Automatically Enriching a Thesaurus with Information from Dictionaries

Hugo Gonçalo Oliveira[1] Paulo Gomes

{hroliv,pgomes}@dei.uc.pt

Cognitive & Media Systems Group
CISUC, Universidade de Coimbra

October 11, 2011

[1]supported by FCT scholarship grant SFRH/BD/44955/2008

Lexical knowledge bases

- Thesaurus, lexical networks, lexical ontologies, ...

Lexical knowledge bases

- Thesaurus, lexical networks, lexical ontologies, ...
- Structured on **words** and their **meanings**

Lexical knowledge bases

- Thesaurus, lexical networks, lexical ontologies, ...
- Structured on **words** and their **meanings**
- Try to cover the **whole language**

Lexical knowledge bases

- Thesaurus, lexical networks, lexical ontologies, ...
- Structured on **words** and their **meanings**
- Try to cover the **whole language**
- **No specific domain**

Lexical knowledge bases

- Thesaurus, lexical networks, lexical ontologies, ...
- Structured on **words** and their **meanings**
- Try to cover the **whole language**
- **No specific domain**
- Essential for developing NLP tools for a language

Lexical knowledge bases

- Thesaurus, lexical networks, lexical ontologies, ...
- Structured on **words** and their **meanings**
- Try to cover the **whole language**
- **No specific domain**
- Essential for developing NLP tools for a language
  - Useful for NLP tasks (eg. word-sense disambiguation, question-answering, determining similarities, ...)

Lexical knowledge bases

- Thesaurus, lexical networks, lexical ontologies, ...
- Structured on **words** and their **meanings**
- Try to cover the **whole language**
- **No specific domain**
- Essential for developing NLP tools for a language
  - Useful for NLP tasks (eg. word-sense disambiguation, question-answering, determining similarities, ...)
  - See Princeton WordNet [Fellbaum, 1998]

Free lexical knowledge bases for Portuguese

- Public domain thesaurus:
  - TeP [Maziero et al., 2008]
  - OpenThesaurus.PT[2]

[2]http://openthesaurus.caixamagica.pt/
[3]http://pt.wiktionary.org/

Free lexical knowledge bases for Portuguese

- Public domain thesaurus:
  - TeP [Maziero et al., 2008]
  - OpenThesaurus.PT[2]
- Collaborative dictionary
  - Portuguese Wiktionary[3]

[2] http://openthesaurus.caixamagica.pt/
[3] http://pt.wiktionary.org/

Free lexical knowledge bases for Portuguese

- Public domain thesaurus:
  - TeP [Maziero et al., 2008]
  - OpenThesaurus.PT[2]
- Collaborative dictionary
  - Portuguese Wiktionary[3]
- Public domain lexical network
  - PAPEL [Gonçalo Oliveira et al., 2010]

[2] http://openthesaurus.caixamagica.pt/
[3] http://pt.wiktionary.org/

Free lexical knowledge bases for Portuguese

- Public domain thesaurus:
  - TeP [Maziero et al., 2008]
  - OpenThesaurus.PT[2]
- Collaborative dictionary
  - Portuguese Wiktionary[3]
- Public domain lexical network
  - PAPEL [Gonçalo Oliveira et al., 2010]
- Lexical ontology [coming soon]
  - Onto.PT [Gonçalo Oliveira and Gomes, 2010]

[2] http://openthesaurus.caixamagica.pt/
[3] http://pt.wiktionary.org/

Free lexical knowledge bases for Portuguese

- Public domain thesaurus:
  - TeP [Maziero et al., 2008]
  - OpenThesaurus.PT[2]
- Collaborative dictionary
  - Portuguese Wiktionary[3]
- Public domain lexical network
  - PAPEL [Gonçalo Oliveira et al., 2010]
- Lexical ontology [coming soon]
  - Onto.PT [Gonçalo Oliveira and Gomes, 2010]

