Cross-Lingual WSD Using Multilingual Co-occurrence Graphs

Simone Paolo Ponzetto
Carina Silberer
University of Heidelberg

Outline

1. Cross-lingual WSD
2. CL-WSD using multilingual co-occurrence graphs
3. Conclusions

Outline

1. Cross-lingual WSD:
   - what is it?
   - why do we need it?
   - how to tackle it?

Word Sense Disambiguation

- WSD is the task of computationally determining which sense of a word is activated by its use in a particular context [Ide and Véronis, 1998; Agirre and Edmonds, 2006; Navigli, 2009b]
- basically, a classification task
  - i.e. how to classify words into word senses

bar = {  ,  ,  , ... ,  }

I drank a cup of chocolate at the **bar**

Source: Navigli [2009, RANLP tutorial]

What is a Word Sense?

- a word sense is a commonly-accepted meaning of a word:
  - We are fond of fruit such as **kiwi**$_{\textbf{/fruit}}$ and banana.
  - The **kiwi**$_{\textbf{/bird}}$ is the national bird of New Zealand.
- how to represent word senses?
  - can we enumerate the senses of a word?

{  ,  ,  , … ? }

"Kiwi is my mother tongue, but I also speak all other English languages"

Source: Navigli [2009, RANLP tutorial]

Typical WSD Framework

kiwi
(target word)

"we are fond of fruit such as **kiwi** and banana"

(context)

WSD system

Source: Navigli [2009, RANLP tutorial]

Lexical sample vs. All words

- Lexical sample
  - disambiguate **a restricted set of words**:
    - We are fond of fruit such as **kiwi** and banana.
    - The **kiwi** is the national bird of New Zealand.
    - ... a perspective from a native **kiwi** speaker (NZ).
- All words
  - disambiguate **all content words** in a sentence:
    - The **kiwi is** the **national bird** of **New Zealand**.
    - ... a **perspective** from a **native kiwi speaker** (NZ).

Why WSD?

- examples of AI/NLP applications which can benefit from word senses:
  - Information Retrieval
  - Information Extraction
  - Machine Translation

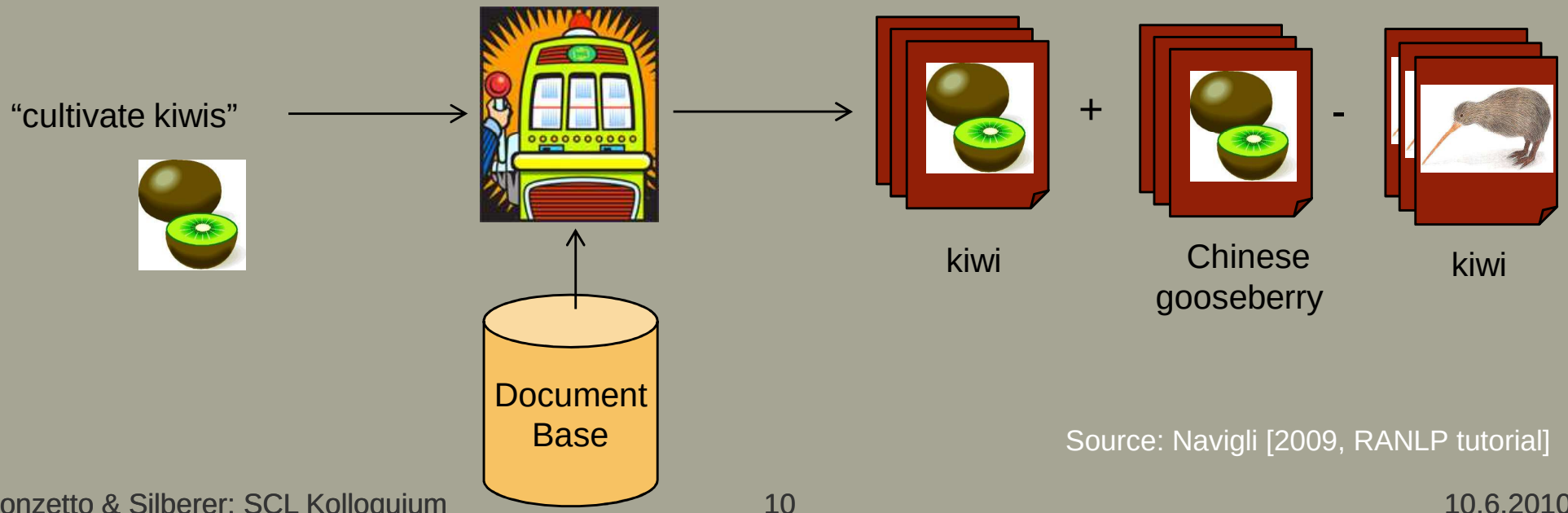
Applications: Information Extraction

- distinguishing between specific instances of concepts
  - Bioinformatics: solve ambiguities in naming genes and proteins
  - acronym expansion (MP: member of parliament or military police?)
  - metonymy recognition (BMW: the company or the car?)
    - Semeval-2007 Metonymy Recognition task [Markert and Nissim, 2007]
  - disambiguate people names
    - Semeval-2007 Web People Search task [Artiles et al. 2007]

Source: Navigli [2009, RANLP tutorial]

Applications: Information Retrieval

- state-of-the-art search engines do not use explicit semantics
- WSD could be used for two purposes:
  - discarding documents which contain a query word w used in a different sense (**polysemy** problem)
  - retrieving documents which contain a word w' which is synonymous of w (**synonymy** problem)



Source: Navigli [2009, RANLP tutorial]

