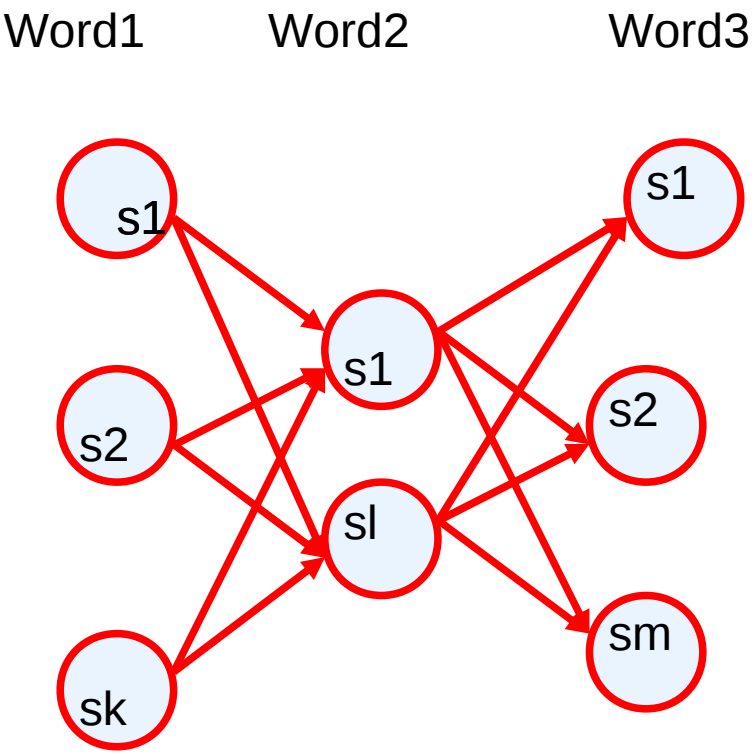
Lesk

Use viterbi



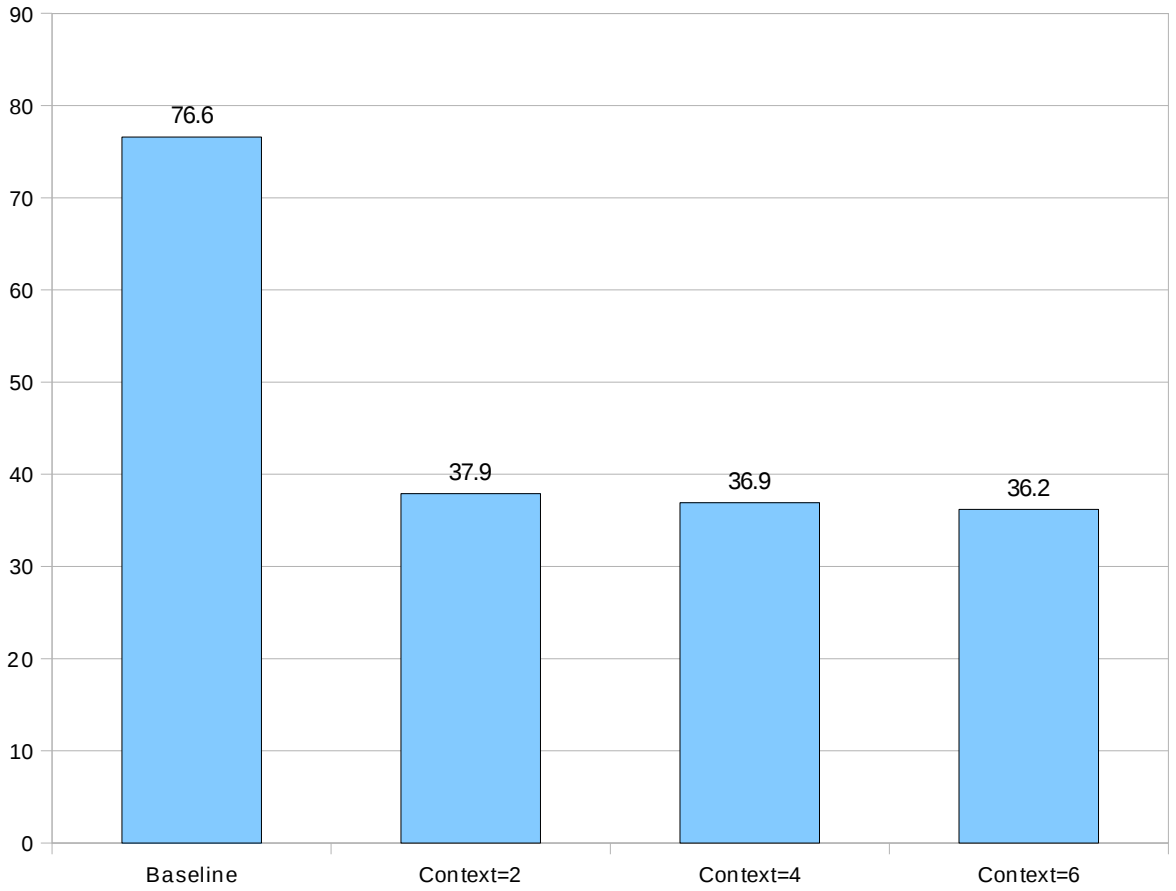
Random Walk

- Create a multi-stage graph with sense nodes
- #stages = #Cws
- Nodes at each stage correspond to all possible senses of a CW
- Add edges to graph with wts as the overlap between sense glosses
- Calculate scores of nodes with Random Walk algorithm
