HW#7

Distributional Semantics

- Goal:
  - Explore distributional semantic models
  - Compare effects of differences in context
  - Evaluate qualitatively & quantitatively

Task

- Construct distributional similarity models
- Use fixed data resources
  - Brown corpus data
- Compare similarity measures under models
- Compare correlation with human judgments

Mechanics

- Corpus reader:
  - Loading Brown corpus via NLTK
  - brown_words = list(nltk.corpus.brown.words())
  - brown_sents = list(nltk.corpus.brown.sents())
    - ~1.2M words
    - May want to develop on subset
      - e.g. brown_words = brown_words[0:10000]
        - Caveat: lexical gaps
- Correlation:
  - from scipy.stats.stats import spearmanr
    - A = spearmanr(list1,list2)
      - Return correlation coefficient, p-value
        - A.correlation

Details

- Windows:
  - “2” means two words before or after the modeled word
    - The quick brown fox jumped over the lazy dog .
- Weights:
  - “FREQ”: straight cooccurrence count (“term frequency”)

“PMI”

- Positive Pointwise Mutual Information
- Given the tabulated context vectors:

$$PPMI_{ij} = \max(\log_2 \frac{P_{ij}}{P_{i*}P_{*j}}, 0)$$

$$P_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}, P_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}, P_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

Word2Vec

- Compare results to word2vec
- Python package gensim
- model = gensim.models.Word2Vec(sents,size=100,window=2,min_count=1,workers=1)
  - sents are lists (arrays) of strings
- model.similarity('man','woman')

Notes

- Can work in any language you like