MP1
Oct.05 M
LING 83600 Language Technology
Yuying Ren

Part 1 result:
Number of word pairs from human judgment file: 203

	Word pairs covered	Coverage rate	Spearman Correlation
Path similarity	107	53%	.5883
Leacock-Chodorow similarity	199	98%	.6
Wu-Palmer similarity	107	53%	.6379
Resnik similarity	203	100%	.5552
Jiang-Conrath similarity	203	100%	.6079
Lin similarity	203	100%	.5217

For part 1, I mainly used dictionaries to store the data. Word pairs are tupled in a list and used as the key of dictionaries. I have 3 types of dictionaries: word pairs – human scores(for calculating the spearman correlation and coverage); word pairs – synsets pairs(for calculating the similarity with each method); word pairs -- scores calculate with each methods(for comparing with human scores and calculating the results).
The main difficulty I faced was the word-pairs that are not calculatable by these methods. My solution was separating the calculation and the loop that runs the word pairs – synset pairs dictionary functions, so the word pairs that can't be calculated will pass the loop without being counted.
For calculating the similarity, I made one function for each of the method, and stored the similarity scores of all the synset pairs of each pair of words in a list. The function will choose the highest similarity via a max() function on the list.

Part 2 result:

	Word pairs covered	Coverage rate	Spearman Correlation
PPMI	158(152?)	77.8%(74.8%?)	-.3799

The Problem I had in this part was tokenizing the news data. I didn't realize there was a .split() function in the ppmi.py script for the tokenized data, so my old file with all words in a single line

didn’t work, and returned 0 pairs of words. I fixed this by adding a while space as separators in my tokenized data. There are 158 word pairs covered in the result file, but when I calculate the spearman correlation, only 78 of them can be found in my human judgement data, I made the function to ignore the order of word pairs in the two files and got 152 word pairs in final, and calculated the correlation with those pairs.

Part3 result:

	Word pairs covered	Coverage rate	Spearman Correlation
Word2vec	202	99.5%	.6474