- At each stage select node with highest wt.

Some issues

- What to do in case of same word overlapping multiple times.
- Handling of stop words
- Special stop words added like <t>, <s>
- Use of WordNet stemmer
- Handling all possible stemmed versions

Conceptual Density



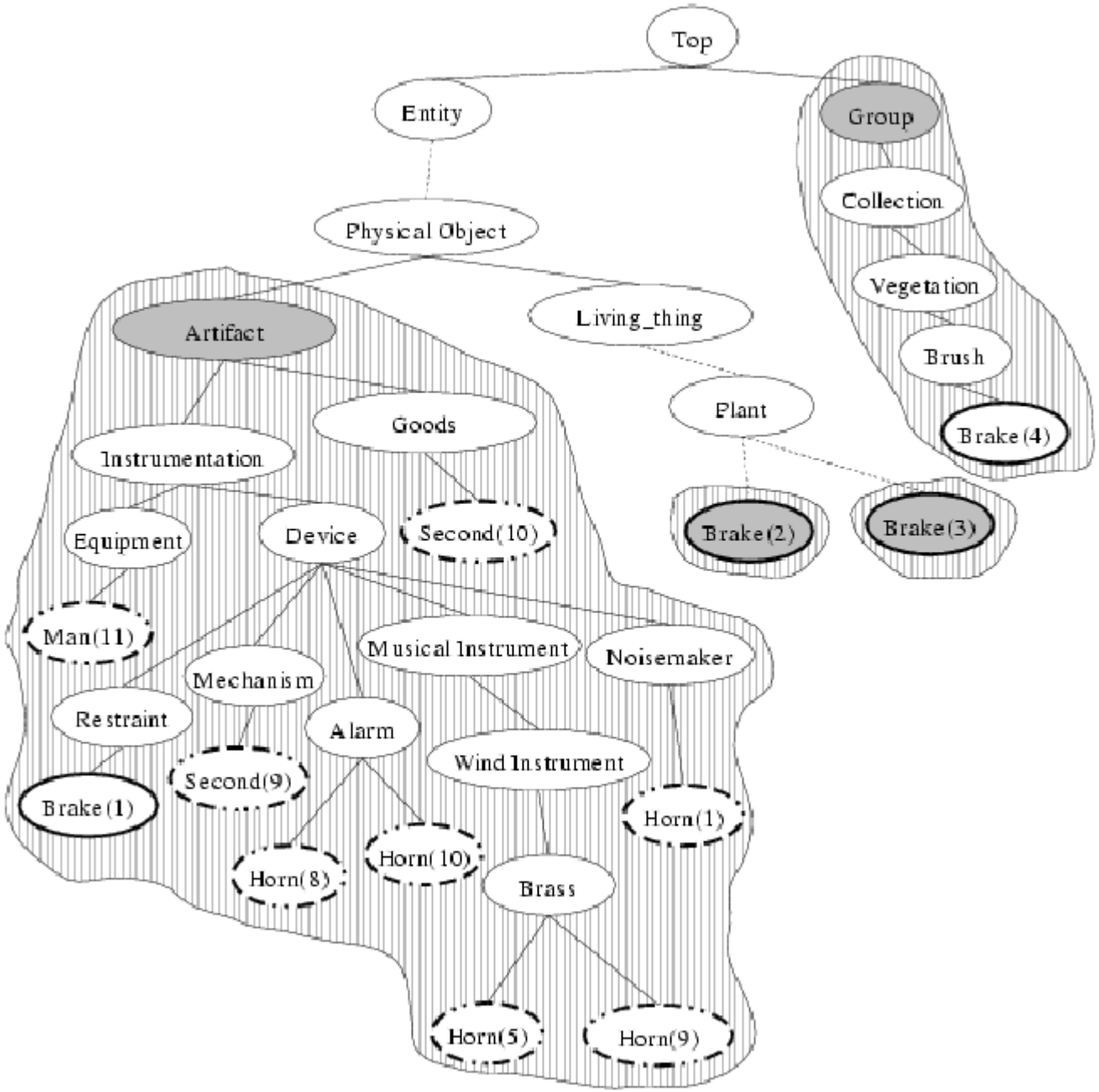
Implementation of knowledge-based WSD algorithms

