CE314 ASSIGNMENT2

Parsing and Word Similarity

1603930

Question 1a)

For S1 needed to add:

To S -> VP

To V -> “Put”

To N -> “block” | “table”

For S2 needed to add:

To PropN -> “Bob”

To P -> “in” | “along”

To N -> “river”

For S3 needed to add:

To V - > “chase”

To Adj -> “furry”

To N -> “dog”

Question 1b)

For S1: 1 derivations.

For S2: 5 derivations.

For S3: 1 derivation.

Output:

```
C:\Users\Alex\.virtualenvs\ass2\Scripts\python.exe
D:/Desktop/ce314/ass2/q1.py

!!!!!!!! S1: !!!!!!!!

(S

  (VP (V Put) (NP (Det the) (Nom (N block)))

  (PP (P on) (NP (Det the) (Nom (N table)))))

!!!!!!!! S2: !!!!!!!!

(S
  (NP (PropN Bob))
  (VP
    (V chased)
    (NP
      (NP
        (NP (Det a) (Nom (N bear)))
        (PP (P in) (NP (Det the) (Nom (N park)))))
      (PP (P along) (NP (Det the) (Nom (N river))))))))
(S
  (NP (PropN Bob))
  (VP
    (V chased)
    (NP
      (NP (Det a) (Nom (N bear)))
      (PP
        (P in)
        (NP
          (NP (Det the) (Nom (N park)))
          (PP (P along) (NP (Det the) (Nom (N river))))))))))
(S
  (NP (PropN Bob))
  (VP
    (VP
      (VP (V chased) (NP (Det a) (Nom (N bear))))
      (PP (P in) (NP (Det the) (Nom (N park)))))
    (PP (P along) (NP (Det the) (Nom (N river)))))))
(S
  (NP (PropN Bob))
  (VP
    (VP
      (V chased)
      (NP
        (NP (Det a) (Nom (N bear)))
        (PP (P in) (NP (Det the) (Nom (N park))))))
    (PP (P along) (NP (Det the) (Nom (N river)))))
(S
  (NP (PropN Bob))
  (VP
    (VP (V chased) (NP (Det a) (Nom (N bear))))
    (PP
      (P in)
      (NP
        (NP (Det the) (Nom (N park)))
        (PP (P along) (NP (Det the) (Nom (N river))))))))
!!!!!!!! S3: !!!!!!!!
(S
  (NP (PropN Bill))
  (VP
    (V saw)
    (S
      (NP (PropN Bob))
      (VP
        (V chase)
        (NP
          (Det the)
          (Nom (Adj angry) (Nom (Adj furry) (Nom (N dog)))))))))

Process finished with exit code 0
```

Question 2a)

S4 is incorrect. Correct equivalents are: **A bear eats a squirrel, Bears eat squirrels**. This is because “bear” and “squirrel” initials are not vowels (“a” is used instead of “an”).

S5 is incorrect. Correct equivalents are: **The dogs eat, The dog eats**. As dogs is an NNS “eat” needs to be used instead of “eats”.

Question 2b)

Output:

```
C:\Users\Alex\.virtualenvs\ass2\Scripts\python.exe
D:/Desktop/ce314/ass2/q2.py

!!!!!!!! S4: !!!!!!!!

(S

  (NP (Det An) (Nom (N bear)))

  (VP (V eat) (NP (Det an) (Nom (N squirrel)))))

!!!!!!!! S5: !!!!!!!!

(S (NP (Det The) (Nom (N dogs))) (V eats))


Process finished with exit code 0
```

S4 is correct because the incorrect words “An” and “eat” are placed in the right location in the chart, “An” is placed as a Det which is also where “a” is placed and “eat” is placed where “eats” is located (V). This means these words will be treated the same as their corrections.

S5 is correct because the two correct equivalents also share the same sentence structure; The (Det) dogs/dog (N) eat/eats (V) which means the output for both parsed sentences will be the same. However, the output is of NP V (which is a valid English sentence structure) is added to S, so S -> NP V.

Question 2c)

(Using the chart parser)

S_incorrect1: “The squirrel eat table”

S_correct1: “The squirrels eat tables”

In this case, the grammar established thus far will work with the first sentence. However, if S -> NP VP was to be removed the sentence would not be parsed as both structures resemble NP VP.