- More **complementary** than overlapping

[2] http://openthesaurus.caixamagica.pt/
[3] http://pt.wiktionary.org/

Free lexical knowledge bases for Portuguese

- Public domain thesaurus:
  - TeP [Maziero et al., 2008]
  - OpenThesaurus.PT[2]
- Collaborative dictionary
  - Portuguese Wiktionary[3]
- Public domain lexical network
  - PAPEL [Gonçalo Oliveira et al., 2010]
- Lexical ontology [coming soon]
  - Onto.PT [Gonçalo Oliveira and Gomes, 2010]

- More **complementary** than overlapping
- Fruitful to **merge** some of them in a unique broader resource

[2] http://openthesaurus.caixamagica.pt/
[3] http://pt.wiktionary.org/

This work

- **Integrate synonymy information from dictionaries in a thesaurus**

This work

- **Integrate synonymy information from dictionaries in a thesaurus**
  1. Extraction of synpairs from dictionaries

This work

- **Integrate synonymy information from dictionaries in a thesaurus**
  1. Extraction of synpairs from dictionaries
  2. Assigning synpairs to synsets

This work

- **Integrate synonymy information from dictionaries in a thesaurus**

  1. Extraction of synpairs from dictionaries
  2. Assigning synpairs to synsets
  3. Clustering remaining pairs

This work

- **Integrate synonymy information from dictionaries in a thesaurus**
  1. Extraction of synpairs from dictionaries
  2. Assigning synpairs to synsets
  3. Clustering remaining pairs
- Apply the procedure in the enrichment of TeP with PAPEL

Extracting synpairs from dictionaries

- **mente**, n: *cérebro, cabeça, intelecto*
  [**mind**, n: *brain, head, intellect*]

- **máquina**, n: **o mesmo que** *computador*
  [**machine**, n: *the same as computer*]

Extracting synpairs from dictionaries

- **mente**, n: *cérebro, cabeça, intelecto*

  [**mind**, n: *brain, head, intellect*]

  - (*cérebro*, *mente*) (*cabeça*, *mente*) (*intelecto*, *mente*)

    [(*brain*, *mind*) (*head*, *mind*) (*intellect*, *mind*)]

- **máquina**, n: **o mesmo que** *computador*

  [**machine**, n: *the same as computer*]

Extracting synpairs from dictionaries

- **mente**, n: *cérebro, cabeça, intelecto*

  [**mind**, n: *brain, head, intellect*]

  - (*cérebro*, *mente*) (*cabeça*, *mente*) (*intelecto*, *mente*)

    [(*brain*, *mind*) (*head*, *mind*) (*intellect*, *mind*)]

- **máquina**, n: **o mesmo que** *computador*

  [**machine**, n: *the same as computer*]

  - (*computador*, *máquina*)

    [(*computer*, *machine*)]

Assigning synpairs to synsets

- $p = (w_x, w_y) + S_a = (w_1, w_2, ..., w_n) \rightarrow S_a = (w_1, w_2, ..., w_n, w_x, w_y)$

Assigning synpairs to synsets

- $p = (w_x, w_y) + S_a = (w_1, w_2, ..., w_n) \rightarrow S_a = (w_1, w_2, ..., w_n, w_x, w_y)$
- Synonymy graph G
  - All the extracted synpairs
  - Nodes represent words (eg. w_x, w_y)
  - $p = (w_x, w_y)$ establishes an edge between w_x and w_y

Assigning synpairs to synsets

For each synpair $p = (w_x, w_y)$

1. If $S_i \in T : w_x \in S_i \wedge w_y \in S_i$, nothing is done.

[a] Any measure for computing the similarity of two vectors can be used

Assigning synpairs to synsets

For each synpair $p = (w_x, w_y)$

1. If $S_i \in T : w_x \in S_i \wedge w_y \in S_i$, nothing is done.
2. Select all synsets $C_j \in C : C \subset T, C = \{C_1, C_2, ..., C_n\}$
   $\forall (C_j \in C) : w_x \in C_j \vee w_y \in C_j$.

