Towards the Automatic Creation of a Wordnet from a Term-based Lexical Network

Hugo Gonçalo Oliveira, Paulo Gomes

(hroliv,pgomes)@dei.uc.pt

Cognitive & Media Systems Group
CISUC, University of Coimbra

Uppsala, July 15, 2010

1. Introduction
   - Lexical ontologies
   - Information extraction
   - Issues
   - Research Goals
2. Approach
   - Clustering for synsets
   - Merging thesauri
   - Assigning terms to synsets
3. Experimentation
   - Preparation
   - Wordnet establishment
4. Concluding remarks

Lexical ontologies

- Such as Princeton WordNet [Fellbaum 1998]

Lexical ontologies

- Such as Princeton WordNet [Fellbaum 1998]
  - Ontology + lexicon [Hirst 2004]

Lexical ontologies

- Such as Princeton WordNet [Fellbaum 1998]
  - Ontology + lexicon [Hirst 2004]
  - Knowledge structured on words and their meanings

Lexical ontologies

- Such as Princeton WordNet [Fellbaum 1998]
  - Ontology + lexicon [Hirst 2004]
  - Knowledge structured on words and their meanings
  - Cover the whole language
  - Not based on a specific domain

Lexical ontologies

- Such as Princeton WordNet [Fellbaum 1998]
  - Ontology + lexicon [Hirst 2004]
  - Knowledge structured on words and their meanings
  - Cover the whole language
  - Not based on a specific domain
- Construction and maintenance involve time-consuming human effort!

Information extraction from text

- From dictionaries:

Information extraction from text

- From dictionaries:
  1. **basketball**, noun – *a game, also known as hoops, played indoors...*
     $\rightarrow$ *game* HYPERNYM_OF *basketball*
     $\rightarrow$ *basketball* SYNONYM_OF *hoops*

Information extraction from text

- From dictionaries:
  1. **basketball**, noun – *a game, also known as hoops, played indoors...*
     → *game* HYPERNYM_OF *basketball*
     → *basketball* SYNONYM_OF *hoops*
  2. **basketball**, noun – *the ball used in playing basketball.*
     → *ball* HYPERNYM_OF *basketball*

Information extraction from text

- From dictionaries:
  1. **basketball**, noun – *a game, also known as hoops, played indoors...*
     → *game* HYPERNYM_OF *basketball*
     → *basketball* SYNONYM_OF *hoops*
  2. **basketball**, noun – *the ball used in playing basketball.*
     → *ball* HYPERNYM_OF *basketball*
- From textual corpora:

Information extraction from text

- From dictionaries:
  1. **basketball**, noun – *a game, also known as hoops, played indoors...*
     → *game* HYPERNYM_OF *basketball*
     → *basketball* SYNONYM_OF *hoops*
  2. **basketball**, noun – *the ball used in playing basketball.*
     → *ball* HYPERNYM_OF *basketball*
- From textual corpora:
  - *... team sports, such as basketball, rugby ...*
    → *team_sport* HYPERNYM_OF *basketball*
    → *team_sport* HYPERNYM_OF *rugby*

Natural language is ambiguous

- Term-based networks are impractical for many applications

Natural language is ambiguous

- Term-based networks are impractical for many applications
- In the previous example: is *hoops* a *team sport*?

Natural language is ambiguous

- Term-based networks are impractical for many applications
- In the previous example: is *hoops* a *team sport*?

- An example extracted from a Portuguese dictionary:
  ruína SYNONYM_OF *queda* $\wedge$ *queda* SYNONYM_OF *habilidade*
  $\rightarrow$ *habilidade* SYNONYM_OF *ruína* ??

Natural language is ambiguous

- Term-based networks are impractical for many applications
- In the previous example: is *hoops* a *team sport*?

- An example extracted from a Portuguese dictionary:
  ruína SYNONYM_OF *queda* $\wedge$ *queda* SYNONYM_OF *habilidade* $\rightarrow$ *habilidade* SYNONYM_OF *ruína* ??
- *queda* can either mean *aptitude* or *downfall*!

