Natural Language Engineering

Assignment2 Report

(Python Version 3.6)

Chao Li – 1802664

Yaowei Lyu - 1802697

Question1

a)

For S1 we need to add grammar:

- S -> VP
- P -> 'on'
- N -> 'block' | 'table'

For S2 we need to add grammar:

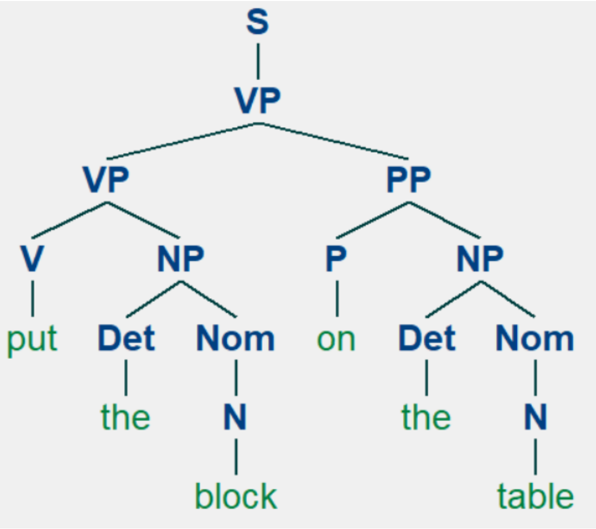
- NP -> PN
- PropN -> 'Bob'
- P -> 'in' | 'along'
- N -> 'river'

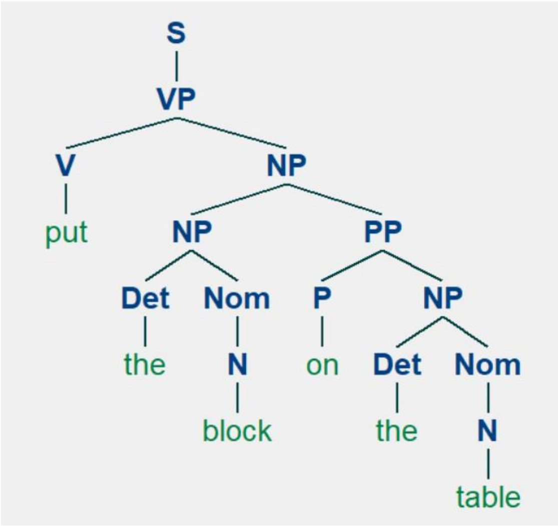
For S3 we need to add grammar:

- Adj -> 'furry'
- N -> 'dog'
- V -> 'chase'

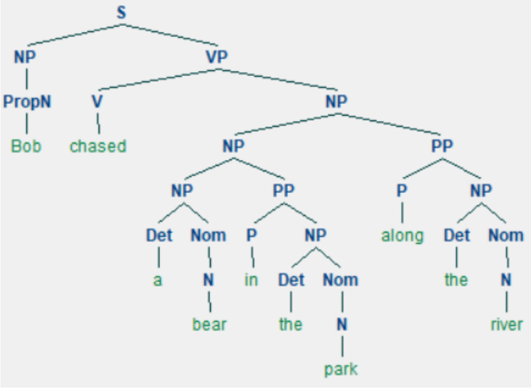
b)

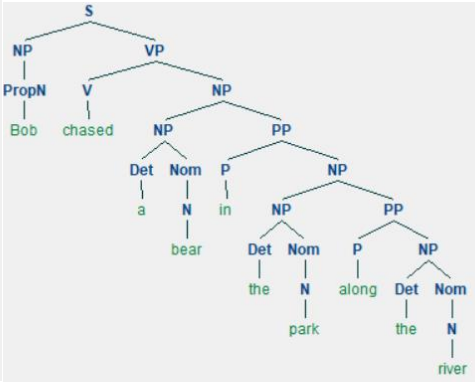
For S1 we get 2 derivations:

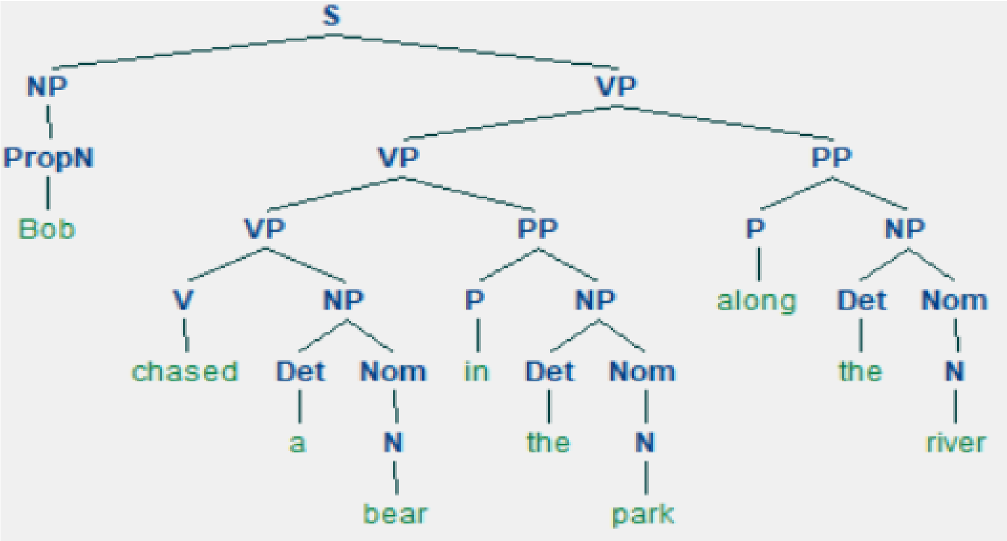


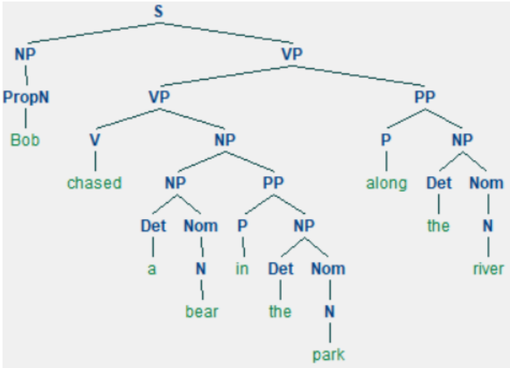


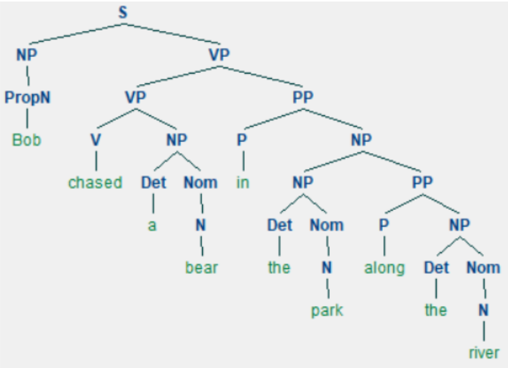
For S2 we get 5 derivations:



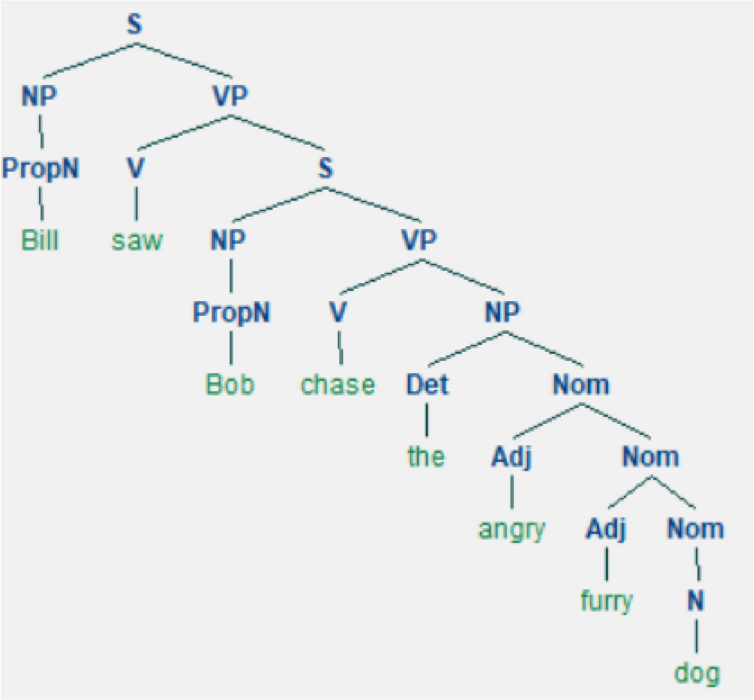








For S3 we get 1 derivation:



Question 2

a)