by
Anup Kulkarni, Prashanth Kamle
& Saurabh Sohoney

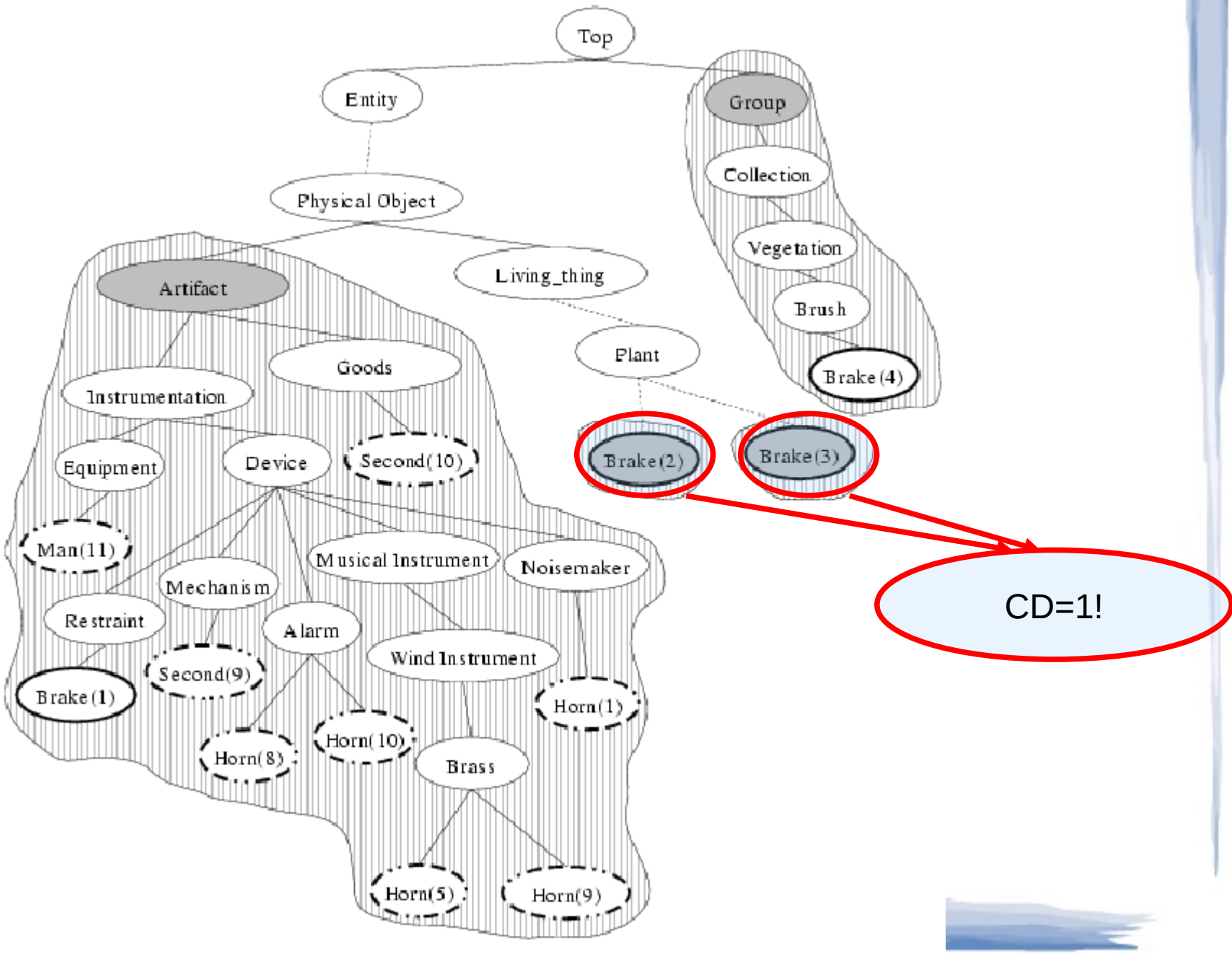
Where it screws up...