Applications: Machine Translation

- choose better translation candidates
  - **English**: We ate a **kiwi**
  - **German**: Wir haben eine **Kiwi** gegessen
    - (instead of e.g. **Neuseeländer**)
- contrasting evidence that ‘classic’ WSD can benefit MT …
- … BUT if seen as **a model for the selection of the most likely translation** (i.e. integrated into the MT procedure), **it has been shown to improve MT** [Carpuat and Wu, 2007; Chan et al., 2007]

CROSS-Lingual WSD

- disambiguates a target word by labeling it with the appropriate translation
  - the plausible translations of a word in context restrict its possible senses to a subset
- does not necessarily perform full disambiguation (words might remain ambiguous from language to language)
  - cf. interest (in English), interesse (in Italian), intérêt (in French)

CROSS-Lingual WSD

- disambiguates a target word by labeling it with the appropriate translation in multiple languages
- Input: an English (i.e. *source language*) sentence
  - “I’ll buy a train or **coach** ticket”.
- Output: **translations in five** (i.e. *target*) **languages**
  - DE: Bus (3); Linienbus (2); Omnibus (2); Reisebus (2);
  - NL: autobus (3); bus (3); busvervoer (1); toerbus (1);
  - IT: autobus (3); corriera (2); pullman (2); pulmino (1);
  - FR: autobus (2); autocar (1); bus (3); car (3);
  - ES: autobús (3); autocar (3);

CROSS-Lingual WSD

! different contexts can trigger different translations

- Input: Agassi's coach came to me with the rackets.
- Output:
  - DE: Coach (2); Trainer (3);
  - NL: coach (3); speler-trainer (1); trainer (3);
  - IT: allenatore (3);
  - FR: capitaine (1); entraîneur (3);
  - ES: entrenador (3);

Take home message 1

- contrasting results about the benefits of WSD in NLP applications:
  - i.e. no clear benefits have been shown in end-to-end applications such as e.g. (semantic) IR
- WSD improves MT when viewing translations as senses
- we can formulate WSD as a translation task

WSD AND MT ARE BENEFICIAL TO EACH OTHER

Main WSd Approaches

- Supervised WSD
  - Formulates the disambiguation problem as a **supervised classification task**
  - Requires sense-tagged training sets (e.g. SemCor)
- Knowledge-based WSD
  - Uses **knowledge resources** to identify word senses in context
  - Weak supervision (i.e. no training phase)
- Unsupervised WSD (aka Word Sense Discrimination/Induction)
  - Does not need manually-tagged datasets
  - Non-fixed sense inventory makes the task **more difficult to evaluate**

Main WSd Approaches

- Supervised WSD
  - + most successful approach
  - − relies on training data; limited portability across domains
- Knowledge-based WSD
  - + more promising on the short-medium term
  - − needs wide-coverage knowledge resources
- Unsupervised WSD (aka Word Sense Discrimination/Induction)
  - + no need manually-tagged datasets
  - + produces sense clusters as output

Take home message 2

- given (a) lack of annotated data; (b) lack of a wide-coverage multilingual knowledge resource

  UNSUPERVISED METHODS ARE THE MOST PROMISING TO PERFORM CL-WSD

- our proposal
  - **build multilingual co-occurrence graphs** from automatically aligned text
  - apply graph-based algorithms, e.g. Hyperlex [Véronis, 2004] and PageRank [Brin and Page, 1998] to **discriminate and assign word senses across languages**

Outline

1. Cross-lingual WSD
2. CL-WSD using multilingual co-occurrence graphs
   - **A. Methods**
     - **a) Multilingual graph construction**
     - **b) Finding root hubs**
     - **c) Multilingual disambiguation**
   - B. Experiments
     - a) Task setup
     - b) Evaluation

Methodology

CL-WSD using multilingual co-occurrence graphs

- in a nutshell, apply the **method from** [Véronis, 2004] and [Agirre et al., 2006] to a structured – i.e. **graph-based** – representation of **multilingual context**

Methodology

1. build for each target word a **multilingual co-occurrence graph** based on the target word's aligned contexts found in parallel corpora

2. use an adapted PageRank algorithm [Agirre et al., 2006] to **select the nodes (*hubs*) which represent the target word's different senses**

3. **compute the Minimum Spanning Tree** (MST), which is used to select the most relevant words for each word sense/usage

4. use the MST to **disambiguate a given test instance in context**

HyperLex [Véronis, 2004]

based on the “small-world assumption”

1) given a target word w, build a co-occurrence graph
   - an edge is added between w_i and w_j if w_i cooccurs with w_j at least 5 times in a corpus

2) the weight of an edge $\{w_i, w_j\}$ is given by:

where:

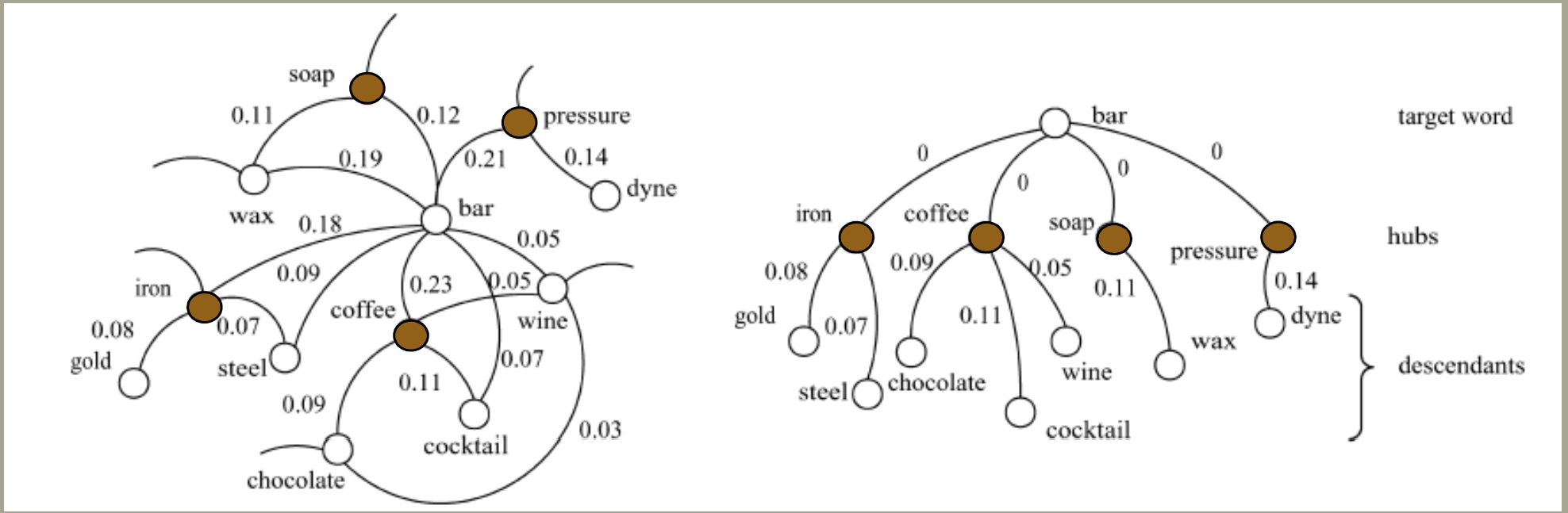
$$P(w_i \mid w_j) = \frac{count(w_i, w_j)}{count(w_j)}$$

3) edges with a weight ≥ threshold are removed

Hub selection and MST

4) select hubs, that is nodes which “represent” senses of the target word:
   a) select as hub the node h with the highest degree in the graph
   b) the neighbors of h are no more eligible as hubs
   c) if highest degree ≤ threshold, go to step 5
   d) otherwise, go to step a
5) connect all hubs to the target word w with weight 0
6) calculate the minimum spanning tree (MST) of the graph

HyperLex: Example



Inital cooccurrence graph

Minimum Spanning Tree

Source: Navigli [2009, RANLP tutorial]

Hub selection with PageRank [Agirre et al., 2006]

- an option is to perform **hub selection with PageRank** rather than by iteratively selecting nodes with highest degree

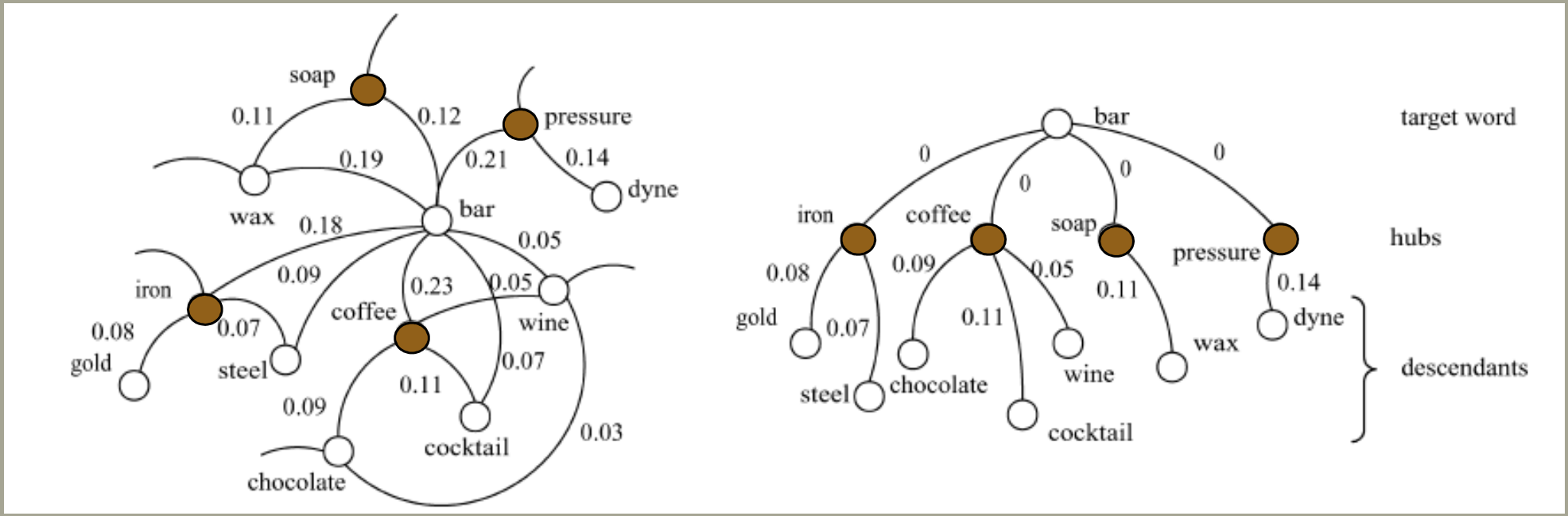
- the initial PageRank of all vertices is set to 1/N, where N is the number of vertices (i.e. N = |V|)
- PageRank is applied
- the top ranking nodes are selected as hubs

Hyperlex: Disambiguation

- each node in the MST is assigned a **score vector** with *as many dimensions as there are components*

- **Step 1**:
  - for a given context, add the score vectors of all words in that context.
- **Step 2**:
  - select the component that receives the highest weight.

