# Criação de thesauros distribucionais: estudo de parâmetros e robusteza

**Muntsa Padró**

Instituto de Informática
UFRGS

Dezembro 2013

# Goals

- Study the robustness of distributional thesauri.
- Evaluate different filter methods and parameters.

# Distributional Thesauri

- Given a target word, learn from data lists of similar words.

Example:
**Eat**: consume, devour, dine, swallow...

- Basic idea: similar words tend to appear in similar contexts.

# Distributional Thesauri

- Several works study different ways to stablish similarity between words:
  - What is context?
    - Bag of Words
    - Dependency relation
  - How do we assess the relevance of a context?
    - Frequency
    - PMI
    - LMI
    - ...
  - How to compute similarity between words?
    - Lin's measure
    - Cosine
    - Jaccard
    - ...

# In this work

- Focus on one common method for building thesauri: Lin measure (Lin, 1998).
- Study how context filtering modify it.
- More work has been done in comparing measures than studying the robustness in a given measure.

# Lin measure

- Use dependency parsed corpus.
- Contexts are triples.
- Asses relevance of triples using PMI.
- Computes similarity using Lin's formula.

# Contexts are triples

"O gato come peixe"

Context of "comer":

- "Gato" is the subject
- "Peixe" is the direct object

Codified as triples:

- (gato,subj,comer)
- (peixe,dobj,comer)

# General Idea

- For each verb, extract the triples it appears in, this is, the set of contexts.
- This triples allow us to compute similarity.
- **But** some triples may introduce noise, some filters need to be applied.

# Context Filters

1. Filter tripples by its frequency: remove triples under *th* frequency.
2. Keep most *p* relevant triples per verb. How to compute relevance?
   - Sort by frequency.
   - Sort by PMI.
   - Sort by LMI (frequency x PMI)

# Context Filters

1. Filter tripples by its frequency: remove triples under *th* frequency.
2. Keep most *p* relevant triples per verb. How to compute relevance?
   - **Sort by frequency.**
   - Sort by PMI.
   - Sort by LMI (frequency x PMI)

# Building the Thesauri

- BNC Corpus (English), parsed with RASP.
- Focus on verbs.
- Extract triples for all verbs appearing more than 50 times.
- Filter triples with one of the filters (minimum threshold or maximum number of triples).
- Compute similarity for all pairs of verbs to create a thesaurus

**Goal:** Study thesauri built with the different filters, studying influence of different parameters of each filter.

# Evaluation

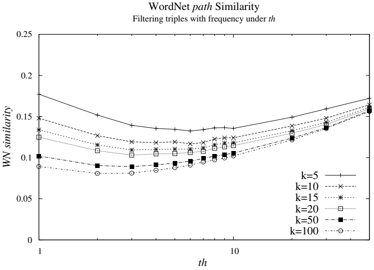
- Each verb has a list of neighbours.
- The neighbours are sorted by decresing similarity.
- Top rated neighbours should be the most similar verbs.
- Evaluate those lists comparing with WordNet

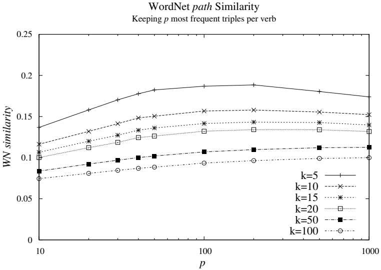
## Compare with WordNet

- Take the first *k* neighbors of a verb
- Compute average WordNet path similarities
- Compute the average over all verbs

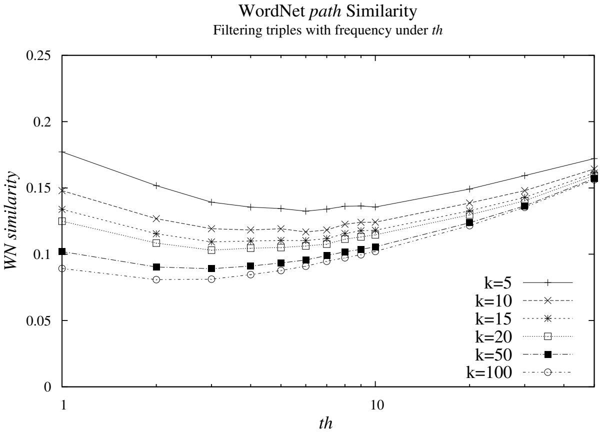
- First neighbours should be closer, so smaller *k* should have more similarity