"Brakes howled and a horn blared furiously, but the man would have
been hit if Phil hadn't called out to him a second before"

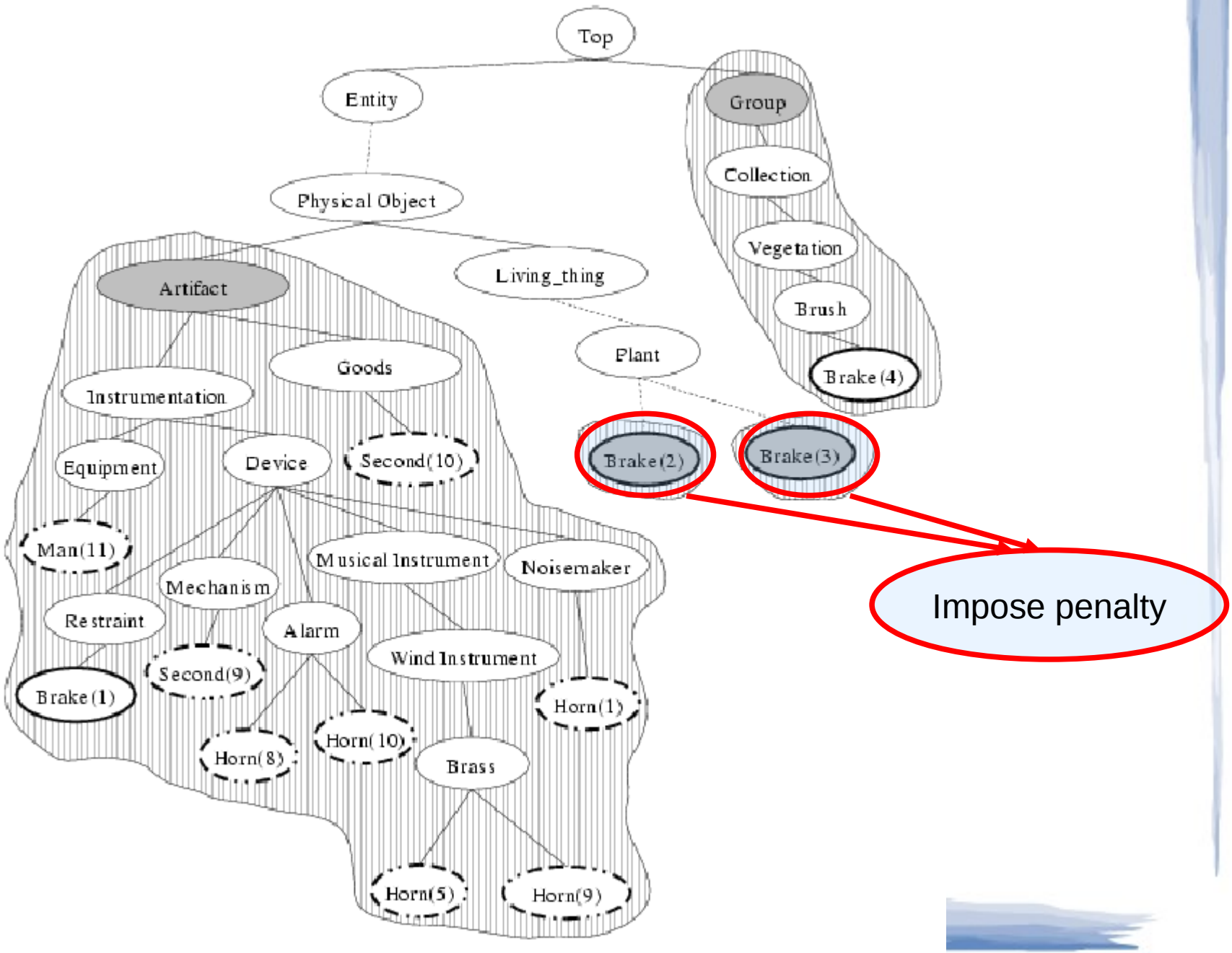
Where it screws up...