Hyperlex: Disambiguation



- example: “I drank a cup of chocolate at the **bar**”
  - **step 1: (0, 0.09, 0, 0)**
  - **step 2: “bar” as [chocolate, wine, cocktail]**

Source: Navigli [2009, RANLP tutorial]

Methodology

- we apply the Hyperlex/PageRank approach in a multilingual setting

1. monolingual graph construction
2. multilingual graph extension
3. computing root hubs and MST
4. multilingual disambiguation

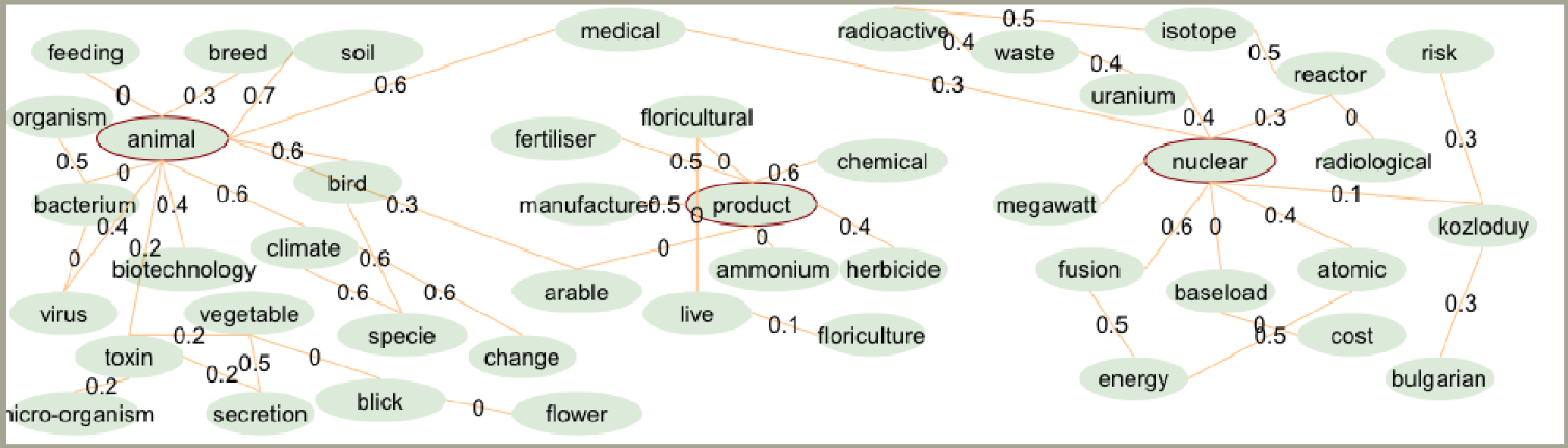
Monolingual Graph

- given (a) a target word *w* in a source language *s*, and (b) all contexts of *w* in a corpus we first construct a monolingual graph

  - we collect all **pairs of co-occurring nouns or adjectives in** (excluding the target word itself) and **add each word as a node** into the initially empty graph
  - each co-occurring word pair is connected with an edge
  - $(v_i, v_j) \in E_s$ which is assigned a "dissimilarity" weight:

$$w(v_i, v_j) = 1 - \max\left[p(cw_i|cw_j), p(cw_j|cw_i)\right]$$

Monolingual Graph



example: monolingual graph for *plant* (excerpt)

Multilingual graph

- given a set of *target* languages *L*, we extend G_s to a labeled multilingual graph $G_{ML} = \langle V_{ML}, E_{ML} \rangle$
  - $V_{ML} = V_s \cup \bigcup_{l \in L} V_l$ is a set of nodes representing content
    words from either the source or the target languages
  - $E_{ML} = E_s \cup \bigcup_{l \in L} \{E_l \cup E_{s,l}\}$ is a set of edges which include:
    - **co-occurrence edges** in the target language
    - **labeled translation edges**

Co-occurrence edges

- for each target language $l \in L$ in turn
- given a parallel corpus of sentences *word-aligned* with the sentences in
- build **co-occurrence edges** $E_l \subseteq V_l \times V_l$ between nodes representing words in a target language (V_l), **weighted *in the same way* as the edges in the monolingual graph**
  - e.g. “Tier” co-occurs with “Biotechnologie” in German