Onto.PT

- Automatic construction of a lexical ontology for Portuguese

Onto.PT

- Automatic construction of a lexical ontology for Portuguese
- Extracted from different sources

Onto.PT

- Automatic construction of a lexical ontology for Portuguese
- Extracted from different sources
  - Manually created thesauri
  - Language dictionaries/encyclopedias
  - Corpora

Onto.PT

- Automatic construction of a lexical ontology for Portuguese
- Extracted from different sources
  - Manually created thesauri
  - Language dictionaries/encyclopedias
  - Corpora
- Modelled after Princeton WordNet

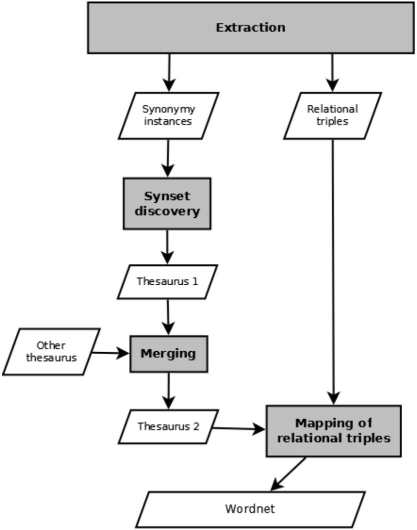
Onto.PT

- Automatic construction of a lexical ontology for Portuguese
- Extracted from different sources
  - Manually created thesauri
  - Language dictionaries/encyclopedias
  - Corpora
- Modelled after Princeton WordNet
  - Synsets: groups of synonymous words
  - Synset-based relational triples

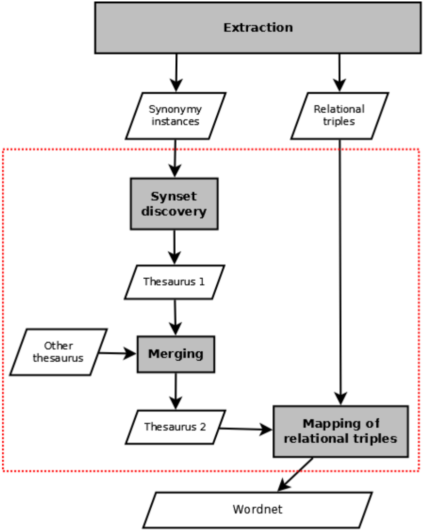
Onto.PT

- Automatic construction of a lexical ontology for Portuguese
- Extracted from different sources
  - Manually created thesauri
  - Language dictionaries/encyclopedias
  - Corpora
- Modelled after Princeton WordNet
  - Synsets: groups of synonymous words
  - Synset-based relational triples
- WSD based on the knowledge already extracted, not on the context.

