shown above
  - PMI
  - WordNet Similarity Score (*WNS*)
  - Extra Patterns Score (*EPS*)
  - Direct Matching Score (*DMS*)

$$ConfScore(ci,c) = \frac{EPS + WNS + DMS + 1}{MaxEPS + MaxWNS + MaxDMS + 1} \times PMI(ci,c)$$

(3) Ontology Population

- Typical noisy information found on the Web can produce incorrect candidate instances

- The UMOPOW promotes as actual instances only the best 10 candidate instances classified by our confidence score

(4) Learning Pattern Step

For each instance i of a class C do {

1. Formulate a web query "C AND i" and gather the first N documents
2. For each occurrence of i in the set of the retrieved documents, extract both W words before and after i
3. Apply a filter based on Part-of-Speech (POS) tags

(4) Learning Pattern Step

4. Classify the best patterns according to the confidence metric

$$Conf(p) = \frac{\sum_{i \in I} hits(p,i)}{hits(p)} \times L_{size}$$

where,

- p is a pattern in a set of pattern candidates *P*
- *i* is each instance in *I*
- L_{size} is the number of distinct instances I responsible to extract *p*

Experimental Evaluation

Experimental Evaluation

- **Experimental Setup**

Material Description:

- 7 linguistic patterns
- 700 snippets for each pattern
- Totalizing 4900 snippets
- Custom ontology with 15 classes: *Mammal, Amphibian, Reptile, Bird, Fish, Insect, City, Country, River, Disease, Symptom, Movie, Sport, TV Series, and University*

Experimental Evaluation

Evaluation Measure

Accuracy:

$$P(n) = \frac{number\ of\ correct\ system\ predictions}{n}$$

where:

- *N* is the total of candidate instances evaluated

We determined **4 cut-off** points corresponding to the **Top 10, 25, 50, 100** in the list of candidate results