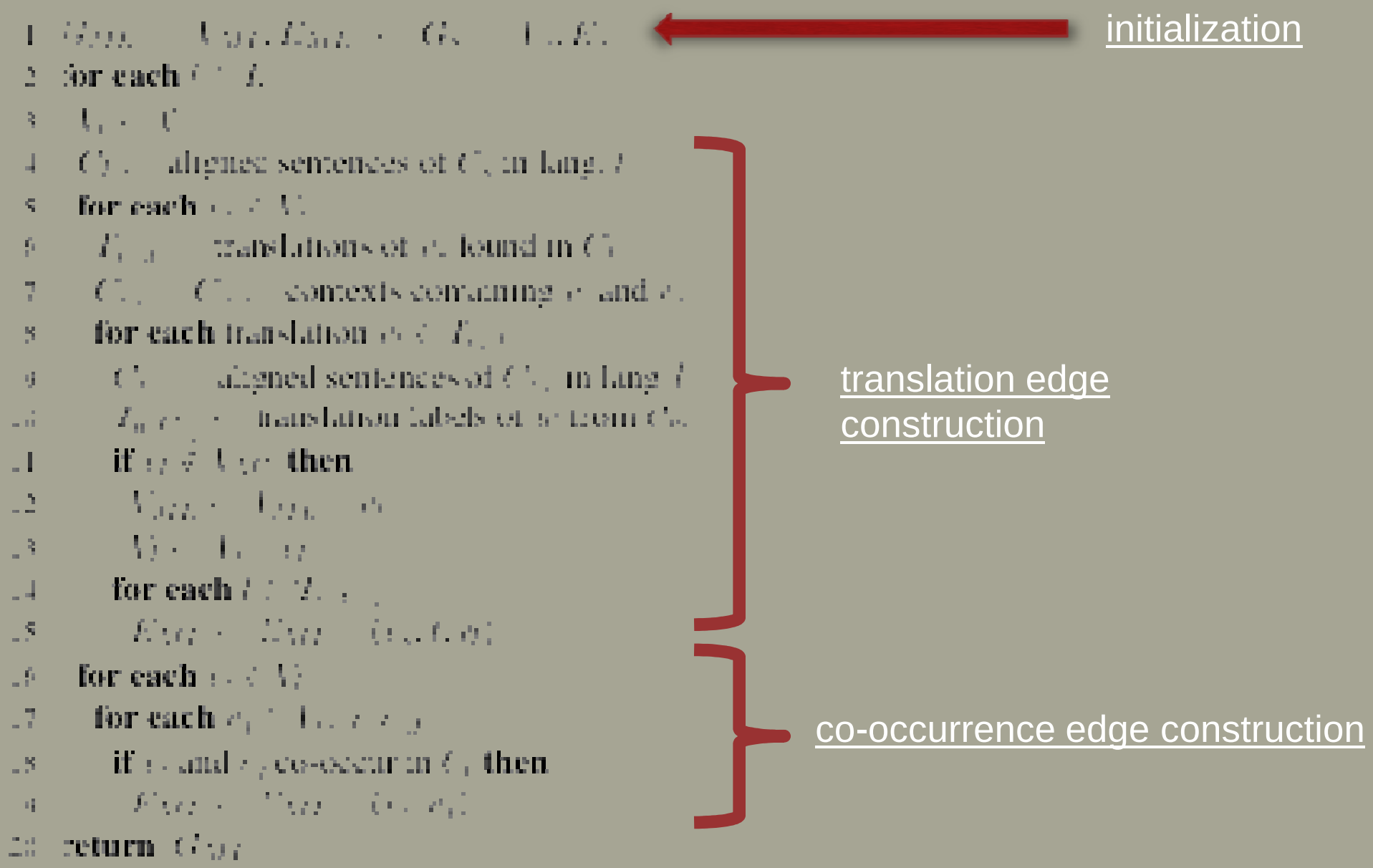
Translation Edges

- translation edges $E_{s,l}$ represent translations of words from the source language *s* into a target language *l*
  - add the translation edges $(v_s, t, v_l) \in E_{s,l}$ of each word in the source language
- in order to include the information about the translations of the target word *w* in the different languages
  - each translation edge receives a **translation label** *t*

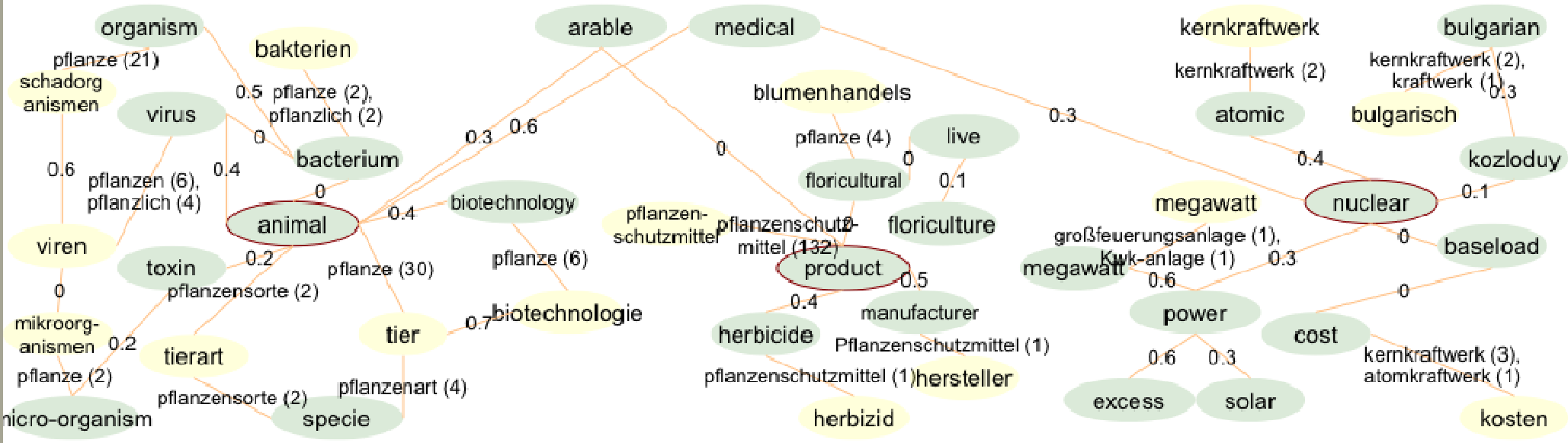
Translation labels

- in a nutshell, a label to capture the **translation of the target word in the aligned context** where the translation of a word in the source language is found
- given:
  - $C_{v,s} \subseteq C_s$: the contexts where v_s and w co-occur,
  - $C_{v,l}$: the word-aligned contexts in language l of $C_{v,s}$, where v_s is translated as v_l
- label the edge between v_s and v_l with
  - (a) **translation** of w in $C_{v,l}$
  - (b) **frequency** of the translation
  - (c) whether the translation is **monosemous** as found in EuroWordNet or PanDictionary [Mausam et al., 2009]

