WordNet Similarity Metrics
Charlie Greenbacker

based on material from Jurafsky and Martin, *Speech and Language Processing* (2nd Ed.)
Ch. 20.6 Word Similarity: Thesaurus Methods, plus NLTK documentation
Overview

• Defining “word similarity”
• Use in Assignment 3
• Thesaurus vs. Distributional methods
• Five thesaurus-based similarity measures
• Tools for implementation
“Word Similarity”

- Synonymy
- Antonymy
- Hyponymy
- Hypernymy
- Meronymy
“Word Similarity”

- Synonymy
- Antonymy
- Hyponymy
- Hypernymy
- Meronymy
“Word Similarity”

• Synonymy
  • Binary relation between words
  • Two words are either synonyms or not
“Word Similarity”

• Synonymy
  • Binary relation between words
  • Two words are either synonyms or not

• Similarity
  • Looser metric of semantic distance
“Word Similarity”

• Similarity

  • More similar if words share more features of meaning

    • “near-synonyms”

  • Less similar if words have fewer common meaning elements

    • greater “semantic distance”
“Word Similarity”

- Relations between words
“Word Similarity”

• Relations between words
“Word Similarity”

• Relations between words

• Relations between senses
“Word Similarity”

- Relations between words
- Relations between senses
  - “bank” (financial sense) is more similar to “fund”
  - “bank” (river sense) is more similar to “slope”
Use in Assignment 3

• Spelling Correction with Semantics
  • Rank candidate corrections by similarity to nearby words
  • Compare candidates to words in some context window
  • Combine with minimum distance edit (or similar) for ranking
Thesaurus vs. Distributional

- Thesaurus methods
  - Measure distance between two senses
  - Use on-line thesaurus (e.g., WordNet, MeSH)

- Distributional methods
  - Estimate word similarity
  - Find words that have similar distributions in a corpus
Thesaurus vs. Distributional

- **Thesaurus methods**
  - Measure distance between two senses
  - Use on-line thesaurus (e.g., WordNet, MeSH)

- **Distributional methods**
  - Estimate word similarity
  - Find words that have similar distributions in a corpus
Five similarity measures

- Path length
- Resnik similarity
- Lin similarity
- Jiang-Conrath distance
- Extended Lesk measure
1. Path length

• Intuition: the shorter the path between two words/senses in a thesaurus hierarchy graph, the more similar they are

• Words are quite similar to parents & siblings

• Less similar to words far away in the network
1. Path length

Figure 19.6 A fragment of the WordNet hypernym hierarchy, showing path lengths from *nickel* to *coin* (1), *dime* (2), *money* (5), and *Richter scale* (7).

$$\text{pathlen}(c_1, c_2) = \text{number of edges in shortest path}$$
I. Path length

\[ \text{pathlen}(c_1, c_2) = \text{number of edges in shortest path} \]

- Path-based similarity often involves a log transform

- **path-length based similarity:**

  \[ \text{sim}_{path}(c_1, c_2) = - \log \text{pathlen}(c_1, c_2) \]

- Weaknesses: requires sense-tagged data, assumes uniform cost
2. Resnik similarity

- information-content word-similarity:
  - Still relies on structure of thesaurus
  - Refines path-based approach using normalizations based on hierarchy depth
  - Represents distance associated with each edge
  - Adds probabilistic information derived from a corpus
2. Resnik similarity

- Probability of random word being an instance of concept \( c \):

\[
P(c) = \frac{\sum_{w \in \text{words}(c)} \text{count}(w)}{N}
\]

where \( \text{words}(c) \) is set of words subsumed by concept \( c \),
\( N \) is the number of words in corpus and also in thesaurus.

- \( P(\text{root}) = 1 \) since all words are subsumed by root concept

- The lower a concept in the hierarchy, the lower the probability
2. Resnik similarity

- Train probabilities by counting in a corpus: each word counts as an occurrence of all concepts “containing” it.
2. Resnik similarity

• Two more definitions are needed...

• Information content of a concept $c$: $IC(c) = -\log P(c)$

• basic information theory

• Lowest common subsumer: $LCS(c_1, c_2)$

  = lowest node in hierarchy that is a hypernym of $c_1$ & $c_2$
2. Resnik similarity

• Finally... the Resnik similarity measure:

\[
\text{sim}_{\text{Resnik}}(c_1, c_2) = -\log P(\text{LCS}(c_1, c_2))
\]

• estimates common amount of information between words by information content of lowest common subsumer
3. Lin similarity

• Similarity is about more than just common information... the more differences between A & B, the less similar they are

• Commonality: \( IC(\text{common}(A, B)) \)

• Difference: \( IC(\text{description}(A, B)) - IC(\text{common}(A, B)) \)

where \( \text{description}(A, B) \) “describes” A and B
3. Lin similarity

- **Similarity Theorem**: The Similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are.