- **These sentences are not correct.**
- **For the first sentence:**
  1) As the initials of ‘bear’ and ’squirrel’ are not vowel, the determiner should be used ‘a’ instead of ‘an’.
  2) NNS should be used in ‘bear’ and ‘squirrel’. We should use make a generally reference in ‘bear’ and ‘squirrel’.
  3) grammatically correct equivalents: Bears eat squirrels
- **For the second sentence:**
  1) As ‘dogs’ is a NNS, we should use ‘eat’ rather than ‘eats’
  2) grammatically correct equivalents: The dogs eat

b) We use Bottom-Up Chart Parser and Earley Chart Parsing for S4 and S5

Use Bottom-Up Chart Parser for S4:

```
|.   An    .   bear  .   eat   .   an    . squirre .|
|[---------]         .         .         .         .| [0:1] 'An'
|.         [---------]         .         .         .| [1:2] 'bear'
|.         .         [---------]         .         .| [2:3] 'eat'
|.         .         .         [---------]         .| [3:4] 'an'
|.         .         .         .         [---------]| [4:5] 'squirrel'
|>         .         .         .         .         .| [0:0] Det -> * 'An'
|[---------]         .         .         .         .| [0:1] Det -> 'An' *
|>         .         .         .         .         .| [0:0] NP -> * Det Nom
|[--------->         .         .         .         .| [0:1] NP -> Det * Nom
|.         >         .         .         .         .| [1:1] N    -> * 'bear'
|.         [---------]         .         .         .| [1:2] N    -> 'bear' *
|.         >         .         .         .         .| [1:1] Nom -> * N
|.         [---------]         .         .         .| [1:2] Nom -> N *
|[-------------------]         .         .         .| [0:2] NP -> Det Nom *
|>         .         .         .         .         .| [0:0] S    -> * NP VP
|[------------------->         .         .         .| [0:2] S    -> NP * VP
|.         .         >         .         .         .| [2:2] V    -> * 'eat'
|.         .         [---------]         .         .| [2:3] V    -> 'eat' *
|.         .         >         .         .         .| [2:2] VP -> * V
|.         .         >         .         .         .| [2:2] VP -> * V NP
|.         .         >         .         .         .| [2:2] VP -> * V S
|.         .         [---------]         .         .| [2:3] VP -> V *
|.         .         [--------->         .         .| [2:3] VP -> V * NP
|.         .         [--------->         .         .| [2:3] VP -> V * S
|.         .         >         .         .         .| [2:2] VP -> * VP PP
|[-----------------------------]         .         .| [0:3] S    -> NP VP *
|.         .         [--------->         .         .| [2:3] VP -> VP * PP
|.         .         .         >         .         .| [3:3] Det -> * 'an'
|.         .         .         [---------]         .| [3:4] Det -> 'an' *
|.         .         .         >         .         .| [3:3] NP -> * Det Nom
|.         .         .         [--------->         .| [3:4] NP -> Det * Nom
|.         .         .         .         >         .| [4:4] N    -> * 'squirrel'
|.         .         .         .         [---------]| [4:5] N    -> 'squirrel' *
|.         .         .         .         >         .| [4:4] Nom -> * N
|.         .         .         .         [---------]| [4:5] Nom -> N *
|.         .         .         [-------------------]| [3:5] NP -> Det Nom *
|.         .         .         >         .         .| [3:3] S    -> * NP VP
|.         .         [-----------------------------]| [2:5] VP -> V NP *
|.         .         .         [------------------->| [3:5] S    -> NP * VP
|[=================================================]| [0:5] S    -> NP VP *
```

```
|.           .           [----------------------->| [2:5] VP -> VP * PP
(S
  (NP (Det An) (Nom (N bear)))
  (VP (V eat) (NP (Det an) (Nom (N squirrel)))))
```

Use Bottom-Up Chart Parser for S5:

```
|.    The    .   dogs    .   eats    .|
|[-----------]           .           .| [0:1] 'The'
|.           [-----------]           .| [1:2] 'dogs'
|.           .           [-----------]| [2:3] 'eats'
|>           .           .           .| [0:0] Det -> * 'The'
|[-----------]           .           .| [0:1] Det -> 'The' *
|>           .           .           .| [0:0] NP -> * Det Nom
|[----------->           .           .| [0:1] NP -> Det * Nom
|.           >           .           .| [1:1] NNS -> * 'dogs'
|.           >           .           .| [1:1] N   -> * 'dogs'
|.           [-----------]           .| [1:2] NNS -> 'dogs' *
|.           [-----------]           .| [1:2] N   -> 'dogs' *
|.           >           .           .| [1:1] Nom -> * N
|.           [-----------]           .| [1:2] Nom -> N *
|[-----------------------]           .| [0:2] NP -> Det Nom *
|>           .           .           .| [0:0] S   -> * NP VP
|[----------------------->           .| [0:2] S   -> NP * VP
|.           >           .           .| [1:1] NP -> * NNS
|.           [-----------]           .| [1:2] NP -> NNS *
|.           >           .           .| [1:1] S   -> * NP VP
|.           [----------->           .| [1:2] S   -> NP * VP
|.           .           >           .| [2:2] V   -> * 'eats'
|.           .           [-----------]| [2:3] V   -> 'eats' *
|.           .           >           .| [2:2] VP -> * V
|.           .           >           .| [2:2] VP -> * V NP
|.           .           >           .| [2:2] VP -> * V S
|.           .           [-----------]| [2:3] VP -> V *
|.           .           [----------->| [2:3] VP -> V * NP
|.           .           [----------->| [2:3] VP -> V * S
|.           .           >           .| [2:2] VP -> * VP PP
|[===================================]| [0:3] S   -> NP VP *
|.           [-----------------------]| [1:3] S   -> NP VP *
|.           .           [----------->| [2:3] VP -> VP * PP
(S (NP (Det The) (Nom (N dogs))) (VP (V eats)))
```

Use Earley Chart Parser for S4:

```
|.  An   . bear  .  eat  .  an   .squirre.|
|[-------]       .       .       .       .| [0:1] 'An'
|.       [-------]       .       .       .| [1:2] 'bear'
|.       .       [-------]       .       .| [2:3] 'eat'
|.       .       .       [-------]       .| [3:4] 'an'
|.       .       .       .       [-------]| [4:5] 'squirrel'
|>       .       .       .       .       .| [0:0] S    -> * NP VP
|>       .       .       .       .       .| [0:0] NP -> * Det Nom
|>       .       .       .       .       .| [0:0] NP -> * PropN
|>       .       .       .       .       .| [0:0] NP -> * NNS
|>       .       .       .       .       .| [0:0] Det -> * 'An'
|[-------]       .       .       .       .| [0:1] Det -> 'An' *
|[------->       .       .       .       .| [0:1] NP -> Det * Nom
|.       >       .       .       .       .| [1:1] Nom -> * N
|.       >       .       .       .       .| [1:1] N    -> * 'bear'
|.       [-------]       .       .       .| [1:2] N    -> 'bear' *
|.       [-------]       .       .       .| [1:2] Nom -> N *
|[---------------]       .       .       .| [0:2] NP -> Det Nom *
|[--------------->       .       .       .| [0:2] S    -> NP * VP
|.       .       >       .       .       .| [2:2] VP -> * VP PP
|.       .       >       .       .       .| [2:2] VP -> * V
|.       .       >       .       .       .| [2:2] VP -> * V NP
|.       .       >       .       .       .| [2:2] VP -> * V S
|.       .       >       .       .       .| [2:2] V    -> * 'eat'
|.       .       [-------]       .       .| [2:3] V    -> 'eat' *
|.       .       [-------]       .       .| [2:3] VP -> V *
|.       .       [------->       .       .| [2:3] VP -> V * NP
|.       .       [------->       .       .| [2:3] VP -> V * S
|.       .       .       >       .       .| [3:3] S    -> * NP VP
|.       .       .       >       .       .| [3:3] NP -> * Det Nom
|.       .       .       >       .       .| [3:3] NP -> * PropN
|.       .       .       >       .       .| [3:3] NP -> * NNS
|.       .       .       >       .       .| [3:3] Det -> * 'an'
|[-----------------------]       .       .| [0:3] S    -> NP VP *
|.       .       [------->       .       .| [2:3] VP -> VP * PP
|.       .       .       [-------]       .| [3:4] Det -> 'an' *
|.       .       .       [------->       .| [3:4] NP -> Det * Nom
|.       .       .       .       >       .| [4:4] Nom -> * N
|.       .       .       .       >       .| [4:4] N    -> * 'squirrel'
|.       .       .       .       [-------]| [4:5] N    -> 'squirrel' *
|.       .       .       .       [-------]| [4:5] Nom -> N *
|.       .       .       [---------------]| [3:5] NP -> Det Nom *
|.       .       [-----------------------]| [2:5] VP -> V NP *
|.           .           .           [-------------->| [3:5] S    -> NP * VP
|.           .           .           .           .           >| [5:5] VP -> * VP PP
|.           .           .           .           .           >| [5:5] VP -> * V
|.           .           .           .           .           >| [5:5] VP -> * V NP
|.           .           .           .           .           >| [5:5] VP -> * V S
|[===========================================================]| [0:5] S    -> NP VP *
|.           .           [----------------------->| [2:5] VP -> VP * PP
(S
  (NP (Det An) (Nom (N bear)))
  (VP (V eat) (NP (Det an) (Nom (N squirrel)))))
```

Use Earley Chart Parser for S5:

```
|.    The    .    dogs    .    eats    .|
|[-----------]           .           .| [0:1] 'The'
|.           [-----------]           .| [1:2] 'dogs'
|.           .           [-----------]| [2:3] 'eats'
|>           .           .           .| [0:0] S    -> * NP VP
|>           .           .           .| [0:0] NP -> * Det Nom
|>           .           .           .| [0:0] NP -> * PropN
|>           .           .           .| [0:0] NP -> * NNS
|>           .           .           .| [0:0] Det -> * 'The'
|[-----------]           .           .| [0:1] Det -> 'The' *
|[----------->           .           .| [0:1] NP -> Det * Nom
|.           >           .           .| [1:1] Nom -> * N
|.           >           .           .| [1:1] N    -> * 'dogs'
|.           [-----------]           .| [1:2] N    -> 'dogs' *
|.           [-----------]           .| [1:2] Nom -> N *
|[-----------------------]           .| [0:2] NP -> Det Nom *
|[----------------------->           .| [0:2] S    -> NP * VP
|.           .           >           .| [2:2] VP -> * VP PP
|.           .           >           .| [2:2] VP -> * V
|.           .           >           .| [2:2] VP -> * V NP
|.           .           >           .| [2:2] VP -> * V S
|.           .           >           .| [2:2] V    -> * 'eats'
|.           .           [-----------]| [2:3] V    -> 'eats' *
|.           .           [-----------]| [2:3] VP -> V *
|.           .           [----------->| [2:3] VP -> V * NP
|.           .           [----------->| [2:3] VP -> V * S
|.           .           .           >| [3:3] S    -> * NP VP
|.           .           .           >| [3:3] NP -> * Det Nom
|.           .           .           >| [3:3] NP -> * PropN
|.           .           .           >| [3:3] NP -> * NNS
|[===================================]| [0:3] S    -> NP VP *
```

