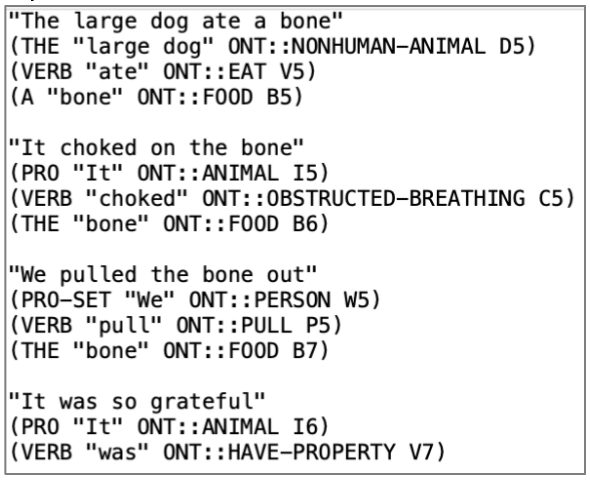
NLP 447 Project 3 Strategy

Yuan Wang / ywang340

- **Step 1: Convert input file to list format** so that it can be convenient for the computation steps.
  Input like this:



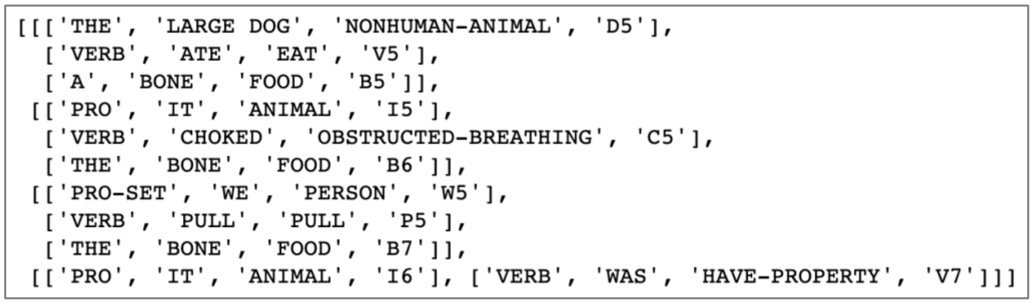
```
"The large dog ate a bone"
(THE "large dog" ONT::NONHUMAN-ANIMAL D5)
(VERB "ate" ONT::EAT V5)
(A "bone" ONT::FOOD B5)

"It choked on the bone"
(PRO "It" ONT::ANIMAL I5)
(VERB "choked" ONT::OBSTRUCTED-BREATHING C5)
(THE "bone" ONT::FOOD B6)

"We pulled the bone out"
(PRO-SET "We" ONT::PERSON W5)
(VERB "pull" ONT::PULL P5)
(THE "bone" ONT::FOOD B7)

"It was so grateful"
(PRO "It" ONT::ANIMAL I6)
(VERB "was" ONT::HAVE-PROPERTY V7)
```

  Convert to like this:

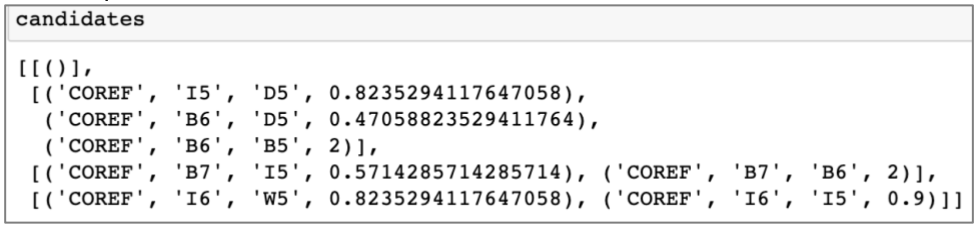


```
[[['THE', 'LARGE DOG', 'NONHUMAN-ANIMAL', 'D5'],
  ['VERB', 'ATE', 'EAT', 'V5'],
  ['A', 'BONE', 'FOOD', 'B5']],
 [['PRO', 'IT', 'ANIMAL', 'I5'],
  ['VERB', 'CHOKED', 'OBSTRUCTED-BREATHING', 'C5'],
  ['THE', 'BONE', 'FOOD', 'B6']],
 [['PRO-SET', 'WE', 'PERSON', 'W5'],
  ['VERB', 'PULL', 'PULL', 'P5'],
  ['THE', 'BONE', 'FOOD', 'B7']],
 [['PRO', 'IT', 'ANIMAL', 'I6'], ['VERB', 'WAS', 'HAVE-PROPERTY', 'V7']]]
```

- **Step 2:** Select **Wu-Palmer similarity** on **the TRIPS ontology** to compute similarity score. Wu-Palmer similarity is between [0, 1].
- **Step 3: Get Candidates list**. The method is optimized based on input4 & 6 examples. The hidden inputs cases are not considered in this code because their special case is unknown.
  1. Get input from step 1, if sentence [i] and sentence [i-1] have same <word> and <type>, pair them and assign similarity = 2, and this similarity is > Wu-Palmer similarity score. This will benefit for the score comparison.
  2. If they are not same, pair them and compute Wu-Palmer similarity according to their <type> Ontology.
  3. If <word> contains "SELF" or "SELVES" like input 4, get instance refer from same sentence, pair, and assign similarity = 2
  4. Compare sentence[i] and the sentence before [i-1], if they have same entry, assign them together. Assign similarity = 0.9 to consider input6 case, where two

boats are the same boat (to make this similarity a little smaller than the other one).