Where it screws up...



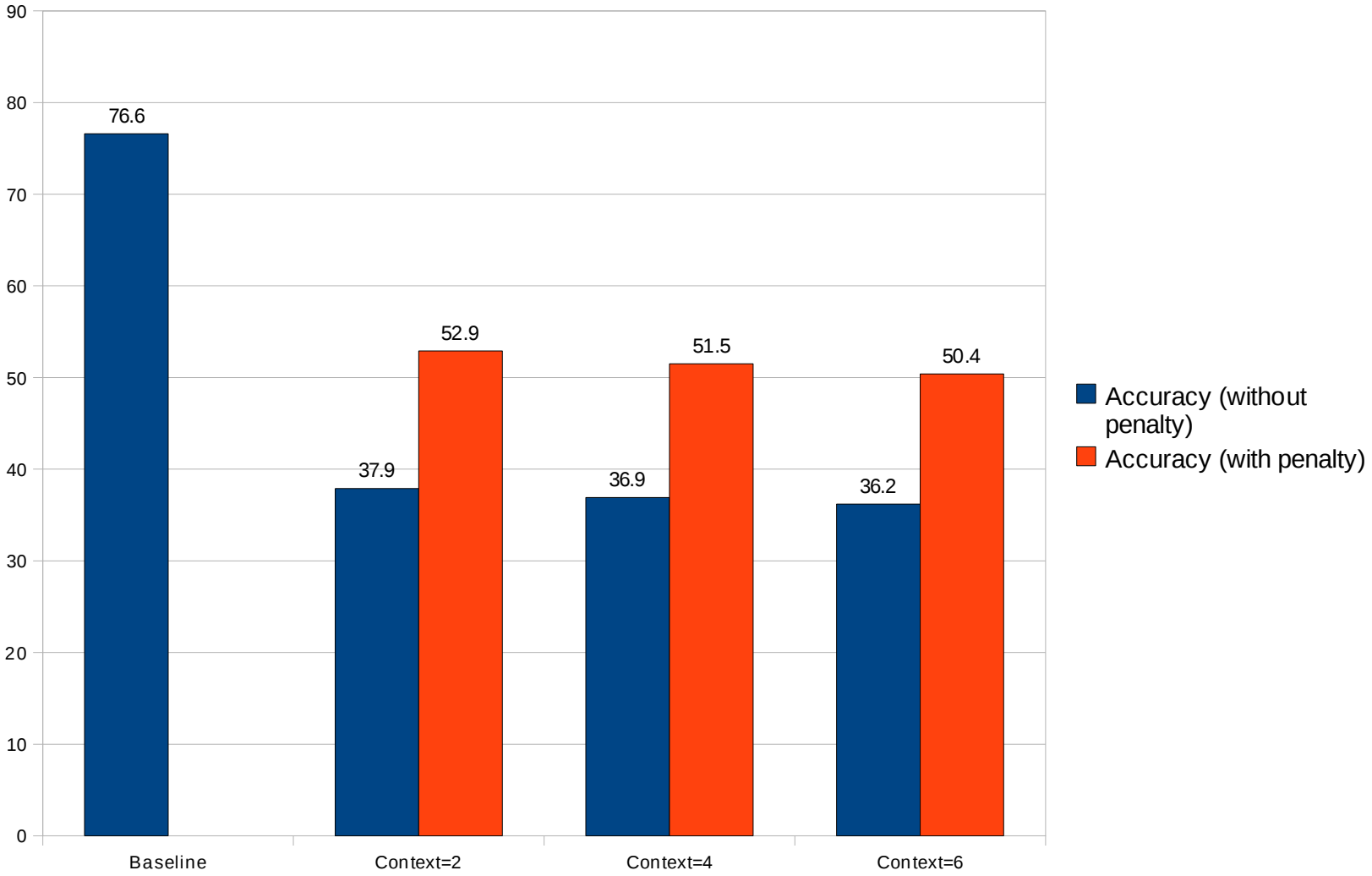
Where it screws up...



Improving Conceptual density

- If CD of sense *s* is 1, impose penalty *p*
- *p = log f,*
  - *f* is the sense number of the sense *s* for the word being disambiguated
- Intuition: Most frequent sense (first sense in Wordnet) for the word gets 0 penalty (log 1=0)

Conceptual density, improved



Lesk vs CD vs Random Walk