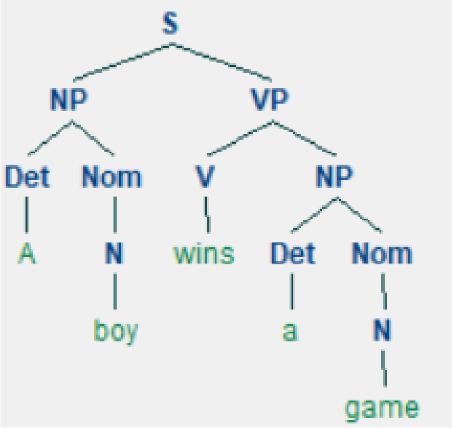
```
|.                    .                    [----------->| [2:3] VP -> VP * PP
(S (NP (Det The) (Nom (N dogs))) (VP (V eats)))
```

Q: Explain why the parsers are correct or incorrect:

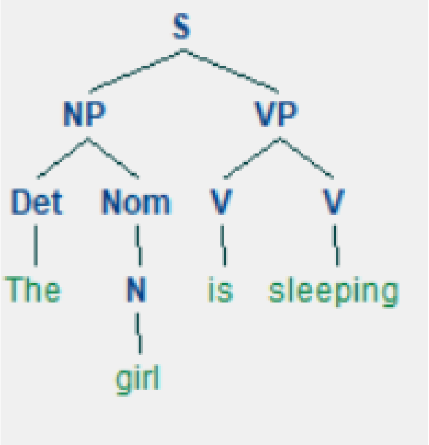
A: The parsers are incorrect, because the grammars are too simple. Although they can grammatically recognize the sentences, they fail to parser the sentence by correct singular-plural pair, tense and vowel.

c)

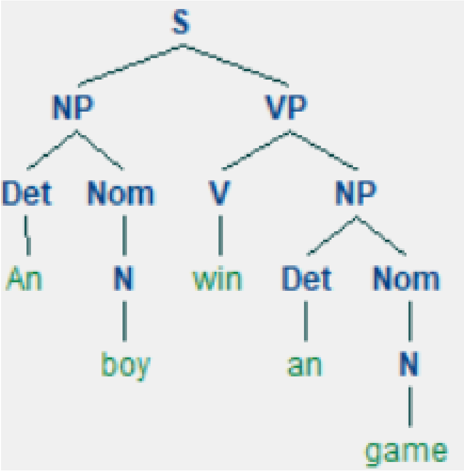
Correct sentence1: A boy wins a game



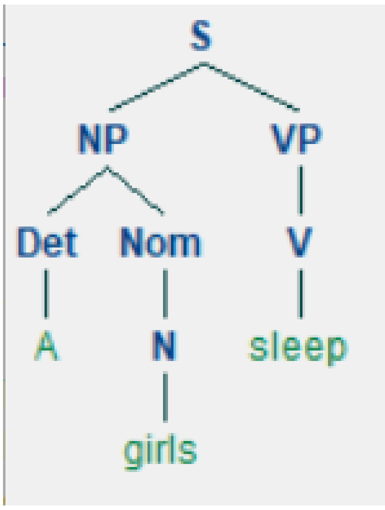
Correct sentence 2: The girl is sleeping



Incorrect sentence 1: An boy win an game



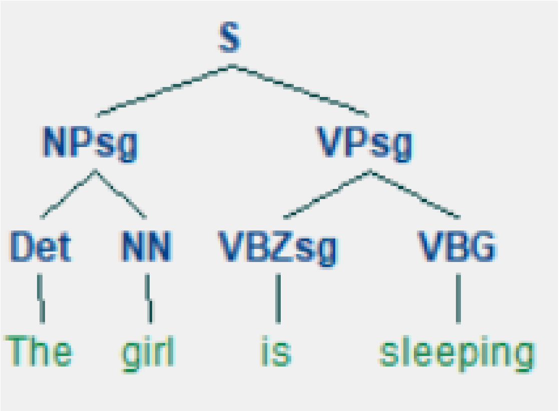
Incorrect sentence2: A girls sleep



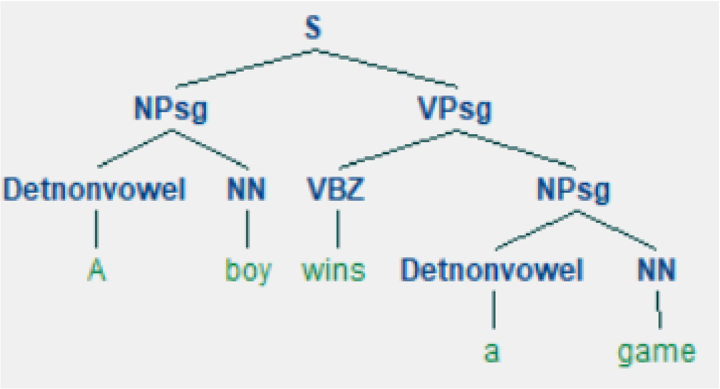
Q: How would you have to change this grammar to prevent these sentences from being parsed?

A: For the incorrect sentence, we can make the grammar more complex and more detailed. For example, instead of Det -> ‘a’ | ‘an’, we can turn them into Detnonvowel -> 'a', Detnvowel -> ‘an’. In this case, if the first letter of a noun is vowel, only ‘an’ can connect with to form a grammar. Similarly, we can make grammar of tense or pronoun more detailed. The following are the revised sentence for the incorrect sentence parsed by more detailed grammar:

Incorrect sentence ‘A girls sleep’ after revised:



Incorrect sentence ‘An boy win an game’ after revised:

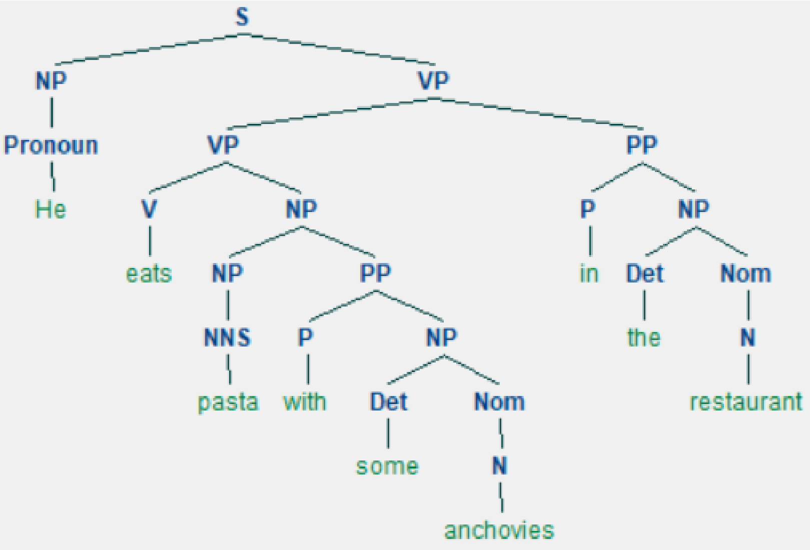


Question3

a) Both S6 and S7 have more than 1 interpretations.

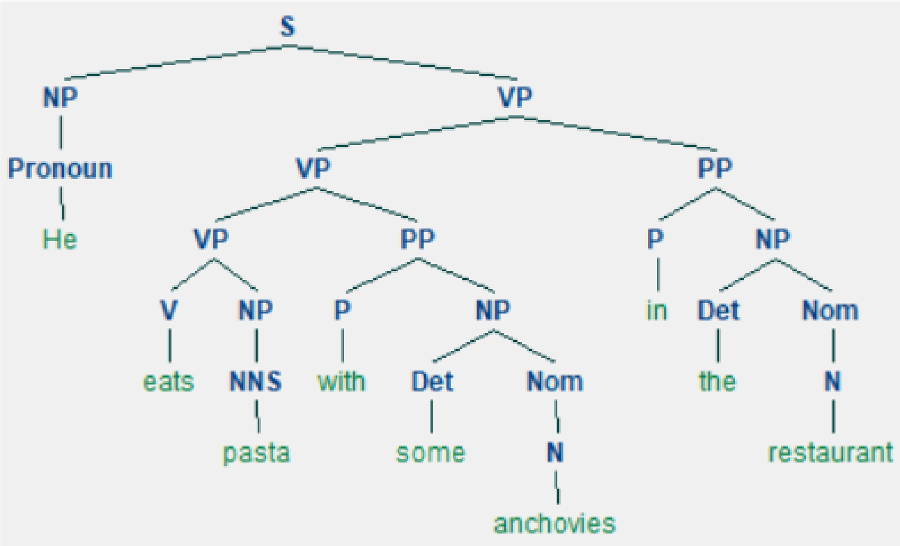
For S6, we get 5 derivations:

Interpretation1: He is in the restaurant, and eats pasta. Anchovies are in the pasta



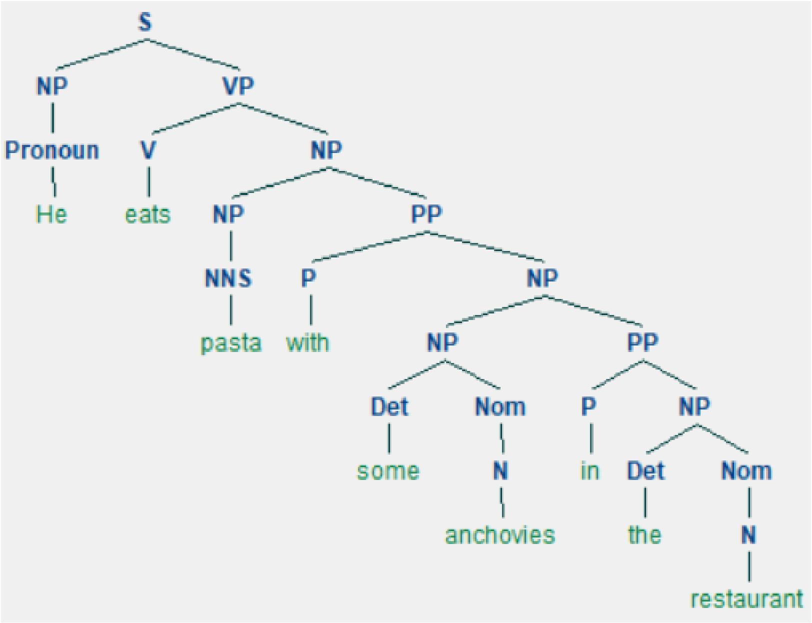
Interpretation1

Interpretation2: He is in the restaurant, and eats past, and some anchovies. (Anchovies are not in the pasta)