S_incorrect2: “Bob saw Bill eats a bear”

S_correct2: “Bob saw Bill eat a bear”

The second sentence requires new values of “squirrels” and “tables” and as a result does not parse. To counter this, an extension is added to Noun as “squirrels and “tables”. However, this does not fix the issue of not parsing the sentence. This fix this done by adding N to NP (NP -> N). This is added because the word “tables” is a noun phrase and “eat” is a verbal phrase in this context. This produces this output:

```
!!!!!!!! S_incorrect1: !!!!!!!!

(S (NP (Det The) (Nom (N squirrel))) (VP (V eat) (NP (N table))))

!!!!!!!! S_correct1: !!!!!!!!

(S (NP (Det The) (Nom (N squirrels))) (VP (V eat) (NP (N tables))))
```

Question 3a)

S6: “He eats pasta with some anchovies in the restaurant”

Interpretation 1 is “He eats pasta served with some anchovies in the restaurant”.

Interpretation 2 is “He eats pasta using some of the anchovies in the restaurant”.

Interpretation 3 is “He eats pasta with some anchovies that are in the restaurant”.

S7: “He eats pasta with a fork in the restaurant”

Interpretation 1 is “He eats pasta using a fork in the restaurant”.

Interpretation 2 is “He eats pasta that contains a fork in a restaurant”.

Interpretation 3 is “He eats pasta with a fork that is in the restaurant”.

Question 3b)

Earley Chart Parser:

(Using grammar from Q1) To parse this I had to add a new category called PRP (personal noun) for "He", new category called VBZ (third person verb) for "eats" and new category called NNS (plural nouns) for "anchovies". Also added "with" to P, "some" to Det, "pasta" and "restaurant" to N.

To make these work these extensions were also added:

```
Nom -> NNS

NP -> PRP | N

VP -> VBZ NP PP
```

To make S7 parse, I had to add "fork" to N.

S6:

Interpretation 1

```
(S
  (NP (PRP He))
  (VP
    (VBZ eats)
    (NP
      (NP (N pasta))
      (PP (P with) (NP (Det some) (Nom (NNS anchovies)))))
    (PP (P in) (NP (Det the) (Nom (N restaurant))))))
```

Interpretation 2

This does not detect ambiguity for this interpretation because the grammar is not programmed to detect the use of "with" using anchovies as a tool/instrument.

Interpretation 3

```
(S
  (NP (PRP He))
  (VP
    (VBZ eats)
    (NP (N pasta))
    (PP
      (P with)
      (NP
        (NP (Det some) (Nom (NNS anchovies)))
        (PP (P in) (NP (Det the) (Nom (N restaurant))))))))
```

S7:

Interpretation 1

This does not detect ambiguity as the grammar is not programmed to detect “fork” as a verb, but instead a noun.

Interpretation 2

```
(S
  (NP (PRP He))
  (VP
    (VBZ eats)
    (NP (NP (N pasta)) (PP (P with) (NP (Det a) (Nom (N fork)))))
    (PP (P in) (NP (Det the) (Nom (N restaurant))))))
```

Interpretation 3

```
(S
  (NP (PRP He))
  (VP
    (VBZ eats)
    (NP (N pasta))
    (PP
      (P with)
      (NP
        (NP (Det a) (Nom (N fork)))
        (PP (P in) (NP (Det the) (Nom (N restaurant))))))))
```

Shift Reduce Parser (using trace=2)

This does not detect ambiguity because by its nature it only prints out one tree.