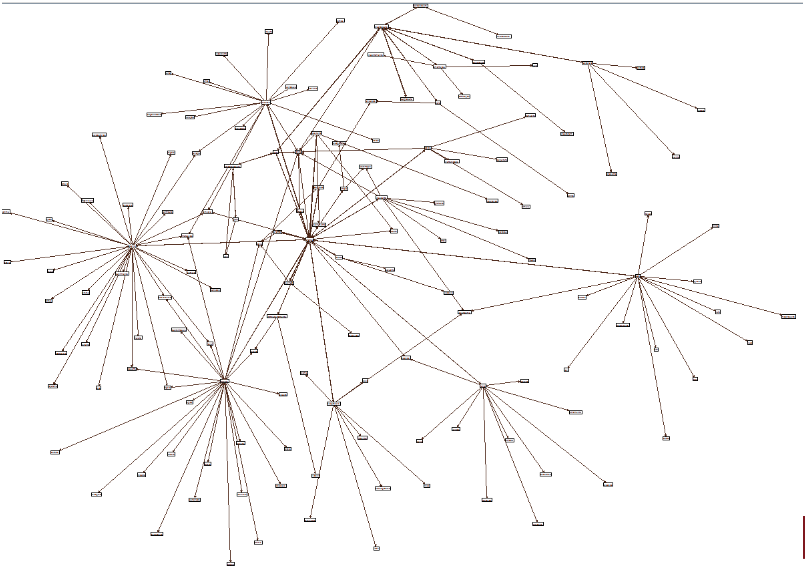
Information flow



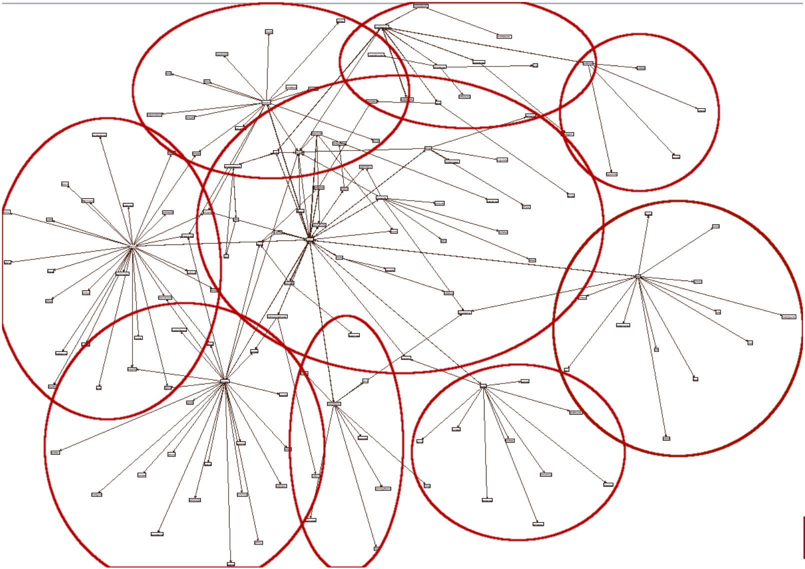
Information flow



Synonymy networks tend to have a clustered structure



Synonymy networks tend to have a clustered structure



Synset discovery (inspired by [Gfeller et al. 2005])

1. Split the original network into sub-networks and calculate the frequency-weighted adjacency matrix F of each sub-network;

Synset discovery (inspired by [Gfeller et al. 2005])

1. Split the original network into sub-networks and calculate the frequency-weighted adjacency matrix F of each sub-network;
2. $F_{ij} = F_{ij} + F_{ij} * \delta$, $-0.5 < \delta < 0.5$;

Synset discovery (inspired by [Gfeller et al. 2005])

1. Split the original network into sub-networks and calculate the frequency-weighted adjacency matrix F of each sub-network;
2. $F_{ij} = F_{ij} + F_{ij} * \delta$, $-0.5 < \delta < 0.5$;
3. Run MCL [van Dongen 2000], with $\gamma = 1.6$, over F for 30 times;

Synset discovery (inspired by [Gfeller et al. 2005])

1. Split the original network into sub-networks and calculate the frequency-weighted adjacency matrix F of each sub-network;
2. $F_{ij} = F_{ij} + F_{ij} * \delta$, $-0.5 < \delta < 0.5$;
3. Run MCL [van Dongen 2000], with $\gamma = 1.6$, over F for 30 times;
4. Use the (hard) clustering from each run to create P, a matrix with the probabilities of each pair of words in F belonging to the same cluster;

Synset discovery (inspired by [Gfeller et al. 2005])

1. Split the original network into sub-networks and calculate the frequency-weighted adjacency matrix F of each sub-network;
2. $F_{ij} = F_{ij} + F_{ij} * \delta$, $-0.5 < \delta < 0.5$;
3. Run MCL [van Dongen 2000], with $\gamma = 1.6$, over F for 30 times;
4. Use the (hard) clustering from each run to create P, a matrix with the probabilities of each pair of words in F belonging to the same cluster;
5. Remove: (a) big clusters, B, if there is a group of clusters $C = C_1, C_2, ... C_n$ such that $B = C_1 \cup C_2 \cup ... \cup C_n$; (b) clusters completely included in other clusters.

Merging synsets from different thesaurus

For each synset $T_i \in T$, select $B_j \in B$ with higher $c = T_i \cap B_j / T_i \cup B_j$[1]

- $B_1 = (diva, beldade, beleza, deidade, deusa, divindade)$
- $B_2 = (divindade, deidade, deus, nume)$

[1] Jaccard coefficient

Merging synsets from different thesaurus

For each synset $T_i \in T$, select $B_j \in B$ with higher $c = T_i \cap B_j / T_i \cup B_j$[1]

- $B_1 = (diva, beldade, beleza, deidade, deusa, divindade)$
- $B_2 = (divindade, deidade, deus, nume)$
- $T_1 = (divindade, diva, deusa)$