[a] Any measure for computing the similarity of two vectors can be used

Assigning synpairs to synsets

For each synpair $p = (w_x, w_y)$

1. If $S_i \in T : w_x \in S_i \wedge w_y \in S_i$, nothing is done.
2. Select all synsets $C_j \in C : C \subset T, C = \{C_1, C_2, ..., C_n\}$
   $\forall (C_j \in C) : w_x \in C_j \vee w_y \in C_j$.
3. If $|C| = 1$, $p + C_1$.

[a] Any measure for computing the similarity of two vectors can be used

Assigning synpairs to synsets

For each synpair $p = (w_x, w_y)$

1. If $S_i \in T : w_x \in S_i \wedge w_y \in S_i$, nothing is done.
2. Select all synsets $C_j \in C : C \subset T, C = \{C_1, C_2, ..., C_n\}$ $\forall (C_j \in C) : w_x \in C_j \vee w_y \in C_j$.
3. If $|C| = 1$, $p + C_1$.
4. Compute the adjacency vector $[p] = [w_x] + [w_y]$. The adjacency vector of a word is a column of the matrix M, $[w_j] = [M_j]$;

[a] Any measure for computing the similarity of two vectors can be used

Assigning synpairs to synsets

For each synpair $p = (w_x, w_y)$

1. If $S_i \in T : w_x \in S_i \wedge w_y \in S_i$, nothing is done.
2. Select all synsets $C_j \in C : C \subset T, C = \{C_1, C_2, ..., C_n\}$
   $\forall (C_j \in C) : w_x \in C_j \vee w_y \in C_j$.
3. If $|C| = 1$, $p + C_1$.
4. Compute the adjacency vector $[p] = [w_x] + [w_y]$. The adjacency vector of a word is a column of the matrix M, $[w_j] = [M_j]$;
5. Compute the adjacency vector of each $C_j \in C$
   $[C_j] = \sum_{k=1}^{|C_j|} [w_k] : w_k \in C_j$;

[a] Any measure for computing the similarity of two vectors can be used

Assigning synpairs to synsets

For each synpair $p = (w_x, w_y)$

1. If $S_i \in T : w_x \in S_i \wedge w_y \in S_i$, nothing is done.
2. Select all synsets $C_j \in C : C \subset T, C = \{C_1, C_2, ..., C_n\}$
   $\forall (C_j \in C) : w_x \in C_j \vee w_y \in C_j$.
3. If $|C| = 1$, $p + C_1$.
4. Compute the adjacency vector $[p] = [w_x] + [w_y]$. The adjacency vector of a word is a column of the matrix M, $[w_j] = [M_j]$;
5. Compute the adjacency vector of each $C_j \in C$
   $[C_j] = \sum_{k=1}^{|C_j|} [w_k] : w_k \in C_j$;
6. Select the most similar synset
   $C_{best} : sim(p, C_{best})^a = max(sim(p, C_j))$;

[a] Any measure for computing the similarity of two vectors can be used

Assigning synpairs to synsets

For each synpair $p = (w_x, w_y)$

1. If $S_i \in T : w_x \in S_i \wedge w_y \in S_i$, nothing is done.
2. Select all synsets $C_j \in C : C \subset T, C = \{C_1, C_2, ..., C_n\}$
   $\forall (C_j \in C) : w_x \in C_j \vee w_y \in C_j$.
3. If $|C| = 1$, $p + C_1$.
4. Compute the adjacency vector $[p] = [w_x] + [w_y]$. The adjacency vector of a word is a column of the matrix M, $[w_j] = [M_j]$;
5. Compute the adjacency vector of each $C_j \in C$
   $[C_j] = \sum_{k=1}^{|C_j|} [w_k] : w_k \in C_j$;
6. Select the most similar synset
   $C_{best} : sim(p, C_{best})^a = max(sim(p, C_j))$;
7. $p + C_{best}$.