Bringing it all together …



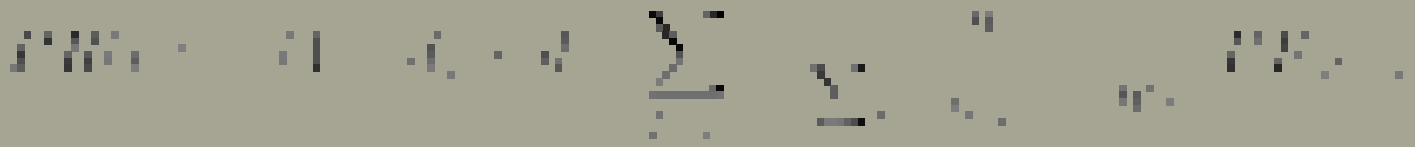
Multilingual graph



example: multilingual graph for *plant* (excerpt)

Computing root hubs

- we use the adapted PageRank from [Agirre et al. 2006]

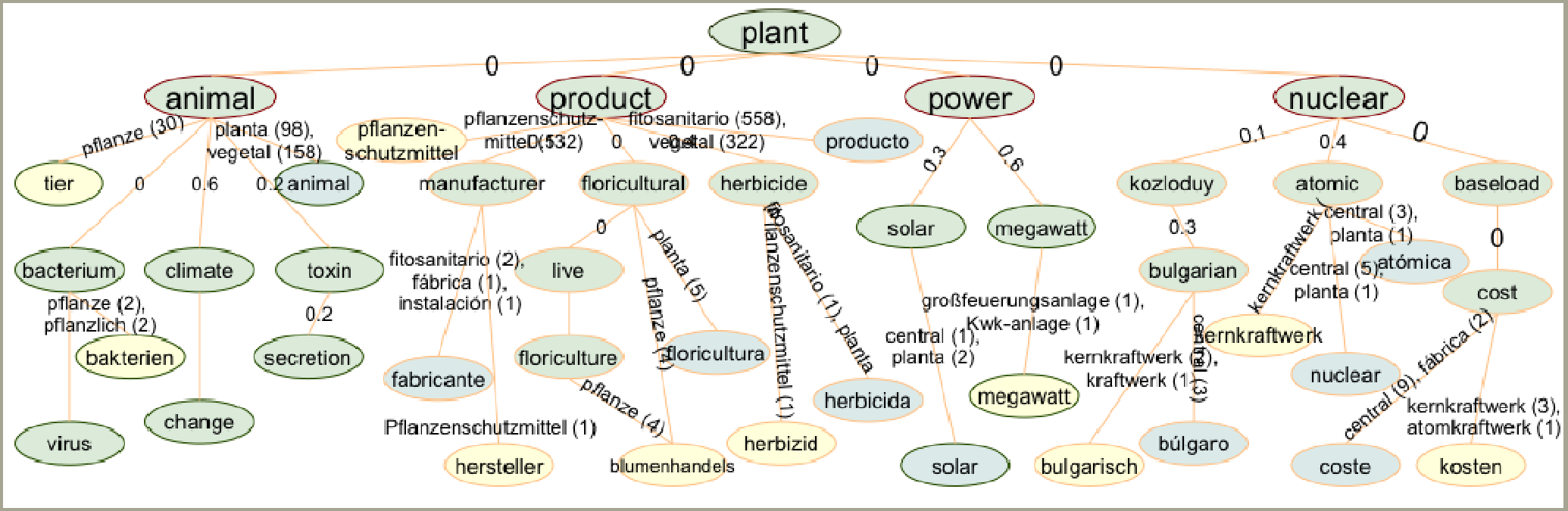


- only nodes referring to English words can be identified as hubs…
- but we also include information from other languages:
  - **include co-occurrence edges from other languages** in the PR computation if the *respective translation edges* are **labeled with monosemous translations**

Computing the MST

- following [Véronis, 2004], a MST is built – e.g. we can use Kruskal's algorithm – with the target word as its root and the root hubs of G_{ML} forming its first level
- by using a multilingual graph, we are able to obtain **MSTs which contain translation nodes and edges**

Multilingual MST



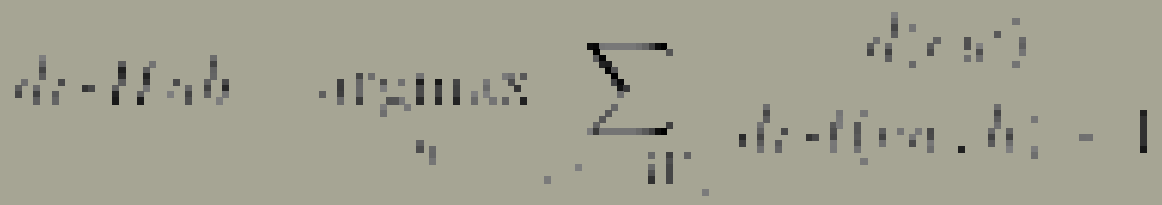
example: MST for *plant* (excerpt)

Multilingual WSD

- given a context $W = \{cw_1 \ldots cw_n\}$ for the target word *w* in the source language, **use the MST to find the most relevant words in *W* for disambiguating *w***