[1] Jaccard coefficient

Merging synsets from different thesaurus

For each synset $T_i \in T$, select $B_j \in B$ with higher $c = T_i \cap B_j / T_i \cup B_j$[1]

- $B_1 = (diva, beldade, beleza, deidade, deusa, divindade)$
- $B_2 = (divindade, deidade, deus, nume)$
- $T_1 = (divindade, diva, deusa)$
  - $c(T_1, B_1) = \frac{1}{3}$
  - $c(T_1, B_2) = \frac{1}{6}$

[1] Jaccard coefficient

Merging synsets from different thesaurus

For each synset $T_i \in T$, select $B_j \in B$ with higher $c = T_i \cap B_j / T_i \cup B_j$[1]

- $B_1 = (diva, beldade, beleza, deidade, deusa, divindade)$
- $B_2 = (divindade, deidade, deus, nume)$
- $T_1 = (divindade, diva, deusa)$
  - $c(T_1, B_1) = \frac{1}{3}$
  - $c(T_1, B_2) = \frac{1}{6}$
- $N = B_1 \cup T_1 = (diva, beldade, beleza, deidade, deusa, divindade)$

[1] Jaccard coefficient

Mapping methods

- Input:
  - Thesaurus T, containing synsets
  - Term-based semantic network, N, where each edge has a type R

Mapping methods

- Input:
  - Thesaurus T, containing synsets
  - Term-based semantic network, N, where each edge has a type R
- Goal: map $a\ R\ b \in N$ to $A\ R\ B$, $(A, B) \in T$

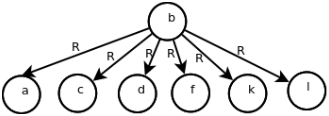
Mapping methods

- Input:
  - Thesaurus T, containing synsets
  - Term-based semantic network, N, where each edge has a type R
- Goal: map $a\ R\ b \in N$ to $A\ R\ B$, $(A, B) \in T$
- Output: semantic network W, whose nodes are synsets, which relate to other synsets by means of semantic relations (wordnet)

Procedure 1

Assignment of a (in $a\ R\ b$) to A:

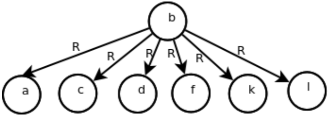
1. Fix b



Procedure 1

Assignment of a (in $a\ R\ b$) to A:

1. Fix b

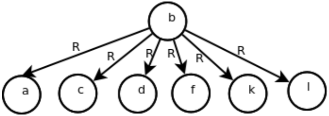


2. $S_a \subset T : S_{ai} \in S_a, a \in S_{ai}$

Procedure 1

Assignment of a (in $a\ R\ b$) to A:

1. Fix b

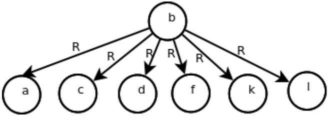


2. $S_a \subset T : S_{ai} \in S_a, a \in S_{ai}$
   - a is not in T ? create synset $A = (a)$, $a \rightarrow A$

Procedure 1

Assignment of a (in $a\ R\ b$) to A:

1. Fix b

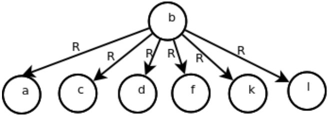


2. $S_a \subset T : S_{ai} \in S_a, a \in S_{ai}$
    - a is not in T ? create synset $A = (a)$, $a \rightarrow A$
3. For each $S_{ai} \in S_a$,

Procedure 1

Assignment of a (in $a\ R\ b$) to A:

1. Fix b

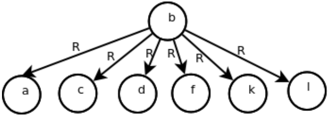


2. $S_a \subset T : S_{ai} \in S_a, a \in S_{ai}$
   - a is not in T ? create synset $A = (a)$, $a \rightarrow A$
3. For each $S_{ai} \in S_a$,
   - $p_{ai} = \frac{n_{ai}}{|S_{ai}|}$, n_{ai} = number of terms $t_j \in S_{ai} : (t_j\ R\ b)$

Procedure 1

Assignment of a (in $a\ R\ b$) to A:

1. Fix b

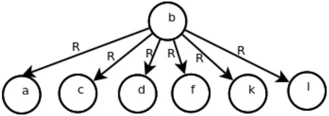


2. $S_a \subset T : S_{ai} \in S_a, a \in S_{ai}$
   - a is not in T ? create synset $A = (a)$, $a \rightarrow A$
3. For each $S_{ai} \in S_a$,
   - $p_{ai} = \frac{n_{ai}}{|S_{ai}|}$, n_{ai} = number of terms $t_j \in S_{ai} : (t_j\ R\ b)$
     - ★ $S_{a1} = (\mathbf{a}, \mathbf{c}, \mathbf{d}, e)$, $p_{a1} = \frac{3}{4}$
     - ★ $S_{a2} = (\mathbf{a}, \mathbf{f}, g)$, $p_{a2} = \frac{2}{3}$
     - ★ $S_{a3} = (\mathbf{a}, h, i, j)$, $p_{a3} = \frac{1}{4}$

Procedure 1

Assignment of a (in $a\ R\ b$) to A:

1. Fix b



2. $S_a \subset T : S_{ai} \in S_a, a \in S_{ai}$
   - a is not in T ? create synset $A = (a)$, $a \rightarrow A$
3. For each $S_{ai} \in S_a$,
   - $p_{ai} = \frac{n_{ai}}{|S_{ai}|}$, n_{ai} = number of terms $t_j \in S_{ai} : (t_j\ R\ b)$
      - ★ $S_{a1} = (\mathbf{a}, \mathbf{c}, \mathbf{d}, e)$, $p_{a1} = \frac{3}{4}$
      - ★ $S_{a2} = (\mathbf{a}, \mathbf{f}, g)$, $p_{a2} = \frac{2}{3}$
      - ★ $S_{a3} = (\mathbf{a}, h, i, j)$, $p_{a3} = \frac{1}{4}$
   - $a \rightarrow S_{a1}$

Procedure 1 (stage 2)

- Only for semi-mapped triples $a\ R\ B$ and $A\ R\ b$

Procedure 1 (stage 2)

- Only for semi-mapped triples $a\ R\ B$ and $A\ R\ b$
- Take advantage of established hypernymy links.

Procedure 1 (stage 2)

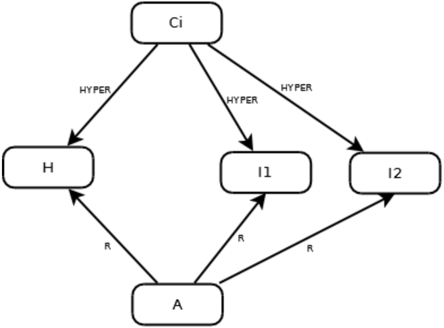
- Only for semi-mapped triples $a\ R\ B$ and $A\ R\ b$
- Take advantage of established hypernymy links.
- Assigning b in $A\ R\ b$

Procedure 1 (stage 2) – examples and additional cleaning

If there is $C_i \in C$ with...

- C_i HYPER_OF $H \wedge A\ R\ H$, $b \rightarrow C_i$

  If all C_i HYPER_OF $I_i \wedge A\ R\ I_i$, triples $A\ R\ I_i$ can be inferred!

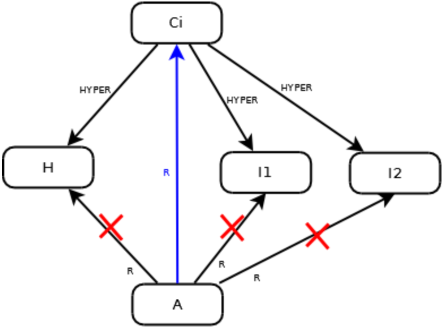


Procedure 1 (stage 2) – examples and additional cleaning

If there is $C_i \in C$ with...

- C_i HYPER_OF $H \wedge A\ R\ H$, $b \rightarrow C_i$

  If all C_i HYPER_OF $I_i \wedge A\ R\ I_i$, triples $A\ R\ I_i$ can be inferred!