[a] Any measure for computing the similarity of two vectors can be used

Clustering remaining pairs

G' is established by the remaining pairs

1. Sparse matrix $M'(|N| \times |N|)$

Clustering remaining pairs

G' is established by the remaining pairs

1. Sparse matrix $M'(|N| \times |N|)$
2. $M'_{ij} = sim([w_i], [w_j])$

Clustering remaining pairs

G' is established by the remaining pairs

1. Sparse matrix $M'(|N| \times |N|)$
2. $M'_{ij} = sim([w_i], [w_j])$
3. Normalise the columns of M, so that $\sum_{k=1}^{|M_j|} M_{jk} = 1$

Clustering remaining pairs

G' is established by the remaining pairs

1. Sparse matrix $M'(|N| \times |N|)$
2. $M'_{ij} = sim([w_i], [w_j])$
3. Normalise the columns of M, so that $\sum_{k=1}^{|M_j|} M_{jk} = 1$
4. Extract cluster S_i from each row M'_i, with the words w_j where $M'_{ij} > \theta$

Clustering remaining pairs

G' is established by the remaining pairs

1. Sparse matrix $M'(|N| \times |N|)$
2. $M'_{ij} = sim([w_i], [w_j])$
3. Normalise the columns of M, so that $\sum_{k=1}^{|M_j|} M_{jk} = 1$
4. Extract cluster S_i from each row M'_i, with the words w_j where $M'_{ij} > \theta$
5. For each $S_i : S_i \cup S_j = S_j$ and $S_i \cap S_j = S_i$, S_i is discarded.

Coverage of the synpairs by TeP

POS	Synpairs	In TeP	$\|C\|^4 = 0$	$\|C\| = 1$	$\|C\| > 1$	$\|\overline{C}\|$
Nouns	37,452	27.38%	14.98%	12.01%	45.63%	3.86
Verbs	21,465	43.01%	1.34%	4.04%	51.66%	6.64
Adjectives	19,073	37.60%	5.58%	8.22%	48.60%	4.26

[4] Number of candidate synsets

Coverage of the synpairs by TeP

POS	Synpairs	In TeP	$\|C\|^4 = 0$	$\|C\| = 1$	$\|C\| > 1$	$\|\overline{C}\|$
Nouns	37,452	27.38%	14.98%	12.01%	45.63%	3.86
Verbs	21,465	43.01%	1.34%	4.04%	51.66%	6.64
Adjectives	19,073	37.60%	5.58%	8.22%	48.60%	4.26

- Experimentation was performed using the cosine similarity

[4] Number of candidate synsets

Results – words

Thesaurus	POS	Words			
		Total	Ambiguous	Avg(senses)	Most ambig.
TeP 2.0	Nouns	17,158	5,805	1.71	20
	Verbs	10,827	4,905	2.08	41
	Adjectives	14,586	3,735	1.46	19
After assignments	Nouns	23,775	10,418	2.09	37
	Verbs	12,818	7,094	2.64	42
	Adjectives	17,158	6,294	1.83	22
Clusters	Nouns	8,546	701	1.15	8
	Verbs	502	8	1.02	3
	Adjectives	1,858	39	1.03	4
Final thesaurus	Nouns	30,369	12,045	1.96	38
	Verbs	13,090	7,221	2.62	42
	Adjectives	18,525	6,550	1.80	23

Results – synsets

Thesaurus	POS	Synsets				
		Total	**Avg(size)**	*size* = 2	*size* > 25	*max*(*size*)
TeP 2.0	Nouns	8,254	3.56	3,079	0	21
	Verbs	3,978	5.67	939	48	53
	Adjectives	6,066	3.50	3,033	19	43
After assignments	Nouns	8,254	6.01	1,930	179	150
	Verbs	3,978	8.50	702	217	148
	Adjectives	6,066	5.17	2,369	120	110
Clusters	Nouns	3,524	2.78	2,247	0	13
	Verbs	220	2.34	174	0	6
	Adjectives	820	2.33	656	0	10
Final thesaurus	Nouns	11,778	5.05	4,177	179	150
	Verbs	4,198	8.18	876	217	148
	Adjectives	6,886	4.84	3,025	120	110

Assignments evaluation

- Manual evaluation of sample assignments
- Two judges for each assignment

Assignments evaluation

- Manual evaluation of sample assignments
- Two judges for each assignment

POS	Sample	Correct		Incorrect		Agreement
Nouns	100 assigns. × 2	153	(76.50%)	47	(23.50%)	77.00%
Verbs	100 assigns. × 2	142	(71.00%)	58	(29.00%)	74.00%
Adjectives	100 assigns. × 2	151	(75.50%)	49	(24.50%)	75.00%