Experimental Evaluation

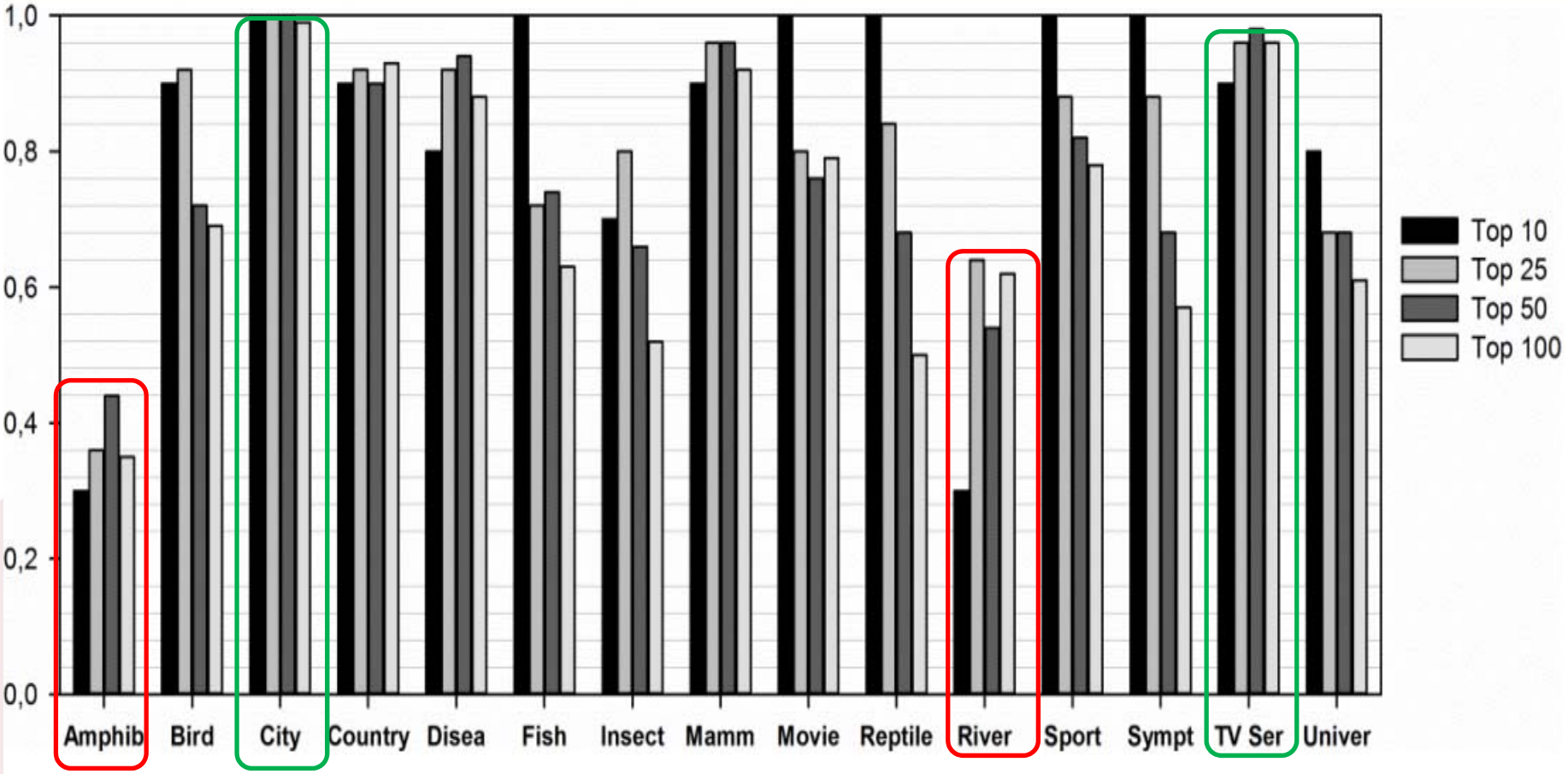


Figure 2. Classification results in Top N.

Experimental Evaluation

- The best 5 learned linguistic pattern order by our Confidence Score (*CS*)

Table 1. Extraction results of learned patterns.

Pattern for the City Class	CS	P	Pattern for the Country Class	CS	P
the city of CANDIDATE	0,1	0,75	CANDIDATE is a country in	0,1	0,83
city university of *CANDIDATE*	0,91	0,09	the country of CANDIDATE	0,86	0,24
cities in CANDIDATE	0,86	0,01	CANDIDATE a country study	0,74	0,80
the great city of CANDIDATE	0,66	0,88	CANDIDATE is a big country	0,63	0,52
heart of the city of CANDIDATE	0,60	0,97	country code of CANDIDATE	0,52	0,59

Conclusion and Future Work

Conclusion and Future Work

- We have proposed an unsupervised method for ontology population, the UMOPOW, which is based on a confidence-weighted metric for assessing candidate instances extracted from the web.

- Future Work:
  - Evaluating the impact of each measure and heuristic that composed the Confidence-weighted Metric
  - Evaluate the impact on precision when the learning module considers, at the same time, the initial and the learned set of extraction patterns

References

1. Berners-Lee, T., Hendler J., Lassila, O.: The Semantic Web. Scientific American 284, pp. 34–43 (2001)
2. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Ap-plications. Springer-Verlag, New York (2006)
3. Wimalasuriya, D. C., Dou, D.: Ontology-based Information Extraction: An introduction and a Survey of Current Approaches. J. of Information Science 36, pp. 306-323 (2010)
4. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Web-Scale Information Extraction in KnowItAll. In: 13th International World Wide Web Conference, pp. 100–110, New York (2004)
5. McDowell, L. K., Cafarella, M.: Ontology-Driven, Unsupervised Instance Population. J. Web Semantics: Science, Services and Agents on the World Wide Web 6, 218-236 (2008)
6. Geleijnse G., Korst, J.: Learning effective surface text patterns for information extraction. In: 11th Conference of the European Chapter of the Association for Computational Linguis-tics Workshop on Adaptive Text Extraction and Mining, pp. 1–8, Trento (2006)

References

7. Downey, D., Etzioni, O., Weld, D. S., Soderland, S.: Learning Text Patterns for Web Infor-mation Extraction and Assessment. In: 19th National Conference on Artificial Intelligence Workshop on Adaptive Text Extraction and Mining, San Jose (2004)
8. Hearst, M.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: 14 th Conference on Computational linguistics, pp. 539 - 545, Nantes (1992)
9. Oliveira, H., Lima, R., Gomes, J., Ferreira, R., Freitas, F., Costa, E.: A Confidence-Weighted Metric for Unsupervised Ontology Population from Web Texts. In: 23rd International Conference on Database and Expert Systems Applications, Vienna (2012) (to appear)
10. Pedersen, T.: Information Content Measures of Semantic Similarity Perform Better Without Sense-Tagged Text. In: 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 329-332, Los Angeles (2010)
11. Lin, D.: An Information-Theoretic Definition of Similarity. In: 15th International Conference on Machine Learning, pp. 296 - 304, Madison (1998)
12. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection. In: 32nd Annual Meeting of the Association for Computational Linguistics, pp. 133 –138, New Mexico (1994)

An Unsupervised Method for Ontology Population from the Web

Questions ???