\[
sim_{Lin}(A, B) = \frac{\text{common}(A, B)}{\text{description}(A, B)}
\]

- the information in common between two concepts is twice the information in the lowest common subsumer
3. Lin similarity

- Final **Lin similarity function** for concepts in a thesaurus:

\[
sim_{Lin}(c_1, c_2) = \frac{2 \times \log P(LCS(c_1, c_2))}{\log P(c_1) + \log P(c_2)}
\]

- For example, from Figure 20.7:

\[
sim_{Lin}(hill, coast) = \frac{2 \times \log P(geological-formation)}{\log P(hill) + \log P(coast)} = 0.59
\]
4. Jiang-Conrath distance

- Related to $sim_{Lin}$ - expressed as distance instead of similarity:

$$dist_{JC}(c_1, c_2) = 2 \times \log P(LCS(c_1, c_2)) - (\log P(c_1) + \log P(c_2))$$

- Transform into a similarity measure by taking the reciprocal:

$$sim_{JC}(c_1, c_2) = \frac{1}{2 \times \log P(LCS(c_1, c_2)) - (\log P(c_1) + \log P(c_2))}$$
5. Extended Lesk measure

- Extends Lesk algorithm for word sense disambiguation
- Dictionary-based method
  - makes use of glosses, a property of dictionaries
5. Extended Lesk measure

• **Extended gloss overlap**: two concepts/senses are similar if their glosses contain overlapping words

  • *drawing paper*: paper that is specially prepared for use in drafting

  • *decal*: the art of transferring designs from specially prepared paper to a wood or glass or metal surface

• For each n-word phrase seen in both glosses, eLesk adds $n^2$; longer overlaps are rare, and should be weighted more heavily
5. Extended Lesk measure

- **Extended gloss overlap**: two concepts/senses are similar if their glosses contain overlapping words

- *drawing paper*: paper that is *specially prepared* for use in drafting

  \[ l^2 + 2^2 = 5 \]

- *decal*: the art of transferring designs from *specially prepared paper* to a wood or glass or metal surface

- For each n-word phrase seen in both glosses, eLesk adds \( n^2 \); longer overlaps are rare, and should be weighted more heavily
5. Extended Lesk measure

- Extends Lesk looks for overlap in hypernyms, hyponyms, meronyms, and other relations of the two concepts... not just in the glosses of the two synsets

- if considering hyponyms only, and \textit{gloss}(\textit{hypo}(A)) as the concatenation of all glosses of all hyponym senses of A, then the total relationship between concepts A and B is:

\[
\text{similarity}(A, B) = \text{overlap}(\text{gloss}(A), \text{gloss}(B))
+ \text{overlap}(\text{gloss}(\text{hypo}(A)), \text{gloss}(\text{hypo}(B)))
+ \text{overlap}(\text{gloss}(A), \text{gloss}(\text{hypo}(B)))
+ \text{overlap}(\text{gloss}(\text{hypo}(A)), \text{gloss}(B))
\]
5. Extended Lesk measure

Let $RELS$ be the set of possible WordNet relations with glosses we compare, then define **Extended Lesk** measure as:

$$sim_{eLesk}(c_1, c_2) = \sum_{r,q \in RELS} \text{overlap}(\text{gloss}(r(c_1)), \text{gloss}(q(c_2)))$$
Implementation Tools

- No need to implement these similarity metrics on your own
  - NLTK WordNet interface for Python
    - Wordnet::Similarity package for Perl
      - ported to Java: http://www.cogs.susx.ac.uk/users/drh21/
>>> import nltk
>>> from nltk.corpus import wordnet as wn
>>> dog = wn.synset('dog.n.01')
>>> cat = wn.synset('cat.n.01')
>>> from nltk.corpus import wordnet_ic
>>> brown_ic = wordnet_ic.ic('ic-brown.dat')
>>> dog.path_similarity(cat)
0.2
>>> dog.res_similarity(cat, brown_ic)
7.911666509036577
>>> dog.lin_similarity(cat, brown_ic)
0.8768009843733973
>>> dog.jcn_similarity(cat, brown_ic)
0.4497755285516739
>>> # NLTK does not implement eLesk(?)
>>> # but offers Leacock-Chodorow & Wu-Palmer measures
Wordnet::Similarity

- Includes all five metrics discussed (even eLesk), and others
  - Sourceforge: http://wn-similarity.sourceforge.net/
  - CPAN: http://search.cpan.org/dist/WordNet-Similarity/
- Web interface demos (servers are often busy):
  - http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi
  - http://talisker.d.umn.edu/cgi-bin/similarity/similarity.cgi
So which metric is best?

• That’s for you to figure out!