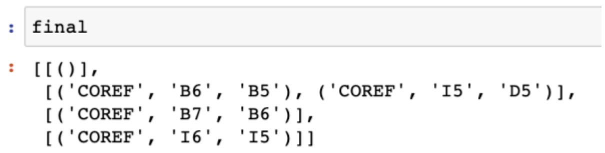
The output of candidate looks like this:



```
candidates

[[()],
 [('COREF', 'I5', 'D5', 0.8235294117647058),
  ('COREF', 'B6', 'D5', 0.47058823529411764),
  ('COREF', 'B6', 'B5', 2)],
 [('COREF', 'B7', 'I5', 0.5714285714285714), ('COREF', 'B7', 'B6', 2)],
 [('COREF', 'I6', 'W5', 0.8235294117647058), ('COREF', 'I6', 'I5', 0.9)]]
```

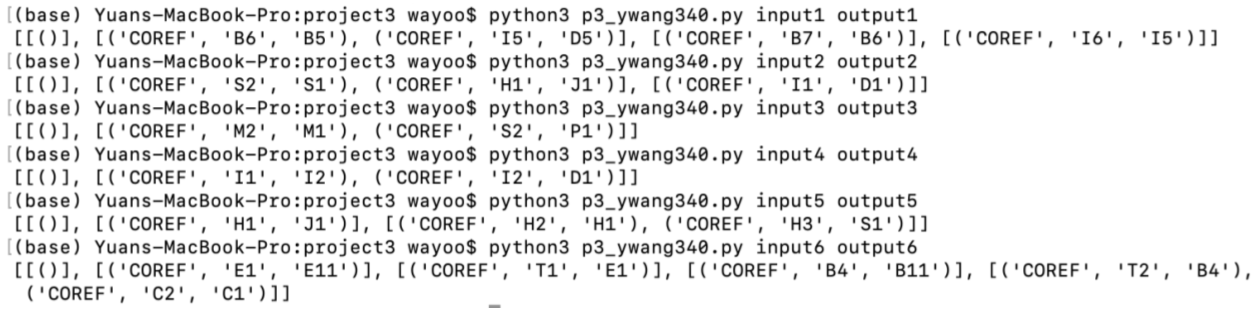
- **Step 4: final output selection**. Pull all candidates that are 100% matching, then filter out candidates by reflexive constrains, finally select the candidates with higher similarity score, remove all the other candidates.
  The final output is like this:



```
: final

: [[()],
   [('COREF', 'B6', 'B5'), ('COREF', 'I5', 'D5')],
   [('COREF', 'B7', 'B6')],
   [('COREF', 'I6', 'I5')]]
```

Result:



```
(base) Yuans-MacBook-Pro:project3 wayoo$ python3 p3_ywang340.py input1 output1
[[()], [('COREF', 'B6', 'B5'), ('COREF', 'I5', 'D5')], [('COREF', 'B7', 'B6')], [('COREF', 'I6', 'I5')]]
(base) Yuans-MacBook-Pro:project3 wayoo$ python3 p3_ywang340.py input2 output2
[[()], [('COREF', 'S2', 'S1'), ('COREF', 'H1', 'J1')], [('COREF', 'I1', 'D1')]]
(base) Yuans-MacBook-Pro:project3 wayoo$ python3 p3_ywang340.py input3 output3
[[()], [('COREF', 'M2', 'M1'), ('COREF', 'S2', 'P1')]]
(base) Yuans-MacBook-Pro:project3 wayoo$ python3 p3_ywang340.py input4 output4
[[()], [('COREF', 'I1', 'I2'), ('COREF', 'I2', 'D1')]]
(base) Yuans-MacBook-Pro:project3 wayoo$ python3 p3_ywang340.py input5 output5
[[()], [('COREF', 'H1', 'J1')], [('COREF', 'H2', 'H1'), ('COREF', 'H3', 'S1')]]
(base) Yuans-MacBook-Pro:project3 wayoo$ python3 p3_ywang340.py input6 output6
[[()], [('COREF', 'E1', 'E11')], [('COREF', 'T1', 'E1')], [('COREF', 'B4', 'B11')], [('COREF', 'T2', 'B4'),
 ('COREF', 'C2', 'C1')]]
```