This assignment is due **Wednesday, December 8 at 11:59PM**.

Goals

Through this assignment you will:

- Explore issues in word sense disambiguation.
- Gain familiarity with WordNet and the WordNet API.
- Gain some further familiarity with NLTK.
- Implement a thesaurus-based word sense disambiguation technique on standard data.

[Back to Top]

Background

Please review the class slides and readings in the textbook on lexical semantics, including WordNet, and word sense disambiguation. Also please read the article Section 5.1, describing Resnik's word sense disambiguation in groupings approach in detail.

Note: You will be implementing a somewhat simplified version of Resnik's approach as detailed below.

For additional information on NLTK's WordNet API and information content measures, see:

- the NLTK WordNet HowTo
- NLTK Word corpus reader info

[Back to Top]

Implementing Word Sense Disambinguation and Similarity using Resnik's Similarity Measure

Based on the examples in the text, class slides, and other resources, implement a program to perform Word Sense Disambiguation based on noun groups, using Resnik's method and WordNet-based similarity measure. Then compute and compare similarity scores for a set of human judgments. Specifically, your program should:

- Load information content values for WordNet from a file.
- Read in a file of (`probe word, noun group`) pairs
- For each (`probe word, noun group`) pair:

a. Use "Resnik similarity" based on WordNet and information content to compute the preferred WordNet sense for the probe word given the noun group context.

b. On a single line, for each `(probe word, noun group word)` pair:

- print the similarity between the probe word and each noun group word in the format
  - `(probe word, noun group word, Resnik similarity score)`

- On a separate line, print out the preferred sense, by synsetID, of the word.
- Read in a file of human judgments of similarity between pairs of words.
- For each word pair in the file:
  - Compute the similarity between the two words, using the **Resnik similarity** measure
  - Print out the similarity as:
    - `wd1,wd2:similarity`

- Lastly, compute and print the Spearman correlation between the similarity scores you have computed and the human-generated similarity scores in the provided file as:
  - `Correlation:computed_correlation`.
  - You may use any available software for computing the correlation. In Python, you can use **`spearmanr`** from **`scipy.stats`**.

NOTE: You do not need to select senses for all words, only for the probe word; this is a simplification of the word group disambiguation model in the paper.
NOTE: You may treat all the words in context groups as nouns. You are not responsible for cross-POS similarity.

[Back to Top]

Programming

Create a program **`hw8_resnik_wsd.sh`** that implements the disambiguation specified as above invoked as:

```
hw8_resnik_wsd.sh <information_content_file> <wsd_test_filename> <judgment_file>
<output_filename>
```

- `<information_content_file>`
  - This string should specify the source of the information content file. It will be passed to `nltk.corpus.wordnet_ic.ic()` on patas.
- `<wsd_test_filename>`
  - This is the name of the file that contains the lines of "`probe-word, noun group words`" pairs on which to evaluate your system.
- `<judgment_filename>`
  - The name of the input file holding human judgments of the pairs of words and their similarity to evaluate against, `mc_similarity.txt`.
  - Each line is of the form:
    - `wd1,wd2,similarity_score`
- `<output_filename>`
  - This is the name of the file to which you should write your results.

Implementation Resources

Resnik's similarity measure relies on two components:

- the WordNet taxonomy
  - NLTK implements a Python implementation of the WordNet API which you are encouraged to use. There are other WordNet APIs, and you may use them, but they come with no warranty, and may require substantial effort to work with.
    NOTE: You may use the API to access components of WordNet, extract synsets, identify hypernyms, etc. You may **NOT** use the methods which directly implement the similarity measure or the identification of Most Informative Subsumer. You **must** implement those functions yourself as

procedures for the similarity calculcation. You may use accessors such as common_hypernyms and information_content. If you have questions about the admissibility of a procedure, please ask; I'll clarify as quickly as I can.

- a corpus-based information content measure.
    - The NLTK corpus reader provides a number of resources for information content calculation including frequency tables indexed by WordNet offset and part-of-speech in `/corpora/nltk/nltk-data/corpora/wordnet_ic/`.
      For consistency and quality, I would suggest that you use `/corpora/nltk/nltk-data/corpora/wordnet_ic/ic-brown-resnik-add1.dat`, which derives its counts from the 'balanced' Brown Corpus, using fractional counts and add1 smoothing to avoid zero counts. (Not that there aren't other problems with words not in WordNet…) You may use this source either through the NLTK API (as in `wnic = nltk.corpus.wordnet_ic.ic('ic-brown-resnik-add1.dat')` or directly through methods that you implement yourself. The file is flat text.

      NOTE: The IC files assume that you are using WordNet 3.0. If you choose to use a different API but want to use the precomputed IC measures, you must make sure to use WordNet version 3.0, or the IC measures will be inconsistent.

Files

All files are found in `/dropbox/21-22/571/hw8/` on patas:

Test, Gold Standard, and Example

- **`wsd_contexts.txt`**
    - File of probe words with disambiguation word grouping lists. Each line is formatted as:
        - `probe_word\tnoun_group`
            - `probe_word` is the word to disambiguate
            - `noun_group` is the comma-separated word list that serves as disambiguation context
- **`wsd_contexts.txt.gold`**
    - Corresponding file with gold standard sense tags, in which the sense id and gloss are prepended to the original line.
    - **NOTE:** These are manually constructed gold standards and glosses; you are not expected to produce the gloss in your own output. This file is for reference purposes.
- **`mc_similarity.txt`**
    - These are the pairs of words whose similarity is to be evaluated under each of your models, along with human similarity judgments from [Miller and Charles, 1991]. Each line is of the form
        - `wd1,wd2,similarity_score`
- **`example_output.txt`**
    - Formatted (partial) example output file.
    - **Note:** Please note that the Resnik WSD approach may not be able to disambiguate these words correctly. You will probably achieve about 60% accuracy overall. The earlier instances are generally easier than the later ones.

Submission Files

- **`hw8.tar.gz`**: Tarball containing the following:
    - **`hw8_resnik_wsd.sh`**: Program which implements and evaluates your WordNet-based word sense disambiguation algorithm based on Resnik's approach.
    - **`hw8_output.txt`**: Output of running your program with the NLTK information content file.
- **`readme.{txt|pdf}`**: Write-up file

- This file should describe and discuss your work on this assignment. Include problems you came across and how (or if) you were able to solve them, any insights, special features, and what you learned. Give examples if possible. If you were not able to complete parts of the project, discuss what you tried and/or what did not work. This will allow you to receive maximum credit for partial work.
    - In particular, you should discuss the successes and failures of the algorithm in predicting the desired word senses, as well as comparing the quality of correlation with human judgment.

[Back to Top]