Ontologising Semantic Relations into a Relationless Thesaurus

FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE DE COIMBRA

Hugo Gonçalo Oliveira & Paulo Gomes
{hroliv,pgomes}@dei.uc.pt
Cognitive & Media Systems Group
CISUC, University of Coimbra, Portugal

cognitive & media systems

1. Introduction

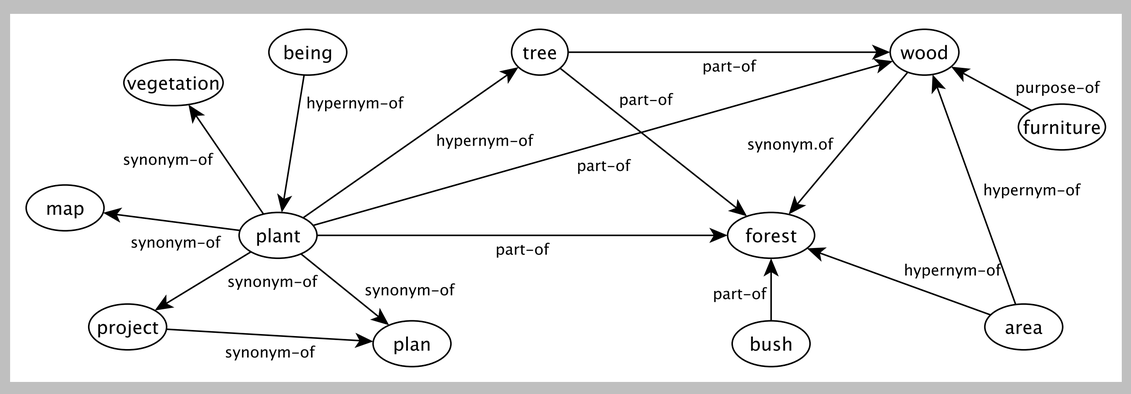
- Information Extraction (IE) systems typically represent semantic relations as term-based triples (tb-triples)
  - t = (a R b) (e.g. *machine* hypernym-of *computer*)
  - one sense of term a is related to one sense of b, by means of relation R
- But natural language is ambiguous...
  - Terms are not enough to unambiguously refer to a concept!
- Possible solution: **ontologising** [1]
  - Moving towards an ontological structure
  - e.g. **wordnet** (as Princeton WordNet [2])
    - **synsets** denote concepts, e.g.:
      A ={*machine*.1}
      B ={*computer*.1, *computing_machine*.1, *computing_device*.1, *data_processor*.1}
      C ={*chip*.7, *microchip*.1, *microprocessor_chip*.1}
    - **relations** between synsets (sb-triples), e.g.: A hypernym-of B, C part-of B, ...

- **Research Goal**: ontologising...
  - **without considering the extraction context**
  - **in a wordnet without glosses nor connections between synsets**

- IE system with **two independent modules**:
  1. Relation extraction
  2. Argument ontologisation

2. Proposed Approach

- Ontologise a tb-triple {a R b}, in a thesaurus T with synsets
- In other words, attach terms in a tb-triple
  - a and b
- To suitable synsets
  - $A_i \in T$ and $B_j \in T$
  - $A_i = \{a_{i0}, a_{i1}, ..., a_{in}\}$, $B_j = \{b_{j0}, b_{j1}, ..., b_{jn}\}$
- Selected from the candidates:
  - for a term a, $A : \forall(A_i \in A) \rightarrow a \in A_i$
  - for a term b, $B : \forall(B_j \in B) \rightarrow b \in B_j$
- **Key:** exploit all extracted information – lexical network $N = (V, E)$
  - Nodes (V) are terms (*plant*, *forest*, *being*, ...)
  - Edges ($E \in V^2$) are tb-triples (*plant* part-of *forest*, *being* hypernym-of *plant*, ...)



- Resulting in a **sb-triple**, st = {A_i R B_j}

3. Experimentation Set-up

- Gold reference:
  - Correct Portuguese tb-triples of PAPEL 2.0 [3] (452 of hypernymy, part-of, purpose-of)
  - Synsets of two Portuguese thesauri (TeP 2.0 [4] + OpenThesaurus.PT)
  - Manual annotation of plausible attachments
- Gold entries:

tb-triple = {*planta* **part-of** *floresta*} (*plant* part-of *forest*)	
A_1: *relação, quadro, planta, mapa* A_2: *vegetal, planta* A_3: *traçado, desenho, projeto, planta, plano*	B_1: *bosque, floresta, mata, brenha, selva*
plausible sb-triples = {A_2, B_1}	
tb-triple = {*passageiro* **purpose-of** *carruagem*} (*passenger* purpose-of *carriage*)	
A_1: *passageiro, viajante* A_2: *passageiro, viador* A_3: *passageiro, transeunte*	B_1: *carriagem, carruagem, carraria* B_2: *carruagem, carro, sege, coche* B_3: *carruagem;caleça;caleche* B_4: *actividade, carruagem, operosidade, diligência*
plausible sb-triples = {A_1, B_1}, {A_1, B_2}, {A_1, B_3}, {A_2, B_1}, {A_2, B_2}, {A_2, B_3}	

- On average, for hypernymy, part-of and purpose-of:
  - Attachment alternatives for each tb-triple: 13.7, 11.2 and 13.5
  - Random chance of selecting a correct attachment: **40.4%**, **49.6%** and **50.1%**

Acknowledgements

This work was developed in the scope of the project Onto.PT [5].
Hugo Gonçalo Oliveira is supported by FCT, grant SFRH/BD/44955/2008, co-funded by FSE.