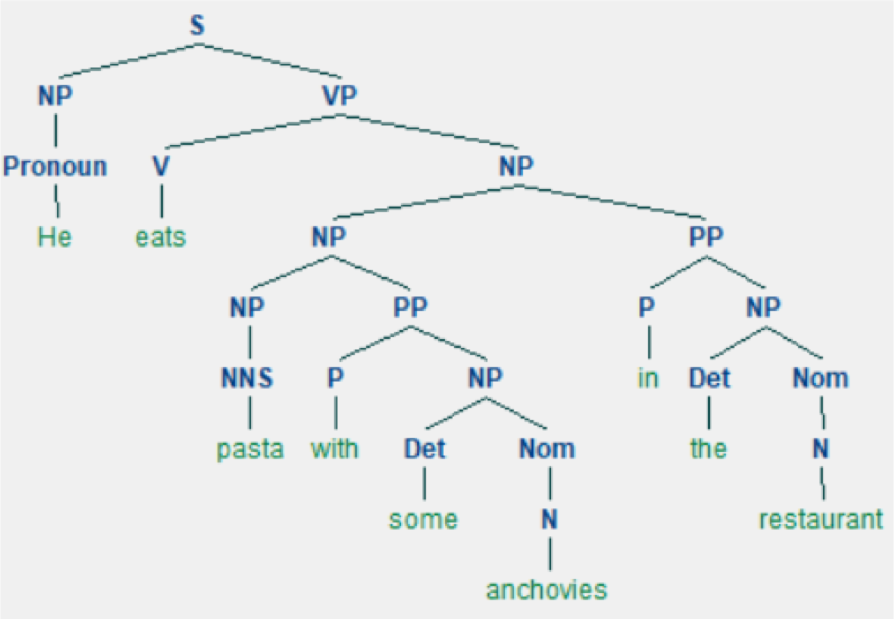
Interpretation2

Interpretation3: He eats pasta. Anchovies are in the pasta. And anchovies are from the restaurant.



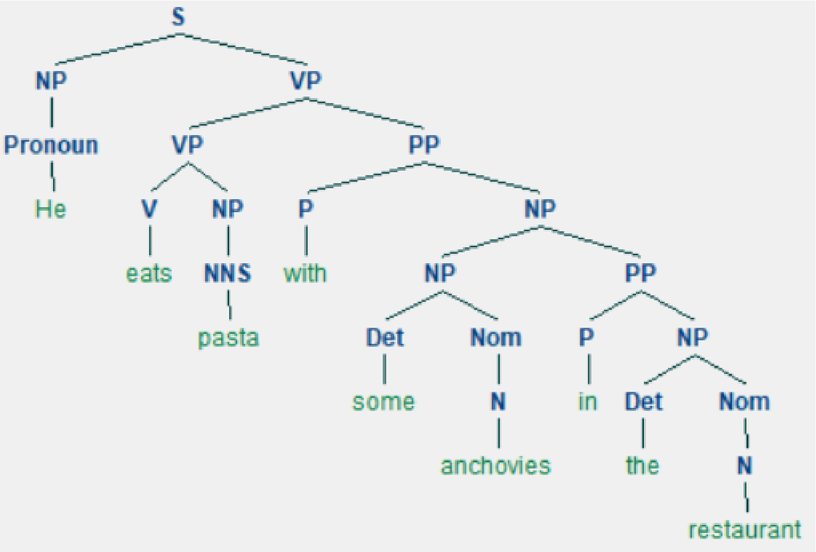
Interpretation3

Interpretation4: He eats pasta. Anchovies are in the pasta. The pasta is from the restaurant.



Interpretation4

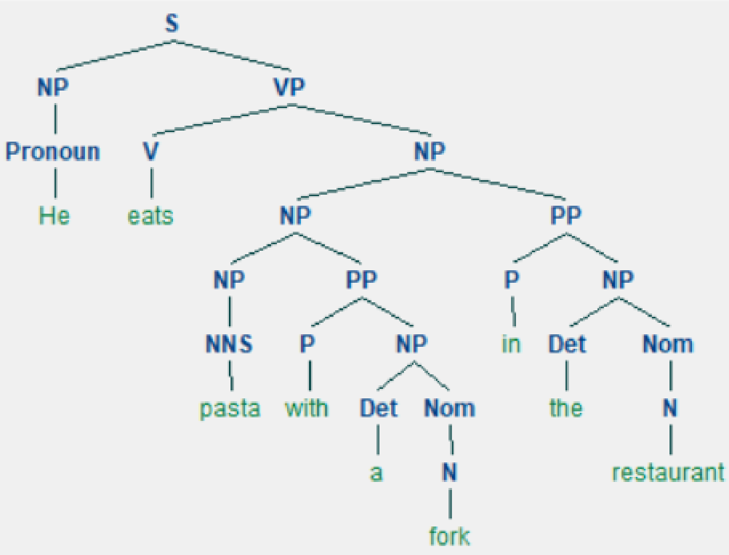
Interpretation5: He eats pasta and some anchovies (Anchovies are not in pasta). Anchovies are from the restaurant.



Interpretation5

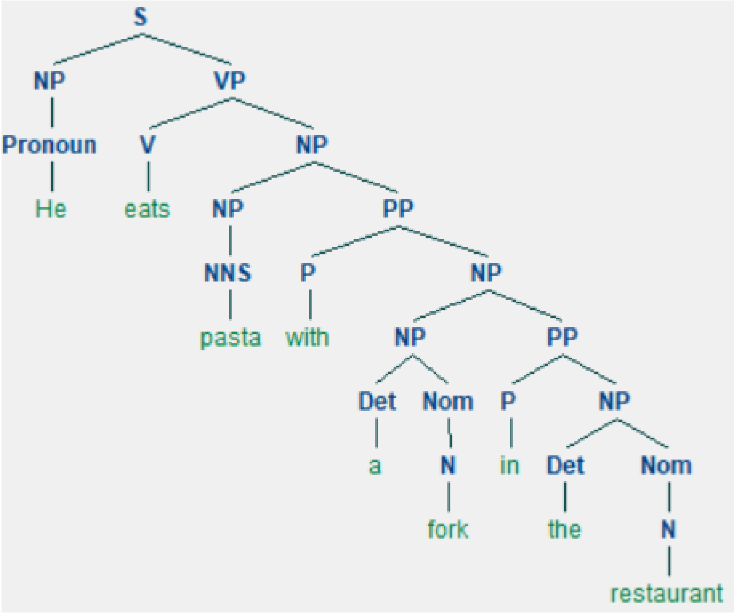
For S7, we get 5 derivations:

Interpretation1: He eats pasta. A fork is in pasta. Pasta is from the restaurant.



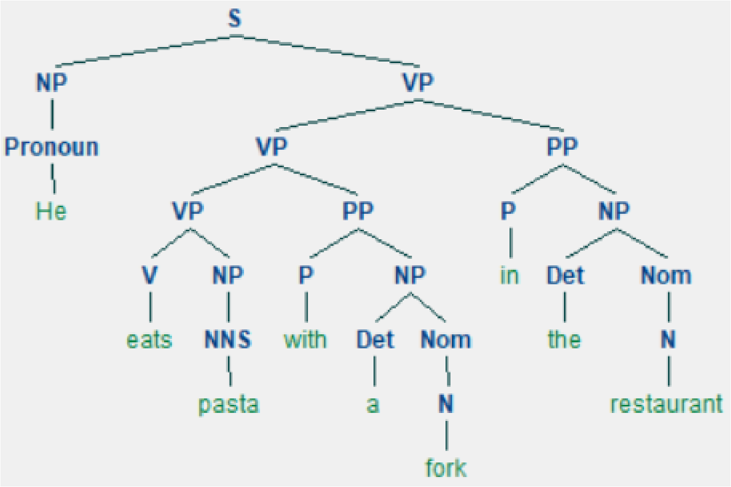
Interpretation1

Interpretation2: He eats pasta. A fork is in pasta. And the fork is from the restaurant.



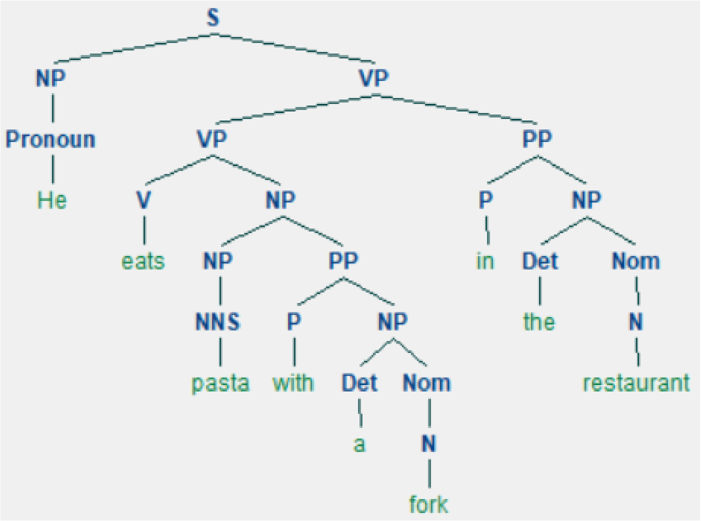
Interpretation2

Interpretation3: He uses a fork to eat pasta and he is in the restaurant



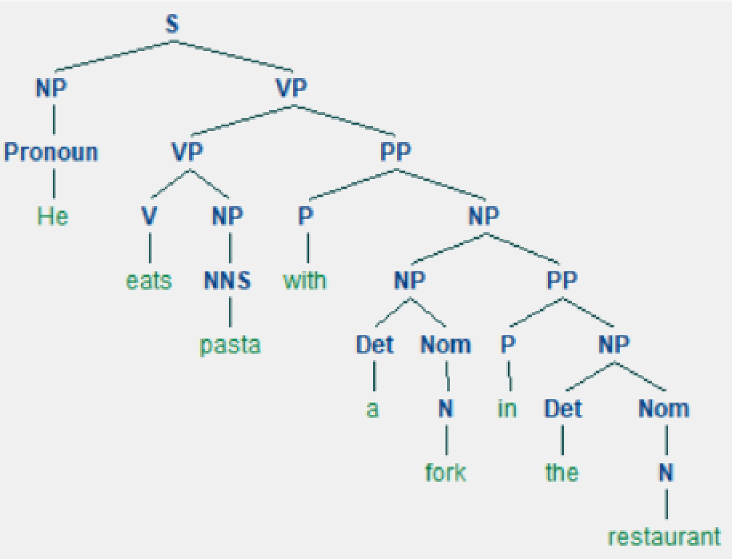
Interpretation3

Interpretation4: He eats pasta, and he is in the restaurant. A fork is in pasta.