- If $H = (dog)$ $I_1 = (cat)$, $I_1 = (mouse)$ and $C_i = (mammal)$:
  - $A = (hair)$ and $R = (PART\_OF)$
  - $A = (animal)$ and $R = (HYPER\_OF)$

Alternative mapping procedure

1. M = term-term matrix based on the adjacencies of the lexical network.

Alternative mapping procedure

1. M = term-term matrix based on the adjacencies of the lexical network.
2. Collect all the synsets with a, $S_a \subset T$, and all synsets with b, $S_b \subset T$.

Alternative mapping procedure

1. M = term-term matrix based on the adjacencies of the lexical network.
2. Collect all the synsets with a, $S_a \subset T$, and all synsets with b, $S_b \subset T$.
3. For each $A \in S_a$ and $B \in S_b$, with terms $A_i \in A$ and $B_j \in B$:

$$sim(A, B) = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} \cos(A_i, B_j)}{|A||B|}$$

Alternative mapping procedure

1. M = term-term matrix based on the adjacencies of the lexical network.
2. Collect all the synsets with a, $S_a \subset T$, and all synsets with b, $S_b \subset T$.
3. For each $A \in S_a$ and $B \in S_b$, with terms $A_i \in A$ and $B_j \in B$:

$$sim(A, B) = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} \cos(A_i, B_j)}{|A||B|}$$

4. Select the pair of synsets with the highest similarity

Resources used (only nouns)

- PAPEL[2] lexical network

[2] http://www.linguateca.pt/PAPEL/
[3] http://www.nilc.icmc.usp.br/tep2/index.htm
[4] http://openthesaurus.caixamagica.pt/

Resources used (only nouns)

- PAPEL[2] lexical network
  - Hypernymy, part-of and member-of triples
  - Synonymy instances

[2] http://www.linguateca.pt/PAPEL/
[3] http://www.nilc.icmc.usp.br/tep2/index.htm
[4] http://openthesaurus.caixamagica.pt/

Resources used (only nouns)

- PAPEL[2] lexical network
  - Hypernymy, part-of and member-of triples
  - Synonymy instances
    - Huge synonymy sub-network with 16k nodes!!!

[2] http://www.linguateca.pt/PAPEL/
[3] http://www.nilc.icmc.usp.br/tep2/index.htm
[4] http://openthesaurus.caixamagica.pt/

Resources used (only nouns)

- PAPEL[2] lexical network
  - Hypernymy, part-of and member-of triples
  - Synonymy instances
    - Huge synonymy sub-network with 16k nodes!!!
- TeP[3] thesaurus
- OpenThesaurus.PT (OT)[4]

- CLIP = clustered PAPEL
- TOP = TeP merged with OT, merged with CLIP

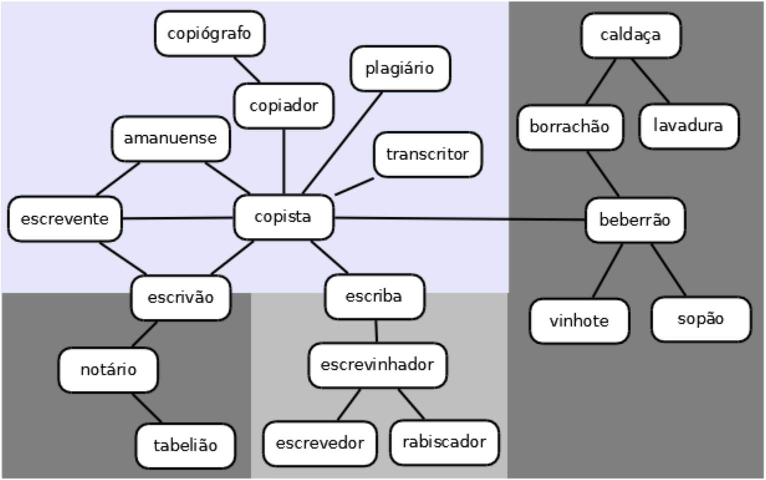
[2] http://www.linguateca.pt/PAPEL/
[3] http://www.nilc.icmc.usp.br/tep2/index.htm
[4] http://openthesaurus.caixamagica.pt/

Resulting Thesaurus

		TeP	OT	CLIP	TOP
Words	**Quantity**	17,158	5,819	23,741	30,554
	Ambiguous	5,867	442	12,196	13,294
	Most ambiguous	20	4	47	21
Synsets	**Quantity**	8,254	1,872	7,468	9,960
	Avg. size	3.51	3.37	12.57	6.6
	Biggest	21	14	103	277

Table: (Noun) thesauruses in numbers.