# Results

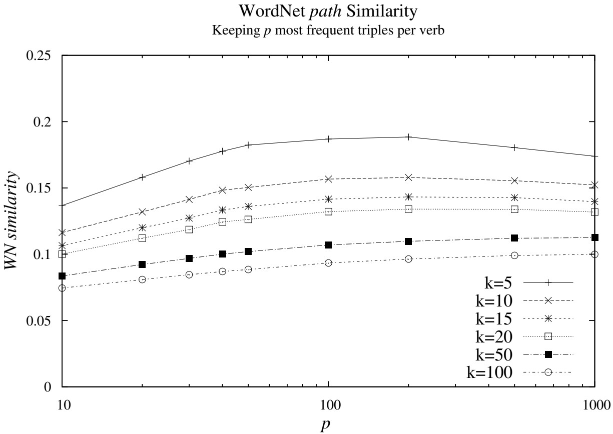

WordNet *path* Similarity
Filtering triples with frequency under *th*
WN similarity
0.25
0.2
0.15
0.1
0.05
0
1
10
*th*
k=5
k=10
k=15
k=20
k=50
k=100


WordNet *path* Similarity
Keeping *p* most frequent triples per verb
WN similarity
0.25
0.2
0.15
0.1
0.05
0
10
100
1000
*p*
k=5
k=10
k=15
k=20
k=50
k=100

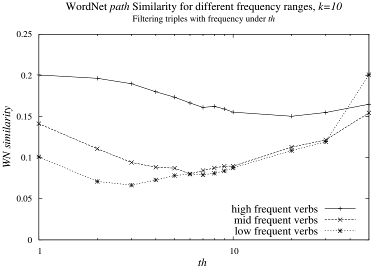
# Results

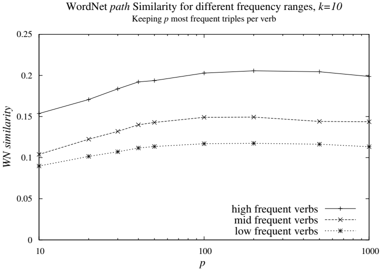

WordNet *path* Similarity
Filtering triples with frequency under *th*
*WN similarity*
0.25
0.2
0.15
0.1
0.05
0
1
10
*th*
k=5
k=10
k=15
k=20
k=50
k=100

# Results

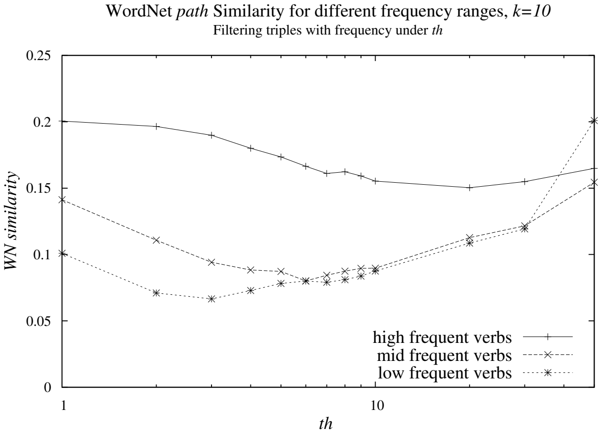

WordNet *path* Similarity
Keeping *p* most frequent triples per verb
*WN similarity*
0.25
0.2
0.15
0.1
0.05
0
10
100
1000
*p*
k=5
k=10
k=15
k=20
k=50
k=100

# Results

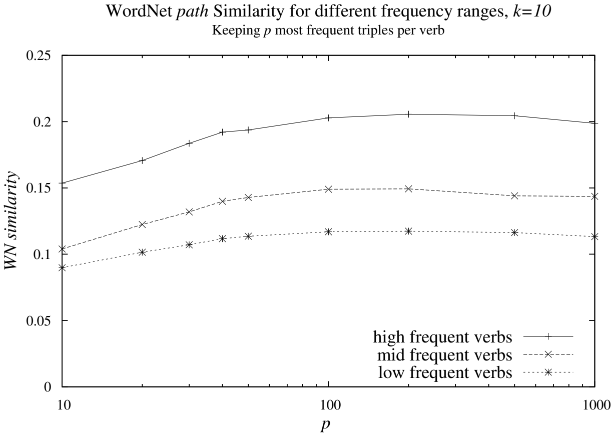

WordNet *path* Similarity for different frequency ranges, *k=10*
Filtering triples with frequency under *th*


WordNet *path* Similarity for different frequency ranges, *k=10*
Keeping *p* most frequent triples per verb

# Results


WordNet *path* Similarity for different frequency ranges, *k=10*
Filtering triples with frequency under *th*
WN similarity
0.25
0.2
0.15
0.1
0.05
0
1
10
th
high frequent verbs
mid frequent verbs
low frequent verbs

# Results


WordNet *path* Similarity for different frequency ranges, *k=10*
Keeping *p* most frequent triples per verb
*WN similarity*
0.25
0.2
0.15
0.1
0.05
0
10
100
1000
*p*
high frequent verbs
mid frequent verbs
low frequent verbs

# Conclusion

- Different filters lead to different results.
- The parameters of a given filter also change the results.
  - We should put more attention on what filters we use
- Frequency issues!
  - Methods for improving similarity measures for low frequent verbs should be developed

## Gràcies!

Obrigada!