Hilário Tomaz (htao@*cin.ufpe.br*)

Fred Freitas (fred@*cin.ufpe.br*)

Rinaldo Lima (*rjl4@cin.ufpe.br*)

Appendix

(1) Extraction of Candidate Instances

- The documents are preprocessed in order to:
  - eliminate unnecessary elements. **E.g.** HTML tags
  - match occurrences of the patterns into the document content
    - Ex. Sentences like *"such cities as CANDIDATES"* in which **CANDIDATES** denotes a list of **Noun Phrases (NP)**

(1) Extraction of Candidate Instances

- In order to avoid invalid and repeated candidate instances, the following filters are applied:
  1. **Stop word filtering**
  2. **Redundant candidates**
  3. **Semantic filtering**
  - 2 candidate instances can be syntactically different, but semantically equivalent
    - EX.: "USA" and "The United States of America"
    - if two candidates are synonyms according to the **WordNet**, they are considered equivalents and just one candidate is kept

(2) Classification of Candidate Instances

1. **Pointwise Mutual Information (PMI)** **(Etzioni et al., 2004)**
   - It assesses the likelihood between the *class* and the *candidate instance* using each extraction pattern
     - Queries are performed using a web search engine

(2) Classification of Candidate Instances

- PMI Scores for candidate instances of Disease class

Extraction Pattern	Query	Hits
candidate is a **class**	"**pneumonia** is a **disease**"	115,000
class(s) such as **candidates**	"**diseases** such as **pneumonia and**"	365,000
such **class(s)** as **candidates**	"such **diseases** as pneumonia **and**"	21,400
candidates and other **class(s)**	"**pneumonia** and other **diseases and**"	447,000
candidates or other **class(s)**	"**pneumonia** or other **diseases and**"	536,000
class(s) especially **candidates**	"**diseases** especially **pneumonia and**"	6,980
class(s) including **candidates**	"**diseases** including **pneumonia and**"	67,500

(2) Classification of Candidate Instances

- It is important the presence of the word "and", either before or after a candidate instance in some patterns **(Geleijnse and Korst, 2006)**

- Using it we try to avoid some misclassifications faced in our previously work **(Oliveira et al, 2012)**

a) If the method had extracted the candidate instance "Las" instead of "Las Vegas"

b) Candidate instances matching the pattern "NP and/or NP" like "New York and California"

(2) Classification of Candidate Instances

- Str-INorm-Thresh
  - A variation of PMI that normalized the sum of the hits(*ci*,*c*,*p*) by a value determined by sorting the set of candidate instances by hits(*ci*) and then selecting the hit count that appears at the 25th percentile (*Count25*)

$$\text{Str-INorm-Thresh(ci,c)} = \frac{\sum_{p \in P} hits(ci, c, p)}{\max(hits(ci), Count_{25})}$$

(2) Classification of Candidate Instances

2. WordNet Similarity

– Take into account the WordNet structure to produce a numerical value for assessing the degree of the semantic similarity between two concepts

We adopted two similarity measures:

- **Lin (Lin, 1998):** defines the similarity between two concepts as the ratio of the **shared information content** to the **information content** that separately describe each concept
- **Wu and Palmer (Wu and Palmer, 1994):** relies on finding the most specific concept that subsumes both the concepts under measurement.

(2) Classification of Candidate Instances

3. Number of Extra Patterns

- If a candidate instance is extracted by many extraction patterns, this gives a strong evidence that this candidate instance is a valid instance for the related class

- Extra Pattern Score (*EPS*) is the number of extraction patterns that extracted a particular candidate instance

(2) Classification of Candidate Instances

4. Direct Matching

- Based on the idea of finding the label of the class within the instance candidate **(Monllaó, 2011)**
- If they match, then the system assigns 1 as its Direct Matching Score (DMS), or 0 otherwise

Example:

- Given the **University class** and the candidate instance, **University of London**, then DMS = 1 is assigned to this candidate