Assignments evaluation

- Manual evaluation of sample assignments
- Two judges for each assignment

POS	Synpair	Synset	Judge 1	Judge 2
Nouns	(**escrutínio**,votação)	votação;voto;sufrágio	1	1
	(decisão,**desempate**)	resolução;objetivação;tenção;intenção	0	1
	(plano,**gizamento**)	planície;chã;chanura;plaino;plano;planura	0	0
Verbs	(venerar,**homenagear**)	venerar;cultuar;adorar;idolatrar	1	1
	(atacar,**combater**)	atacar;inciar	0	1
	(**obter**,rapar)	depilar;despelar;pelar;raspar;rapar;rascar	0	0
Adjectives	(grandioso,**épico**)	admirável;fabuloso;grandioso	1	1
	(delicado,**requintado**)	difícil;complicado;delicado	0	1
	(**falido**,queimado)	queimado;incendiado	0	0

Clustering

- Manual evaluation of clusters
- Two judges for each cluster

Clustering

- Manual evaluation of clusters
- Two judges for each cluster
- Cluster is correct if, in some context, all its words might have the same meaning

Clustering

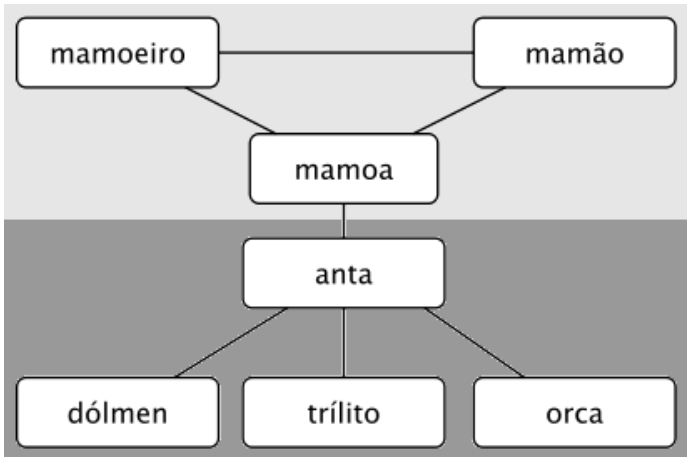
- Manual evaluation of clusters
- Two judges for each cluster
- Cluster is correct if, in some context, all its words might have the same meaning

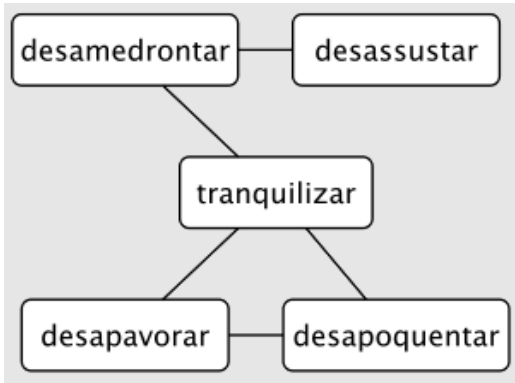
Table: Evaluation of clustering

POS	Sample	Correct		Incorrect		Agreement
Nouns	105 × 2	179	(85.24%)	31	(14.76%)	91.43%
Verbs	105 × 2	193	(91.90%)	17	(8.10%)	87.62%
Adjectives	105 × 2	189	(90.00%)	21	(10.00%)	85.71%

Clustering

- Manual evaluation of clusters
- Two judges for each cluster
- Cluster is correct if, in some context, all its words might have the same meaning





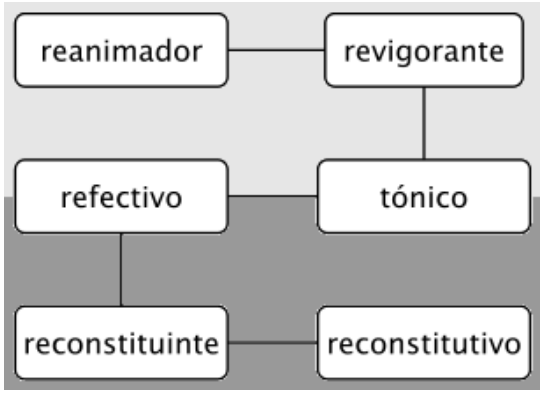


Figure: Examples of connected subgraphs and resulting clusters.