Multilingual WSD

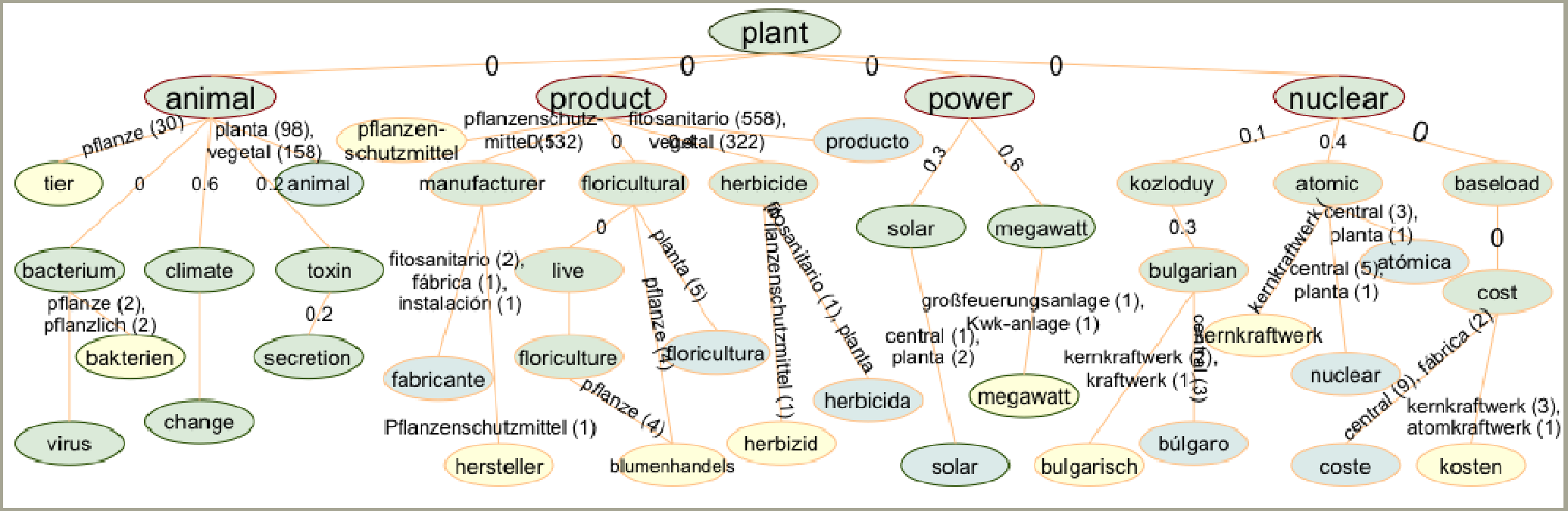
- **find the correct hub** (i.e. sense) as



  - *d(cw)*: distance in words between *cw* and *w*
  - *dist(cw,h)*: number of edges between *cw* and *h* in the MST

- **retain only those context nodes linked to** *disHub*
- **collect the translations of the target word along the translation edges** (use the counts to rank them)

Multilingual MST



example: “a virus which attack animals and plants”

Outline

1. Cross-lingual WSD
2. CL-WSD using multilingual co-occurrence graphs
   A. Methods
      a) Multilingual graph construction
      b) Finding root hubs
      c) Multilingual disambiguation
   B. Experiments
      a) Task setup
      b) Evaluation

Experimental setup

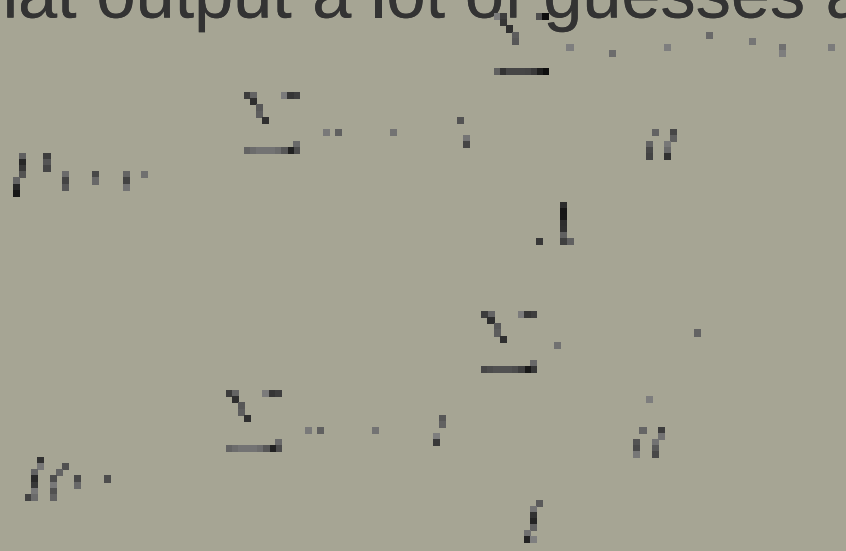
- dev set : 5 words / 20 sentences per word
- test set : 20 words / 50 sentences per word
- for each language, each sentence is annotated with **at most three translations from the same multilingual cluster**
- **“**Strangely, the national **coach** of the Irish teams down the years has had little direct contact with the four provincial coaches.**”**
  - **DE: Nationaltrainer (2)/ Trainer (3) / Coach (1)**
  - **NL: trainer (3) / coach (3) / voetbaltrainer (1)**

Evaluation

- Evaluation scheme inspired by the English lexical substitution task in SemEval 2007 [McCarthy and Navigli, 2007]
- (**standard**) Precision / Recall
- **Mode** Precision / Recall (computed on the most preferred translation)
- **Best result** vs. **Out-of-five** evaluation

Best result evaluation

- systems can propose as many guesses as the system believes are correct
- the resulting score is divided by the number of guesses
- (systems that output a lot of guesses are not favored)



OOF Evaluation

- systems can propose **up to five guesses**
- the resulting score is not divided by the number of guesses
- (more relaxed evaluation)

Baselines

- **best result evaluation**: select the most frequent lemmatized translation from the GIZA++ alignments
- **out-of-five evaluation**: select the five most frequent translations from the GIZA++ alignments.

