Assignment

UE17CS333- Natural Language Processing

In this exercise you will:

- Measure semantic similarity of words using WordNet

- calculate WordNet polysemy

- Compute lesk similarity of two concepts using WordNet

Exercise 1: WordNet semantic similarity

Use the path-based similarity measures to score the similarity of each of the following pairs of words: car-automobile, gem-jewel, journey-voyage, boy-lad, coast-shore, asylum-madhouse, magician-wizard, midday-noon, furnace- stove, food-fruit, bird-cock, bird-crane, tool-implement, brother-monk, lad- brother, crane-implement, journey-car, monk-oracle, cemetery-woodland, food- rooster, coast-hill, forest-graveyard, shore-woodland, monk-slave, coast-forest, lad-wizard, chord-smile, glass-magician, rooster-voyage, noon-string.

- In noun_similarity.py implement the function get similarity scores (pairs) so that it ranks the pairs in order of decreasing similarity.

Hint: the similarity of a pair should be represented by the similarity of the most similar pair of synsets they have.

Exercise 2: WordNet polysemy

The polysemy of a word is the number of senses it has. Using WordNet, we can determine that the noun dog has 7 senses.

- In average_polysemy.py implement the function average polysemy (part_of_speech) so that it computes the average polysemy of nouns, verbs, adjectives and ad-verbs according to WordNet.

Exercise 3: Lesk similarity

Lesk similarity is defined as the textual overlap between the corresponding definitions, as provided by a dictionary. In this exercise you will have to implement some methods to measure lesk similarity of two concepts.

Submission will be via mail on ID amj.nlp2019@gmail.com