Update: computing similarity

- Sum the adjacencies
  - One vector per synset: $[C_j] = \sum_{k=1}^{|C_j|}[w_k] : w_k \in C_j$;
  - $sim(p, C_j) = sim([p], [C_j])$

Update: computing similarity

- Sum the adjacencies
  - One vector per synset: $[C_j] = \sum_{k=1}^{|C_j|}[w_k] : w_k \in C_j$;
  - $sim(p, C_j) = sim([p], [C_j])$
- Average similarity of the pair with each synset element
  - One vector per synset element: $[C_j] = ([w_1], ..., [w_n]), n = |C_j|$
  - $sim(p, C_j) = \frac{\sum_{k=1}^{|C_j|} \cos([p],[M_{wk}])}{|C_j|}, w_k \in C_j$

Update: computing similarity

- Sum the adjacencies
  - One vector per synset: $[C_j] = \sum_{k=1}^{|C_j|}[w_k] : w_k \in C_j$;
  - $sim(p, C_j) = sim([p], [C_j])$
- Average similarity of the pair with each synset element
  - One vector per synset element: $[C_j] = ([w_1], ..., [w_n]), n = |C_j|$
  - $sim(p, C_j) = \frac{\sum_{k=1}^{|C_j|} \cos([p],[M_{wk}])}{|C_j|}, w_k \in C_j$
- Gold resource of 220 synpairs and possible assignments

Update: computing similarity

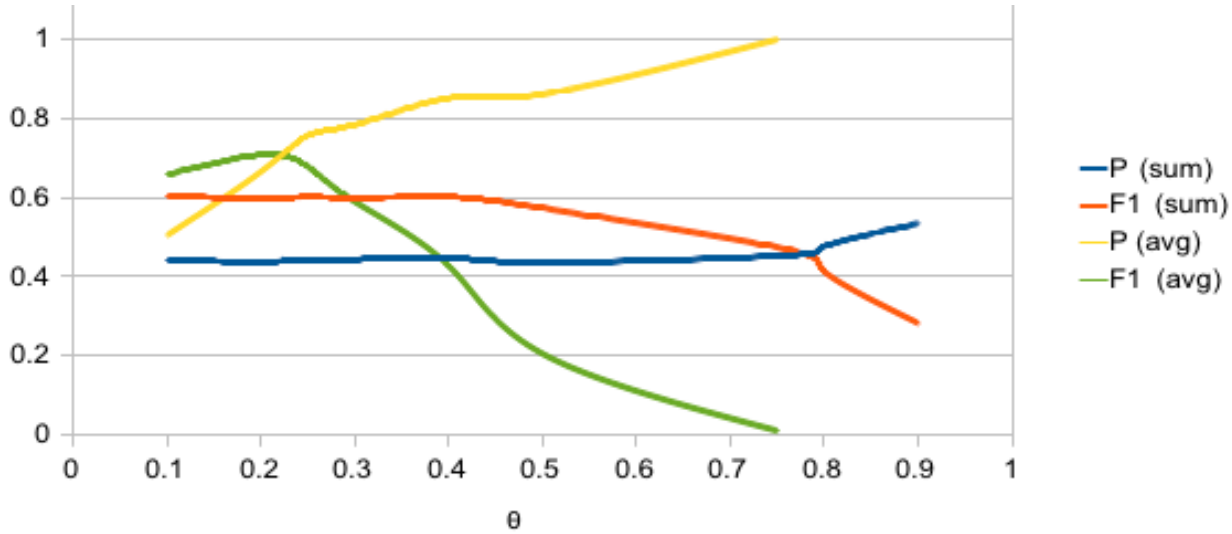
- Sum the adjacencies
  - One vector per synset: $[C_j] = \sum_{k=1}^{|C_j|}[w_k] : w_k \in C_j$;
  - $sim(p, C_j) = sim([p], [C_j])$
- Average similarity of the pair with each synset element
  - One vector per synset element: $[C_j] = ([w_1], ..., [w_n]), n = |C_j|$
  - $sim(p, C_j) = \frac{\sum_{k=1}^{|C_j|} \cos([p],[M_{wk}])}{|C_j|}, w_k \in C_j$
- Gold resource of 220 synpairs and possible assignments
  - Variable cut point θ on similarity