Output:

```
C:\Users\Alex\.virtualenvs\ass2\Scripts\python.exe
D:/Desktop/ce314/ass2/q3.py

Warning: S -> VP PP will never be used

Warning: NP -> N will never be used

Warning: V -> 'eats' will never be used

!!!!!!!! S6: !!!!!!!!

Parsing 'He eats pasta with some anchovies in the restaurant'

    [ * He eats pasta with some anchovies in the restaurant]

  S [ 'He' * eats pasta with some anchovies in the restaurant]

  R [ PRP * eats pasta with some anchovies in the restaurant]

  R [ NP * eats pasta with some anchovies in the restaurant]

  S [ NP 'eats' * pasta with some anchovies in the restaurant]

  R [ NP V * pasta with some anchovies in the restaurant]

  S [ NP V 'pasta' * with some anchovies in the restaurant]

  R [ NP V N * with some anchovies in the restaurant]

  R [ NP V NP * with some anchovies in the restaurant]

  R [ NP VP * with some anchovies in the restaurant]

  R [ S * with some anchovies in the restaurant]

  S [ S 'with' * some anchovies in the restaurant]

  R [ S P * some anchovies in the restaurant]

  S [ S P 'some' * anchovies in the restaurant]

  R [ S P Det * anchovies in the restaurant]

  S [ S P Det 'anchovies' * in the restaurant]

  R [ S P Det NNS * in the restaurant]

  R [ S P Det Nom * in the restaurant]

  R [ S P NP * in the restaurant]

  R [ S PP * in the restaurant]

  S [ S PP 'in' * the restaurant]

  R [ S PP P * the restaurant]

  S [ S PP P 'the' * restaurant]

  R [ S PP P Det * restaurant]

  S [ S PP P Det 'restaurant' * ]
  R [ S PP P Det N * ]
  R [ S PP P Det NP * ]
!!!!!!!! S7: !!!!!!!!
Parsing 'He eats pasta with a fork in the restaurant'
    [ * He eats pasta with a fork in the restaurant]
  S [ 'He' * eats pasta with a fork in the restaurant]
  R [ PRP * eats pasta with a fork in the restaurant]
  R [ NP * eats pasta with a fork in the restaurant]
  S [ NP 'eats' * pasta with a fork in the restaurant]
  R [ NP V * pasta with a fork in the restaurant]
  S [ NP V 'pasta' * with a fork in the restaurant]
  R [ NP V N * with a fork in the restaurant]
  R [ NP V NP * with a fork in the restaurant]
  R [ NP VP * with a fork in the restaurant]
  R [ S * with a fork in the restaurant]
  S [ S 'with' * a fork in the restaurant]
  R [ S P * a fork in the restaurant]
  S [ S P 'a' * fork in the restaurant]
  R [ S P Det * fork in the restaurant]
  S [ S P Det 'fork' * in the restaurant]
  R [ S P Det N * in the restaurant]
  R [ S P Det NP * in the restaurant]
  S [ S P Det NP 'in' * the restaurant]
  R [ S P Det NP P * the restaurant]
  S [ S P Det NP P 'the' * restaurant]
  R [ S P Det NP P Det * restaurant]
  S [ S P Det NP P Det 'restaurant' * ]
  R [ S P Det NP P Det N * ]
  R [ S P Det NP P Det NP * ]

Process finished with exit code 0
```

Question 4)

Complete output is in text files. All programs have been made so that they print but also output into a file. However, here are the **first 10** values of outputs for each task:

Task 1 (in BioSim-100-predicted.txt):

```
word1 word2 GoldSimilartiy WordNetSimilarity

old new 1.58 0.0

smart intelligent 9.2 0.25

hard difficult 8.77 1.0

happy cheerful 9.55 0.0

hard easy 0.95 0.0

fast rapid 8.75 0.25

happy glad 9.17 1.0

short long 1.23 0.25

stupid dumb 9.58 0.0

weird strange 8.93 0.0

...
```

Task 2 (in original-pairs.txt):

```
word1 word2 Similarity

novelty novelty 1.0

novelty roof 0.16666666666666666

novelty gratitude 0.14285714285714285

novelty short 0.125

novelty warm 0.0

novelty war 0.14285714285714285

novelty misfortune 0.125

novelty peculiar 0.0

novelty forgotten 0.0

novelty forty 0.09090909090909091

...
```

Task 3 (in original-pairs-hypernyms.txt):

```
word1 word2 Similarity1 hyp1 hyp2 Similarity2
saw saw 1.0 saying saying 1.0
saw bother 0.25 saying perturbation 0.1
saw break 0.3333333333333333 saying happening 0.125
saw lord 0.14285714285714285 saying happening 0.0
saw dog 0.2 saying canine 0.05555555555555555
saw wait 0.3333333333333333 saying pause 0.125
saw question 0.2 saying questioning 0.09090909090909091
saw middling 0.3333333333333333 saying commodity 0.09090909090909091
saw deep 0.3333333333333333 saying middle 0.125
saw eye 0.25 saying sense_organ 0.08333333333333333
...
```

Task 4 (in top.txt):

```
word1 word2 Similarity1
roof cap 1.0
short dead 1.0
short little 1.0
short poor 1.0
warm strong 1.0
evening even 1.0
peril danger 1.0
place laying 1.0
place lay 1.0
place put 1.0
```