Clustered sub-network of PAPEL – example



Manual validation

	Sample	Correct	Incorrect	N/A	Agreement
CLIP	519 sets	65.8%	31.7%	2.5%	76.1%
CLIP'	310 sets	81.1%	16.9%	2.0%	84.2%
TOP	480 sets	83.2%	15.8%	1.0%	82.3%
TOP'	448 sets	86.8%	12.3%	0.9%	83.0%

Table: Results of manual synset validation.

- CLIP' and TOP' only consider synsets with 10 or less words.
  - The quality is higher for smaller synsets.

Resulting WordNet

		Hypernym_of	Part_of	Member_of
Term-based triples		62,591	2,805	5,929
1st	**Mapped**	27,750	1,460	3,962
	Same synset	233	5	12
	Already present	3,970	40	167
Semi-mapped triples		7,952	262	357
2nd	**Mapped**	88	1	0
	Could be inferred	50	0	0
	Already present	13	0	0
Synset-based triples		23,572	1,416	3,783

Table: Results of triples mapping

Automatic validation

For each triple, *A R B*

1. Compile a set of textual patterns denoting *R*, e.g.:
   - (hypo) `é um|uma (tipo|forma|variedade|...)* de` (hyper)
   - (whole/group) `é um (grupo|conjunto|...)  de` (part/member)

Automatic validation

For each triple, $A\ R\ B$

1. Compile a set of textual patterns denoting R, e.g.:
   - (hypo) `é um|uma (tipo|forma|variedade|...)* de` (hyper)
   - (whole/group) `é um (grupo|conjunto|...)  de` (part/member)
2. Score the triple with the help of Google:

$$score = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} found(A_i, B_j, R)}{|A| * |B|}$$

Automatic validation

For each triple, $A\ R\ B$

1. Compile a set of textual patterns denoting R, e.g.:
   - (hypo) `é um|uma (tipo|forma|variedade|...)* de` (hyper)
   - (whole/group) `é um (grupo|conjunto|...)  de` (part/member)
2. Score the triple with the help of Google:

$$score = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} found(A_i, B_j, R)}{|A| * |B|}$$

Relation	Sample size	Validation
Hypernymy_of	419 synsets	44,1%
Member_of	379 synsets	24,3%
Part_of	290 synsets	24,8%

Table: Automatic validation of triples

Concluding remarks

- Our way to achieve WSD without a context continues...
  - Clustering is a suitable alternative for establishing synsets

Concluding remarks

- Our way to achieve WSD without a context continues...
  - Clustering is a suitable alternative for establishing synsets
    - What about for networks not extracted from dictionaries?

Concluding remarks

- Our way to achieve WSD without a context continues...
  - Clustering is a suitable alternative for establishing synsets
    - What about for networks not extracted from dictionaries?
  - Rules can be defined to map terms in triples to synsets

Concluding remarks

- Our way to achieve WSD without a context continues...
  - Clustering is a suitable alternative for establishing synsets
    - What about for networks not extracted from dictionaries?
  - Rules can be defined to map terms in triples to synsets
    - Though some triples remain unmapped...

Concluding remarks

- Our way to achieve WSD without a context continues...
  - Clustering is a suitable alternative for establishing synsets
    - What about for networks not extracted from dictionaries?
  - Rules can be defined to map terms in triples to synsets
    - Though some triples remain unmapped...
- Future:
  - Evaluate the alternative mapping method
  - Exploit other resources: e.g. Wiktionary and Wikipedia

References

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

Graeme Hirst (2004).
Ontology and the lexicon.
In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 209–230. Springer.

S. M. van Dongen (2000).
Graph Clustering by Flow Simulation.
Ph.D. thesis, University of Utrecht.

David Gfeller, Jean-Cédric Chappelier and Paulo De Los Rios (2005).
Synonym Dictionary Improvement through Markov Clustering and Clustering Stability.
In *Proc. of International Symposium on Applied Stochastic Models and Data Analysis (ASMDA)*, pages 106–113.

Thank you!