Update: computing similarity

- Sum the adjacencies
  - One vector per synset: $[C_j] = \sum_{k=1}^{|C_j|}[w_k] : w_k \in C_j$;
  - $sim(p, C_j) = sim([p], [C_j])$
- Average similarity of the pair with each synset element
  - One vector per synset element: $[C_j] = ([w_1], ..., [w_n]), n = |C_j|$
  - $sim(p, C_j) = \frac{\sum_{k=1}^{|C_j|} \cos([p],[M_{wk}])}{|C_j|}, w_k \in C_j$
- Gold resource of 220 synpairs and possible assignments
  - Variable cut point θ on similarity
  - Possible to assign the same synpair to $0 \leq n \leq |C|$ synsets

Update: computing similarity

- Sum the adjacencies
  - One vector per synset: $[C_j] = \sum_{k=1}^{|C_j|}[w_k] : w_k \in C_j$;
  - $sim(p, C_j) = sim([p], [C_j])$
- Average similarity of the pair with each synset element
  - One vector per synset element: $[C_j] = ([w_1], ..., [w_n]), n = |C_j|$
  - $sim(p, C_j) = \frac{\sum_{k=1}^{|C_j|} \cos([p],[M_{wk}])}{|C_j|}, w_k \in C_j$
- Gold resource of 220 synpairs and possible assignments
  - Variable cut point θ on similarity
  - Possible to assign the same synpair to $0 \leq n \leq |C|$ synsets



Final remarks

- Flexible method for enriching thesaurus with synonymy in dictionaries

Final remarks

- Flexible method for enriching thesaurus with synonymy in dictionaries
- Applied to the enrichment of a Portuguese thesaurus

Final remarks

- Flexible method for enriching thesaurus with synonymy in dictionaries
- Applied to the enrichment of a Portuguese thesaurus
- This work was made in the scope of **Onto.PT**
  - Automatic creation of a lexical ontology for Portuguese

Final remarks

- Flexible method for enriching thesaurus with synonymy in dictionaries
- Applied to the enrichment of a Portuguese thesaurus
- This work was made in the scope of **Onto.PT**
  - Automatic creation of a lexical ontology for Portuguese
  - Extraction + integration of lexical information from textual sources

Final remarks

- Flexible method for enriching thesaurus with synonymy in dictionaries
- Applied to the enrichment of a Portuguese thesaurus
- This work was made in the scope of **Onto.PT**
  - Automatic creation of a lexical ontology for Portuguese
  - Extraction + integration of lexical information from textual sources
  - Soon freely available!
  - Check `http://ontopt.dei.uc.pt`

Thank you!

References I

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

[Gonçalo Oliveira and Gomes, 2010] Gonçalo Oliveira, H. and Gomes, P. (2010).
Onto.PT: Automatic Construction of a Lexical Ontology for Portuguese.
In *Proc. 5th European Starting AI Researcher Symposium (STAIRS 2010)*. IOS Press.

[Gonçalo Oliveira et al., 2010] Gonçalo Oliveira, H., Santos, D., and Gomes, P. (2010).
Extracção de relações semânticas entre palavras a partir de um dicionário: o PAPEL e sua avaliação.
Linguamática, 2(1):77–93.

[Maziero et al., 2008] Maziero, E. G., Pardo, T. A. S., Felippo, A. D., and Dias-da-Silva, B. C. (2008).
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 da Linguagem Humana (TIL)*, pages 390–392.