Interpretation4

Interpretation5: **He use a fork to eat pasta. The fork is from the restaurant.**



Interpretation5

b)

S6: He eats pasta with some anchovies in the restaurant - Shift-reduce:

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 [ Pronoun * eats pasta with some anchovies in the restaurant]
S [ Pronoun 'eats' * pasta with some anchovies in the restaurant]
R [ Pronoun V * pasta with some anchovies in the restaurant]
S [ Pronoun V 'pasta' * with some anchovies in the restaurant]
R [ Pronoun V NNS * with some anchovies in the restaurant]
S [ Pronoun V NNS 'with' * some anchovies in the restaurant]
```

```
  R [ Pronoun V NNS P * some anchovies in the restaurant]
  S [ Pronoun V NNS P 'some' * anchovies in the restaurant]
  R [ Pronoun V NNS P Det * anchovies in the restaurant]
  S [ Pronoun V NNS P Det 'anchovies' * in the restaurant]
  R [ Pronoun V NNS P Det N * in the restaurant]
  R [ Pronoun V NNS P NP * in the restaurant]
  R [ Pronoun V NNS PP * in the restaurant]
  R [ Pronoun V NP * in the restaurant]
  R [ Pronoun VP * in the restaurant]
  R [ S * in the restaurant]
  S [ S 'in' * the restaurant]
  R [ S P * the restaurant]
  S [ S P 'the' * restaurant]
  R [ S P Det * restaurant]
  S [ S P Det 'restaurant' * ]
  R [ S P Det N * ]
  R [ S P NP * ]
  R [ S PP * ]
  R [ S * ]
(S
  (S
    (Pronoun He)
    (VP
      (V eats)
      (NP (NNS pasta) (PP (P with) (NP (Det some) (N anchovies))))))
  (PP (P in) (NP (Det the) (N restaurant))))
```

S6: He eats pasta with some anchovies in the restaurant – Earley Chart Parser:

```
|. He .eats.past.with.some.anch. in .the .rest.|
|[----]    .    .    .    .    .    .    .    .| [0:1] 'He'
|.    [----]    .    .    .    .    .    .    .| [1:2] 'eats'
|.    .    [----]    .    .    .    .    .    .| [2:3] 'pasta'
|.    .    .    [----]    .    .    .    .    .| [3:4] 'with'
|.    .    .    .    [----]    .    .    .    .| [4:5] 'some'
|.    .    .    .    .    [----]    .    .    .| [5:6] 'anchovies'
|.    .    .    .    .    .    [----]    .    .| [6:7] 'in'
|.    .    .    .    .    .    .    [----]    .| [7:8] 'the'
|.    .    .    .    .    .    .    .    [----]| [8:9] 'restaurant'
|>    .    .    .    .    .    .    .    .    .| [0:0] S  -> * NP VP
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * Det Nom
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * NP PP
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * NNS
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * Pronoun
|>    .    .    .    .    .    .    .    .    .| [0:0] Pronoun -> * 'He'
|[----]    .    .    .    .    .    .    .    .| [0:1] Pronoun -> 'He' *
|[----]    .    .    .    .    .    .    .    .| [0:1] NP -> Pronoun *
|[---->    .    .    .    .    .    .    .    .| [0:1] S  -> NP * VP
|[---->    .    .    .    .    .    .    .    .| [0:1] NP -> NP * PP
|.    >    .    .    .    .    .    .    .    .| [1:1] PP -> * P NP
|.    >    .    .    .    .    .    .    .    .| [1:1] VP -> * VP PP
|.    >    .    .    .    .    .    .    .    .| [1:1] VP -> * V NP
|.    >    .    .    .    .    .    .    .    .| [1:1] VP -> * V S
|.    >    .    .    .    .    .    .    .    .| [1:1] V  -> * 'eats'
|.    [----]    .    .    .    .    .    .    .| [1:2] V  -> 'eats' *
|.    [---->    .    .    .    .    .    .    .| [1:2] VP -> V * NP
|.    [---->    .    .    .    .    .    .    .| [1:2] VP -> V * S
|.    .    >    .    .    .    .    .    .    .| [2:2] S  -> * NP VP
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * Det Nom
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * NP PP
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * NNS
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * Pronoun
|.    .    >    .    .    .    .    .    .    .| [2:2] NNS -> * 'pasta'
|.    .    [----]    .    .    .    .    .    .| [2:3] NNS -> 'pasta' *
|.    .    [----]    .    .    .    .    .    .| [2:3] NP -> NNS *
|.    [---------]    .    .    .    .    .    .| [1:3] VP -> V NP *
|.    .    [---->    .    .    .    .    .    .| [2:3] S  -> NP * VP
|.    .    [---->    .    .    .    .    .    .| [2:3] NP -> NP * PP
|.    .    .    >    .    .    .    .    .    .| [3:3] PP -> * P NP
|.    .    .    >    .    .    .    .    .    .| [3:3] P  -> * 'with'
|.    .    .    >    .    .    .    .    .    .| [3:3] VP -> * VP PP
|.    .    .    >    .    .    .    .    .    .| [3:3] VP -> * V NP
|.    .    .    >    .    .    .    .    .    .| [3:3] VP -> * V S
|[--------------]    .    .    .    .    .    .| [0:3] S   -> NP VP *
|.    [--------->    .    .    .    .    .    .| [1:3] VP -> VP * PP
|.    .    .    [----]    .    .    .    .    .| [3:4] P   -> 'with' *
|.    .    .    [---->    .    .    .    .    .| [3:4] PP -> P * NP
|.    .    .    .    >    .    .    .    .    .| [4:4] NP -> * Det Nom
|.    .    .    .    >    .    .    .    .    .| [4:4] NP -> * NP PP
|.    .    .    .    >    .    .    .    .    .| [4:4] NP -> * NNS
|.    .    .    .    >    .    .    .    .    .| [4:4] NP -> * Pronoun
|.    .    .    .    >    .    .    .    .    .| [4:4] Det -> * 'some'
|.    .    .    .    [----]    .    .    .    .| [4:5] Det -> 'some' *
|.    .    .    .    [---->    .    .    .    .| [4:5] NP -> Det * Nom
|.    .    .    .    .    >    .    .    .    .| [5:5] Nom -> * N
|.    .    .    .    .    >    .    .    .    .| [5:5] N   -> * 'anchovies'
|.    .    .    .    .    [----]    .    .    .| [5:6] N   -> 'anchovies' *
|.    .    .    .    .    [----]    .    .    .| [5:6] Nom -> N *
|.    .    .    .    [---------]    .    .    .| [4:6] NP -> Det Nom *
|.    .    .    [--------------]    .    .    .| [3:6] PP -> P NP *
|.    .    .    .    [--------->    .    .    .| [4:6] NP -> NP * PP
|.    .    .    .    .    .    >    .    .    .| [6:6] PP -> * P NP
|.    .    .    .    .    .    >    .    .    .| [6:6] P   -> * 'in'
|.    .    [-------------------]    .    .    .| [2:6] NP -> NP PP *
|.    [------------------------]    .    .    .| [1:6] VP -> VP PP *
|[-----------------------------]    .    .    .| [0:6] S   -> NP VP *
|.    [------------------------>    .    .    .| [1:6] VP -> VP * PP
|.    [------------------------]    .    .    .| [1:6] VP -> V NP *
|.    .    [------------------->    .    .    .| [2:6] S   -> NP * VP
|.    .    [------------------->    .    .    .| [2:6] NP -> NP * PP
|.    .    .    .    .    .    >    .    .    .| [6:6] VP -> * VP PP
|.    .    .    .    .    .    >    .    .    .| [6:6] VP -> * V NP
|.    .    .    .    .    .    >    .    .    .| [6:6] VP -> * V S
|[-----------------------------]    .    .    .| [0:6] S   -> NP VP *
|.    [------------------------>    .    .    .| [1:6] VP -> VP * PP
|.    .    .    .    .    .    [----]    .    .| [6:7] P   -> 'in' *
|.    .    .    .    .    .    [---->    .    .| [6:7] PP -> P * NP
|.    .    .    .    .    .    .    >    .    .| [7:7] NP -> * Det Nom
|.    .    .    .    .    .    .    >    .    .| [7:7] NP -> * NP PP
|.    .    .    .    .    .    .    >    .    .| [7:7] NP -> * NNS
|.    .    .    .    .    .    .    >    .    .| [7:7] NP -> * Pronoun
|.    .    .    .    .    .    .    >    .    .| [7:7] Det -> * 'the'
|.    .    .    .    .    .    .    [----]    .| [7:8] Det -> 'the' *
|.    .    .    .    .    .    .    [---->    .| [7:8] NP -> Det * Nom
|.    .    .    .    .    .    .    .    >    .| [8:8] Nom -> * N
|.    .    .    .    .    .    .    .    >    .| [8:8] N   -> * 'restaurant'
|.    .    .    .    .    .    .    .    [----]| [8:9] N   -> 'restaurant' *
|.    .    .    .    .    .    .    .    [----]| [8:9] Nom -> N *
|.    .    .    .    .    .    .    [---------]| [7:9] NP -> Det Nom *
|.    .    .    .    .    .    [--------------]| [6:9] PP -> P NP *
|.    .    .    .    .    .    .    [--------->| [7:9] NP -> NP * PP
|.    .    .    .    .    .    .    .    .    >| [9:9] PP -> * P NP
|.    .    .    .    [------------------------]| [4:9] NP -> NP PP *
|.    [---------------------------------------]| [1:9] VP -> VP PP *
|.    .    [----------------------------------]| [2:9] NP -> NP PP *
|.    [---------------------------------------]| [1:9] VP -> V NP *
|.    .    [---------------------------------->| [2:9] S   -> NP * VP
|.    .    [---------------------------------->| [2:9] NP -> NP * PP
|.    .    .    .    .    .    .    .    .    >| [9:9] VP -> * VP PP
|.    .    .    .    .    .    .    .    .    >| [9:9] VP -> * V NP
|.    .    .    .    .    .    .    .    .    >| [9:9] VP -> * V S
|[============================================]| [0:9] S   -> NP VP *
|.    [--------------------------------------->| [1:9] VP -> VP * PP
|[============================================]| [0:9] S   -> NP VP *
|.    [--------------------------------------->| [1:9] VP -> VP * PP
|.    .    .    [-----------------------------]| [3:9] PP -> P NP *
|.    .    .    .    [----------------------->| [4:9] NP -> NP * PP
|.    .    [----------------------------------]| [2:9] NP -> NP PP *
|.    [---------------------------------------]| [1:9] VP -> VP PP *
(S
  (NP (Pronoun He))
  (VP
    (V eats)
    (NP
      (NP
        (NP (NNS pasta))
        (PP (P with) (NP (Det some) (Nom (N anchovies)))))
      (PP (P in) (NP (Det the) (Nom (N restaurant)))))))
(S
  (NP (Pronoun He))
  (VP
    (V eats)
    (NP
      (NP (NNS pasta))
      (PP
        (P with)
        (NP
          (NP (Det some) (Nom (N anchovies)))
          (PP (P in) (NP (Det the) (Nom (N restaurant)))))))))
(S
   (NP (Pronoun He))
   (VP
      (VP
         (VP (V eats) (NP (NNS pasta)))
         (PP (P with) (NP (Det some) (Nom (N anchovies)))))
      (PP (P in) (NP (Det the) (Nom (N restaurant))))))
(S
   (NP (Pronoun He))
   (VP
      (VP
         (V eats)
         (NP
            (NP (NNS pasta))
            (PP (P with) (NP (Det some) (Nom (N anchovies))))))
      (PP (P in) (NP (Det the) (Nom (N restaurant))))))
(S
   (NP (Pronoun He))
   (VP
      (VP (V eats) (NP (NNS pasta)))
      (PP
         (P with)
         (NP
            (NP (Det some) (Nom (N anchovies)))
            (PP (P in) (NP (Det the) (Nom (N restaurant))))))))
```

S7: He eats pasta with a fork in the restaurant – Shift Reduce:

```
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 [ Pronoun * eats pasta with a fork in the restaurant]
    S [ Pronoun 'eats' * pasta with a fork in the restaurant]
    R [ Pronoun V * pasta with a fork in the restaurant]
    S [ Pronoun V 'pasta' * with a fork in the restaurant]
    R [ Pronoun V NNS * with a fork in the restaurant]
    R [ Pronoun 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 NP * in the restaurant]
    R [ S PP * in the restaurant]
    R [ S * in the restaurant]
    S [ S 'in' * the restaurant]
    R [ S P * the restaurant]
    S [ S P 'the' * restaurant]
    R [ S P Det * restaurant]
    S [ S P Det 'restaurant' * ]
    R [ S P Det N * ]
    R [ S P NP * ]
    R [ S PP * ]
    R [ S * ]
(S
    (S
        (S (Pronoun He) (VP (V eats) (NNS pasta)))
        (PP (P with) (NP (Det a) (N fork))))
    (PP (P in) (NP (Det the) (N restaurant))))
```

S7: He eats pasta with a fork in the restaurant – Earley Chart Parser:

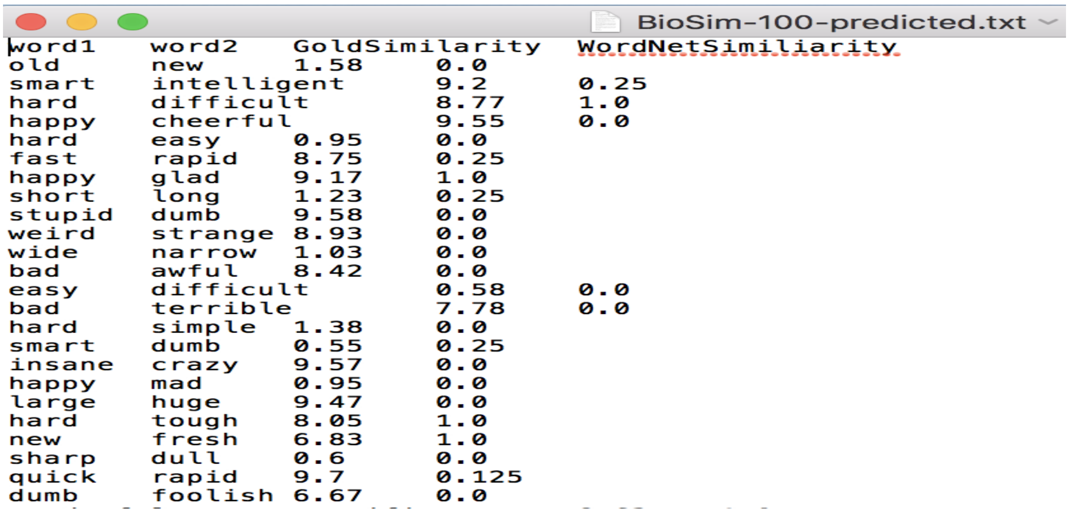
```
|. He .eats.past.with. a  .fork. in .the .rest.|
|[----]    .    .    .    .    .    .    .    .| [0:1] 'He'
|.    [----]    .    .    .    .    .    .    .| [1:2] 'eats'
|.    .    [----]    .    .    .    .    .    .| [2:3] 'pasta'
|.    .    .    [----]    .    .    .    .    .| [3:4] 'with'
|.    .    .    .    [----]    .    .    .    .| [4:5] 'a'
|.    .    .    .    .    [----]    .    .    .| [5:6] 'fork'
|.    .    .    .    .    .    [----]    .    .| [6:7] 'in'
|.    .    .    .    .    .    .    [----]    .| [7:8] 'the'
|.    .    .    .    .    .    .    .    [----]| [8:9] 'restaurant'
|>    .    .    .    .    .    .    .    .    .| [0:0] S    -> * NP VP
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * Det Nom
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * NP PP
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * NNS
|>    .    .    .    .    .    .    .    .    .| [0:0] NP -> * Pronoun
|>    .    .    .    .    .    .    .    .    .| [0:0] Pronoun -> * 'He'
|[----]    .    .    .    .    .    .    .    .| [0:1] Pronoun -> 'He' *
|[----]    .    .    .    .    .    .    .    .| [0:1] NP -> Pronoun *
|[---->    .    .    .    .    .    .    .    .| [0:1] S    -> NP * VP
|[---->    .    .    .    .    .    .    .    .| [0:1] NP -> NP * PP
|.    >    .    .    .    .    .    .    .    .| [1:1] PP -> * P NP
|.    >    .    .    .    .    .    .    .    .| [1:1] VP -> * VP PP
|.    >    .    .    .    .    .    .    .    .| [1:1] VP -> * V NP
|.    >    .    .    .    .    .    .    .    .| [1:1] VP -> * V S
|.    >    .    .    .    .    .    .    .    .| [1:1] V    -> * 'eats'
|.    [----]    .    .    .    .    .    .    .| [1:2] V    -> 'eats' *
|.    [---->    .    .    .    .    .    .    .| [1:2] VP -> V * NP
|.    [---->    .    .    .    .    .    .    .| [1:2] VP -> V * S
|.    .    >    .    .    .    .    .    .    .| [2:2] S    -> * NP VP
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * Det Nom
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * NP PP
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * NNS
|.    .    >    .    .    .    .    .    .    .| [2:2] NP -> * Pronoun
|.    .    >    .    .    .    .    .    .    .| [2:2] NNS -> * 'pasta'
|.    .    [----]    .    .    .    .    .    .| [2:3] NNS -> 'pasta' *
|.    .    [----]    .    .    .    .    .    .| [2:3] NP -> NNS *
|.    [---------]    .    .    .    .    .    .| [1:3] VP -> V NP *
|.    .    [---->    .    .    .    .    .    .| [2:3] S    -> NP * VP
|.    .    [---->    .    .    .    .    .    .| [2:3] NP -> NP * PP
|.    .    .    >    .    .    .    .    .    .| [3:3] PP -> * P NP
|.    .    .    >    .    .    .    .    .    .| [3:3] P    -> * 'with'
|.    .    .    >    .    .    .    .    .    .| [3:3] VP -> * VP PP
|.   .   .   >   .   .   .   .   .   .| [3:3] VP -> * V NP
|.   .   .   >   .   .   .   .   .   .| [3:3] VP -> * V S
|[-----------]   .   .   .   .   .   .| [0:3] S    -> NP VP *
|.   [------->   .   .   .   .   .   .| [1:3] VP -> VP * PP
|.   .   .   [---]   .   .   .   .   .| [3:4] P    -> 'with' *
|.   .   .   [--->   .   .   .   .   .| [3:4] PP -> P * NP
|.   .   .   .   >   .   .   .   .   .| [4:4] NP -> * Det Nom
|.   .   .   .   >   .   .   .   .   .| [4:4] NP -> * NP PP
|.   .   .   .   >   .   .   .   .   .| [4:4] NP -> * NNS
|.   .   .   .   >   .   .   .   .   .| [4:4] NP -> * Pronoun
|.   .   .   .   >   .   .   .   .   .| [4:4] Det -> * 'a'
|.   .   .   .   [---]   .   .   .   .| [4:5] Det -> 'a' *
|.   .   .   .   [--->   .   .   .   .| [4:5] NP -> Det * Nom
|.   .   .   .   .   >   .   .   .   .| [5:5] Nom -> * N
|.   .   .   .   .   >   .   .   .   .| [5:5] N    -> * 'fork'
|.   .   .   .   .   [---]   .   .   .| [5:6] N    -> 'fork' *
|.   .   .   .   .   [---]   .   .   .| [5:6] Nom -> N *
|.   .   .   .   [-------]   .   .   .| [4:6] NP -> Det Nom *
|.   .   .   [-----------]   .   .   .| [3:6] PP -> P NP *
|.   .   .   .   [------->   .   .   .| [4:6] NP -> NP * PP
|.   .   .   .   .   .   >   .   .   .| [6:6] PP -> * P NP
|.   .   .   .   .   .   >   .   .   .| [6:6] P    -> * 'in'
|.   .   [---------------]   .   .   .| [2:6] NP -> NP PP *
|.   [-------------------]   .   .   .| [1:6] VP -> VP PP *
|[-----------------------]   .   .   .| [0:6] S    -> NP VP *
|.   [------------------->   .   .   .| [1:6] VP -> VP * PP
|.   [-------------------]   .   .   .| [1:6] VP -> V NP *
|.   .   [--------------->   .   .   .| [2:6] S    -> NP * VP
|.   .   [--------------->   .   .   .| [2:6] NP -> NP * PP
|.   .   .   .   .   .   >   .   .   .| [6:6] VP -> * VP PP
|.   .   .   .   .   .   >   .   .   .| [6:6] VP -> * V NP
|.   .   .   .   .   .   >   .   .   .| [6:6] VP -> * V S
|[-----------------------]   .   .   .| [0:6] S    -> NP VP *
|.   [------------------->   .   .   .| [1:6] VP -> VP * PP
|.   .   .   .   .   .   [---]   .   .| [6:7] P    -> 'in' *
|.   .   .   .   .   .   [--->   .   .| [6:7] PP -> P * NP
|.   .   .   .   .   .   .   >   .   .| [7:7] NP -> * Det Nom
|.   .   .   .   .   .   .   >   .   .| [7:7] NP -> * NP PP
|.   .   .   .   .   .   .   >   .   .| [7:7] NP -> * NNS
|.   .   .   .   .   .   .   >   .   .| [7:7] NP -> * Pronoun
|.   .   .   .   .   .   .   >   .   .| [7:7] Det -> * 'the'
|.   .   .   .   .   .   .   [---]   .| [7:8] Det -> 'the' *
|.   .   .   .   .   .   .   [--->   .| [7:8] NP -> Det * Nom
|.   .   .   .   .   .   .   .   >   .| [8:8] Nom -> * N
|.    .    .    .    .    .    .    .    >    .| [8:8] N   -> * 'restaurant'
|.    .    .    .    .    .    .    .    [----]| [8:9] N   -> 'restaurant' *
|.    .    .    .    .    .    .    .    [----]| [8:9] Nom -> N *
|.    .    .    .    .    .    .    [---------]| [7:9] NP -> Det Nom *
|.    .    .    .    .    .    [--------------]| [6:9] PP -> P NP *
|.    .    .    .    .    .    .    [--------->| [7:9] NP -> NP * PP
|.    .    .    .    .    .    .    .    .    >| [9:9] PP -> * P NP
|.    .    .    .    [------------------------]| [4:9] NP -> NP PP *
|.    [---------------------------------------]| [1:9] VP -> VP PP *
|.    .    [----------------------------------]| [2:9] NP -> NP PP *
|.    [---------------------------------------]| [1:9] VP -> V NP *
|.    .    [---------------------------------->| [2:9] S   -> NP * VP
|.    .    [---------------------------------->| [2:9] NP -> NP * PP
|.    .    .    .    .    .    .    .    .    >| [9:9] VP -> * VP PP
|.    .    .    .    .    .    .    .    .    >| [9:9] VP -> * V NP
|.    .    .    .    .    .    .    .    .    >| [9:9] VP -> * V S
|[============================================]| [0:9] S   -> NP VP *
|.    [--------------------------------------->| [1:9] VP -> VP * PP
|[============================================]| [0:9] S   -> NP VP *
|.    [--------------------------------------->| [1:9] VP -> VP * PP
|.    .    .    [-----------------------------]| [3:9] PP -> P NP *
|.    .    .    .    [------------------------>| [4:9] NP -> NP * PP
|.    .    [----------------------------------]| [2:9] NP -> NP PP *
|.    [---------------------------------------]| [1:9] VP -> VP PP *
(S
  (NP (Pronoun He))
  (VP
    (V eats)
    (NP
      (NP (NP (NNS pasta)) (PP (P with) (NP (Det a) (Nom (N fork)))))
      (PP (P in) (NP (Det the) (Nom (N restaurant)))))))
(S
  (NP (Pronoun He))
  (VP
    (V eats)
    (NP
      (NP (NNS pasta))
      (PP
        (P with)
        (NP
          (NP (Det a) (Nom (N fork)))
          (PP (P in) (NP (Det the) (Nom (N restaurant)))))))))
(S
  (NP (Pronoun He))
  (VP
    (VP
      (VP (V eats) (NP (NNS pasta)))
      (PP (P with) (NP (Det a) (Nom (N fork)))))
    (PP (P in) (NP (Det the) (Nom (N restaurant))))))
(S
  (NP (Pronoun He))
  (VP
    (VP
      (V eats)
      (NP (NP (NNS pasta)) (PP (P with) (NP (Det a) (Nom (N fork))))))
    (PP (P in) (NP (Det the) (Nom (N restaurant))))))
(S
  (NP (Pronoun He))
  (VP
    (VP (V eats) (NP (NNS pasta)))
    (PP
      (P with)
      (NP
        (NP (Det a) (Nom (N fork)))
        (PP (P in) (NP (Det the) (Nom (N restaurant))))))))
```

Q: Which of the parsers detects the ambiguity for S6 and S7?
A: Earley Chart Parser detects the ambiguity for S6 and S7. Earley Chart Parser is a dynamic programming. It will record all the parser results.
While Shift-reduce parser does not implement any backtracking, and it will only find at most one parse, even if more parses exist.

Question 4

Q4. Task 1

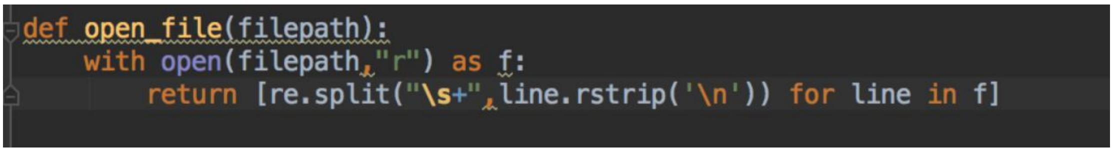
Answer:



```
word1   word2   GoldSimilarity  WordNetSimiliarity
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
wide    narrow  1.03    0.0
bad     awful   8.42    0.0
easy    difficult       0.58    0.0
bad     terrible        7.78    0.0
hard    simple  1.38    0.0
smart   dumb    0.55    0.25
insane  crazy   9.57    0.0
happy   mad     0.95    0.0
large   huge    9.47    0.0
hard    tough   8.05    1.0
new     fresh   6.83    1.0
sharp   dull    0.6     0.0
quick   rapid   9.7     0.125
dumb    foolish 6.67    0.0
```

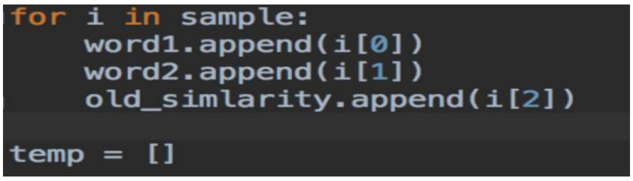
Process:

in task 1. we are going to use the re,os.path,wordnetand product package.



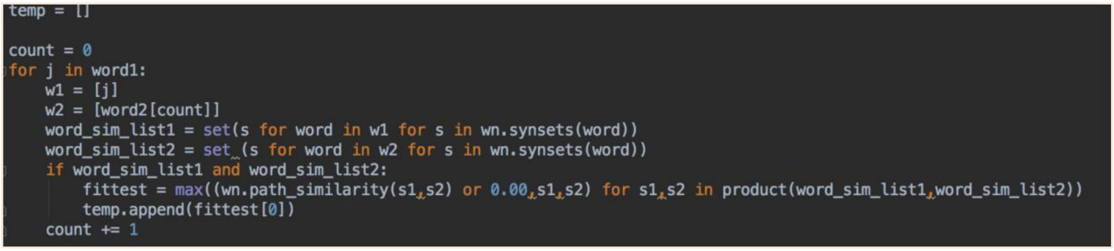
```
def open_file(filepath):
    with open(filepath,"r") as f:
        return [re.split("\s+",line.rstrip('\n')) for line in f]
```

First, we create an open_file method by using open function. After we open the file, we add the element to the array in order.



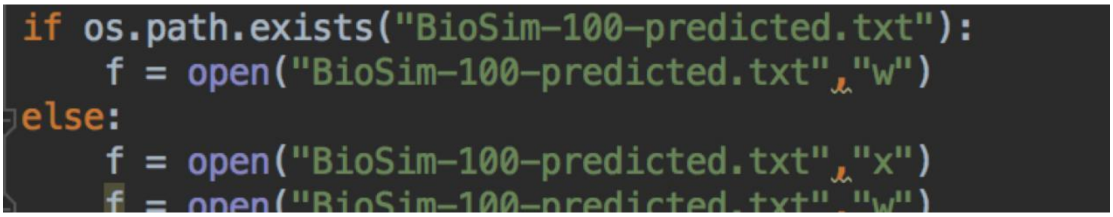
```
for i in sample:
    word1.append(i[0])
    word2.append(i[1])
    old_simlarity.append(i[2])