UHD Experimental setup

- submitted two runs (UHD-1/UHD-2)
- **UHD-1** uses Europarl [Koehn, 2005] …
- incremented with JRC-Aquis [Steinberger et al., 2006] for **UHD-2**
- PoS-tagging: TreeTagger [Schmid, 1994]
- word-alignments: GIZA++ [Och and Ney, 2003]
- morphological analysis for German: Morphisto [Zielinski et al., 2009]

Results: UHD

- **Best** result evaluation

Language	P	R	Mode P	Mode R
FRENCH	20.22	16.21	17.59	14.56
GERMAN	12.20	9.32	11.05	7.78
ITALIAN	15.94	12.78	12.84	8.48
SPANISH	20.48	16.33	28.48	22.19

- **OOF** evaluation

Language	P	R	Mode P	Mode R
FRENCH	30.06	32.00	37.08	26.79
GERMAN	27.62	22.82	25.68	21.16
ITALIAN	33.72	27.49	27.54	21.81
SPANISH	38.78	31.81	40.68	32.35

Results: Best overall

	Prec	Rec	M_F	M_R
Spanish				
UvT-v	33.12	21.98	21.98	21.98
UvT-g	19.92	19.92	21.17	21.17
I8-COLEUR	19.78	19.59	21.59	21.79
VUH-1	30.18	16.43	28.18	22.19
VUH-2	20.2	16.09	28.28	22.46
FCC-WSD1	15.09	15.09	21.31	21.31
FCC-WSD3	11.13	11.13	23.11	23.11
French				
I8-COLEUR	21.96	21.73	26.25	25.34
VUH-2	30.93	16.67	27.78	24.15
VUH-1	30.22	16.21	27.59	24.56
OWNS2	16.05	16.07	24.21	24.21
OWNS1	16.05	16.07	24.21	24.21
OWNS3	12.53	12.53	24.21	24.21
OWNS4	10.19	10.19	24.21	24.21

Italian				
I8-COLEUR	15.55	15.4	16.2	16.12
VUH-2	16.28	13.63	24.59	9.46
VUH-1	15.91	12.78	23.24	8.48
Dutch				
UvT-v	17.7	17.7	23.15	23.05
UvT-g	17.63	17.43	26.74	26.74
I8-COLEUR	10.72	10.56	8.48	6.26
German				
I8-COLEUR	13.79	13.63	8.2	8.1
VUH-1	12.2	9.32	24.15	7.78
VUH-2	12.03	9.23	23.24	9.23

Results: OOF overall

	Prec	Rec	M_F	M_A
Spanish				
UWL-g	13.13	13.13	13.91	13.91
UWL-c	13.17	13.17	10.62	10.62
HCC-WSD2	10.70	10.70	11.81	11.81
HCC-WSD1	38.10	38.10	39.19	39.19
TH-COLLECTR	35.81	35.10	39.01	38.78
UBC-1	38.78	31.81	10.48	32.48
UBC-2	37.71	31.3	39.09	32.05
French				
TH-COLLECTR	19.11	18.90	12.14	12.77
OWNS1	13.21	13.21	38.29	38.29
OWNS2	38.71	38.71	37.74	37.74
UBC-1	29.10	32	37.08	26.79
UBC-2	37.92	31.38	37.46	27.08
Italian				
TH-COLLECTR	10.7	10.41	38.99	38.76
UBC-1	38.72	27.19	27.51	21.81
UBC-2	32.68	27.12	29.82	24.26
Dutch				
UWL-c	11.97	11.97	21.62	21.62
UWL-g	11.92	11.92	20.72	20.72
TH-COLLECTR	21.17	21.27	23.05	23.04
German				
TH-COLLECTR	33.21	32.82	33.00	31.76
UBC-1	27.62	22.82	25.68	21.16
UBC-2	27.21	22.57	27.29	22.36

Discussion

- BEST: evaluation we **rank in the middle** for those languages where the majority of systems participated
  - i.e. second and fourth out of 7 submissions for FRENCH and SPANISH
- compared against the baseline in the BEST evaluation
  - higher **precision** for ITALIAN and SPANISH (+1.9% and +2.1%, respectively); FRENCH and GERMAN lie near below the baseline scores (−0.5% and −1.0%, respectively)
  - trade-off is a **recall** always below the baseline

Discussion

- compared against the baseline in the BEST evaluation
  - higher **precision** for ITALIAN and SPANISH (+1.9% and +2.1%)
  - FRENCH and GERMAN lie near below the baseline scores (−0.5% and −1.0%)
  - trade-off is a **recall** always below the baseline
  - we beat the Mode precision baseline for all languages, i.e. up to +5.1% for SPANISH

⇨ our system is strongly precision-oriented

- (cf. low performance in the OOF evaluation)

Conclusions

- **outcome:** a graph-based approach to perform CL-WSD
- **results:** a upper/middle-tier ranking system showing the feasibility of approaching CL-WSD from a graph-based perspective
- **limitations:** a limited performance – shared with all task participants.

Future work

- building a unique graph with all languages at the same time could introduce noise …
  - ➪ possible extension/modification of the algorithm, e.g. **computing hubs in each language independently** and combining them as a **joint problem**
- most frequent translations tend to receive too much weight and accordingly crowd out more appropriate translations
  - ✸ developing robust techniques for unsupervised tuning of the graph weights

Future work

- investigate the application of our approach *directly* to multilingual lexical resources

  Examples:

  - PanDictionary [Mausam et al., 2009]
  - ??? ← your resource here?
  - … stay tuned for our ACL dry-run ☺

Questions?

YOUR TURN! ☺

- Acknowledgements: SCL for infrastructure and inspirational environment
- check out our papers (out from beginning of July) @

http://www.cl.uni-heidelberg.de/~ponzetto