4. Ontologising Algorithms

- **Related Proportion (RP)**
  1. For attaching term a, fix b
  2. For each synset $A_i \in A$, $n_i = |a_{ik} \in A_i : E(a_{ik}, R, b) \in N|$
  3. Compute the related proportion rp:

$$rp(A_i, \{a, R, b\}) = \frac{n_i}{1 + log_2(|A_i|)}$$

  4. If $rp \geq \theta$, $C' = \{A_l \in A : rp_{A_l,\{a,R,b\}} = max(rp)\}$. Otherwise, do not perform attachment.
  5. Attach a to $A_l \in C'$, such that $n_k = max(n_i)$
  6. Attach term b using the same procedure, but fixing a.

- **Average Cosine (AC)**
  1. Represent candidate synsets, $A_i \in A$ and $B_j \in B$, as adjacency vectors
     - $\vec{A}_i = \{\vec{a}_{i0}, \vec{a}_{i1}, ..., \vec{a}_{in}\}$, $n = |A_i|$
     - $\vec{B}_j = \{\vec{b}_{j0}, \vec{b}_{j1}, ..., \vec{b}_{jm}\}$, $m = |B_j|$
  2. Select the most similar pair of candidates:

$$sim(A_i, B_j) = \frac{\sum_{k=1}^{|A_i|} \sum_{l=1}^{|B_j|} \cos(\vec{a}_{ik}, \vec{b}_{jl})}{|A_i||B_j|}$$

- **RP+AC**
  1. Use RP with a high θ
  2. If RP does not select a suitable candidate, use AC

- **(Average) Number of Triples (NT)**
  1. Score pairs of candidate synsets according to the number of tb-triples of type R between any of their terms

$$nt(A_i, B_j) = \frac{\sum_{k=1}^{|A_i|} \sum_{l=1}^{|B_j|} E(a_{ik}, R, b_{jl}) \in E}{log_2(|A_i||B_j|)}$$

  2. Select the best ranked pair

- **PageRank (PR)**
  1. Give initial weights of 0.5 for the nodes with a and b, and 0 to the others
  2. Run PageRank [6] on N for several iterations
  3. Score each synset with the average PageRank of the terms it includes:

$$\overline{PR}(A_i) = \frac{\sum_{k=1}^{|A_i|} PR(a_{ik})}{1 + log_2(|A_i|)}$$

  4. Select the pair $\{A_i, B_j\}$ maximising $\overline{PR}(A_i)$ and $\overline{PR}(B_j)$

- **Minimum Distance (MD)**
  1. Compute the distance between each pair of candidate synsets, given the average distance (edges between) of their terms:

$$\overline{dist}(A_i, B_j) = \frac{\sum_{k=1}^{|A_i|} \sum_{l=1}^{|B_j|} dist(a_{ik}, b_{jl})}{|A_i||B_j|}$$

  2. Select the closest pair of synsets

5. Algorithms Performance

Algorithm	**Hypernym-of** (210 tb-triples)					**Part-of** (175 tb-triples)					**Purpose-of** (67 tb-triples)				
	P %	**R %**	**F_1%**	**$F_{0.5}$%**	**RF_1%**	**P %**	**R %**	**F_1%**	**$F_{0.5}$%**	**RF_1%**	**P %**	**R %**	**F_1%**	**$F_{0.5}$%**	**RF_1%**
RP	53.8	12.4	20.2	32.3	50.3	56.9	10.6	17.9	30.4	47.0	51.5	5.1	9.3	18.3	32.6
AC	**60.1**	15.7	24.9	38.4	**59.8**	58.7	14.9	23.8	37.0	58.7	63.2	13.0	21.5	35.6	63.2
RP+AC	56.3	14.9	23.6	36.2	56.1	**63.3**	**16.3**	**25.9**	**40.1**	**63.3**	**63.4**	13.6	22.3	**36.5**	**63.4**
NT	57.7	**17.3**	**26.6**	**39.4**	57.7	50.7	15.8	24.1	35.2	50.7	48.1	**15.4**	**23.3**	33.7	48.1
PR	46.2	11.5	18.5	28.9	45.7	50.6	12.6	20.2	31.6	49.9	56.3	10.8	18.2	30.6	56.3
MD	58.6	15.8	24.9	38.0	58.6	59.1	15.3	24.3	37.6	59.1	60.9	12.7	20.9	34.5	60.9

5. Conclusions

- Ontologising is a challenging task, especially without a context
- Still, the precision of the algorithms outperform the random chance
- Considering RF_1 measure, the best algorithms are
  - AC for hypernymy
  - RP+AC for part-of and purpose-of

References

[1] M. Pennacchiotti and P. Pantel, "Ontologizing semantic relations," in *Proc. 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the ACL*, pp. 793–800, ACL Press, 2006.

[2] C. Fellbaum, ed., *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. MIT Press, 1998.

[3] H. Gonçalo Oliveira, D. Santos, and P. Gomes, "Relations extracted from a portuguese dictionary: results and first evaluation," in *Local Proc. 14th Portuguese Conf. on Artificial Intelligence (EPIA)*, pp. 541–552, APPIA, 2009.

[4] E. Maziero, T. Pardo, A. D. Felippo, and B. C. Dias-da-Silva, "A base de dados lexical e a interface web do TeP 2.0 - thesaurus eletrônico para o português do Brasil," in *VI Workshop em Tecnologia da Informação e Linguagem Humana (TIL)*, pp. 390–392, 2008.

[5] H. Gonçalo Oliveira and P. Gomes, "Integrating lexical-semantic knowledge to build a public lexical ontology for Portuguese," in *Natural Language Processing and Information Systems, Proc. 17h NLDB*, LNCS 7337, (Groningen, The Netherlands), pp. 210–215, Springer, 2012.

[6] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 30, no. 1-7, pp. 107–117, 1998.