temp = []
```

The next step is going to compare the similarity of the two words.



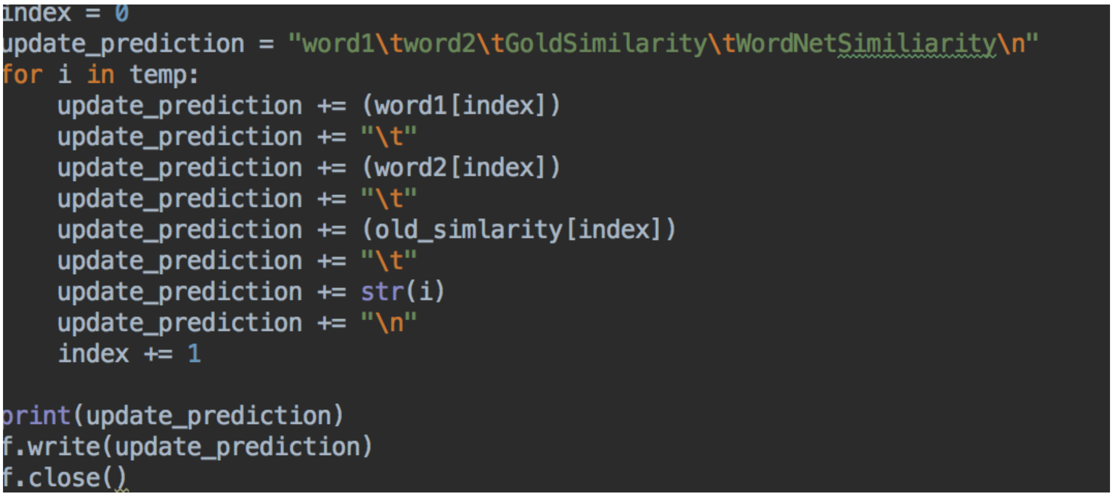
```
temp = []

count = 0
for j in word1:
    w1 = [j]
    w2 = [word2[count]]
    word_sim_list1 = set(s for word in w1 for s in wn.synsets(word))
    word_sim_list2 = set (s for word in w2 for s in wn.synsets(word))
    if word_sim_list1 and word_sim_list2:
        fittest = max((wn.path_similarity(s1,s2) or 0.00,s1,s2) for s1,s2 in product(word_sim_list1,word_sim_list2))
        temp.append(fittest[0])
    count += 1
```

We are going to use the Path_similarity function to compare the similarity of two words. We also use the set function to reduce the duplicate word. If there are no root of two words, we define the similarity between two words is 0.00.



```
if os.path.exists("BioSim-100-predicted.txt"):
    f = open("BioSim-100-predicted.txt","w")
else:
    f = open("BioSim-100-predicted.txt","x")
    f = open("BioSim-100-predicted.txt","w")
```

We create a file named “Bioxim-100-predicted.txt” to load the data we generate in the last function.

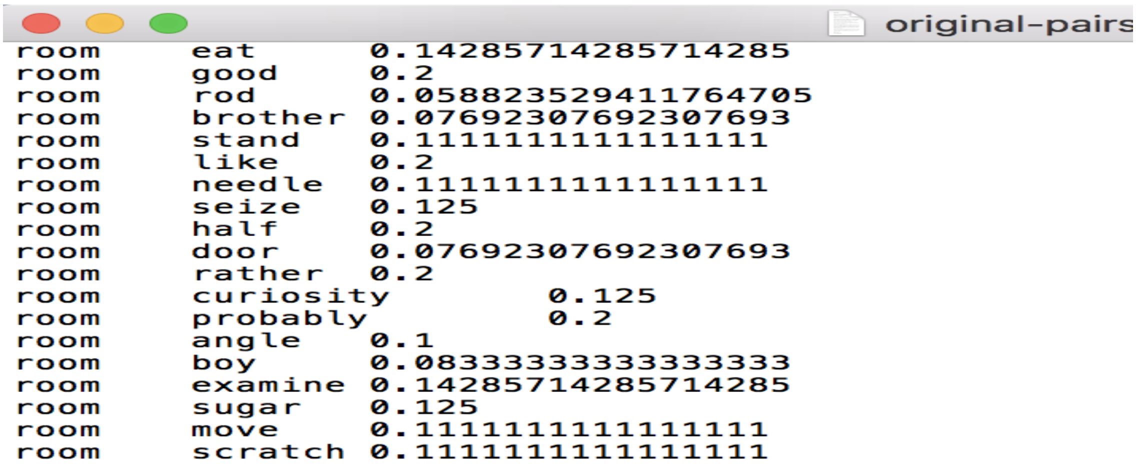


```
index = 0
update_prediction = "word1\tword2\tGoldSimilarity\tWordNetSimiliarity\n"
for i in temp:
    update_prediction += (word1[index])
    update_prediction += "\t"
    update_prediction += (word2[index])
    update_prediction += "\t"
    update_prediction += (old_simlarity[index])
    update_prediction += "\t"
    update_prediction += str(i)
    update_prediction += "\n"
    index += 1

print(update_prediction)
f.write(update_prediction)
f.close()
```

After create the file, we need to upload the data into this file. “word1” is the left-side word and “word2” is the right-side word. And “i” is the wordnet similarity. When we finish the string, we write to the file.

Q4.Task 2

Answer:

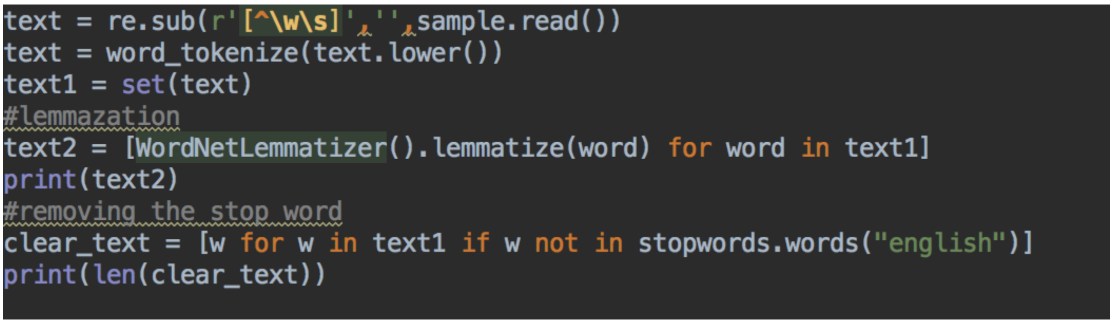


```
room    eat     0.14285714285714285
room    good    0.2
room    rod     0.058823529411764705
room    brother 0.07692307692307693
room    stand   0.1111111111111111
room    like    0.2
room    needle  0.1111111111111111
room    seize   0.125
room    half    0.2
room    door    0.07692307692307693
room    rather  0.2
room    curiosity       0.125
room    probably        0.2
room    angle   0.1
room    boy     0.08333333333333333
room    examine 0.14285714285714285
room    sugar   0.125
room    move    0.1111111111111111
room    scratch 0.1111111111111111
```

Process:

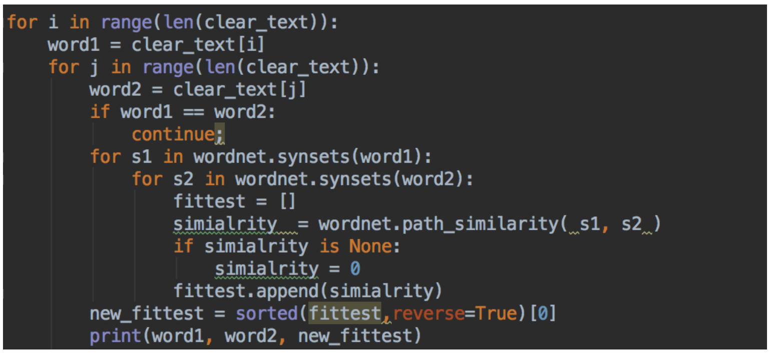
In task 2, we are going to use the word_tokenize, wordnet,stopwords and WordNetLemmatizer packages.

First of all, we need to clear the punctuation， remove the stopwords and remove the duplicated words.



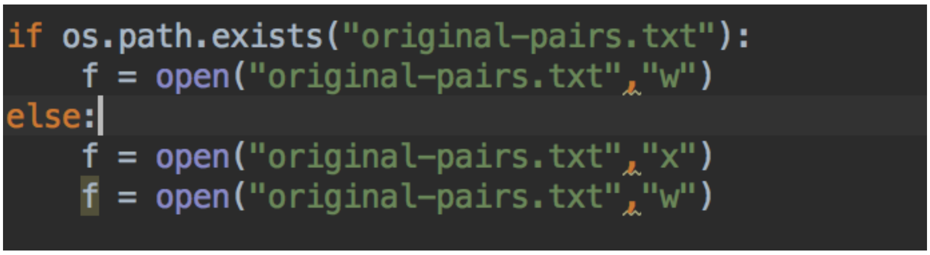
```
text = re.sub(r'[^\w\s]','',sample.read())
text = word_tokenize(text.lower())
text1 = set(text)
#lemmazation
text2 = [WordNetLemmatizer().lemmatize(word) for word in text1]
print(text2)
#removing the stop word
clear_text = [w for w in text1 if w not in stopwords.words("english")]
print(len(clear_text))
```

After we clean the data, we use two loops to compare the similarity between the words by using Path_similarity function.



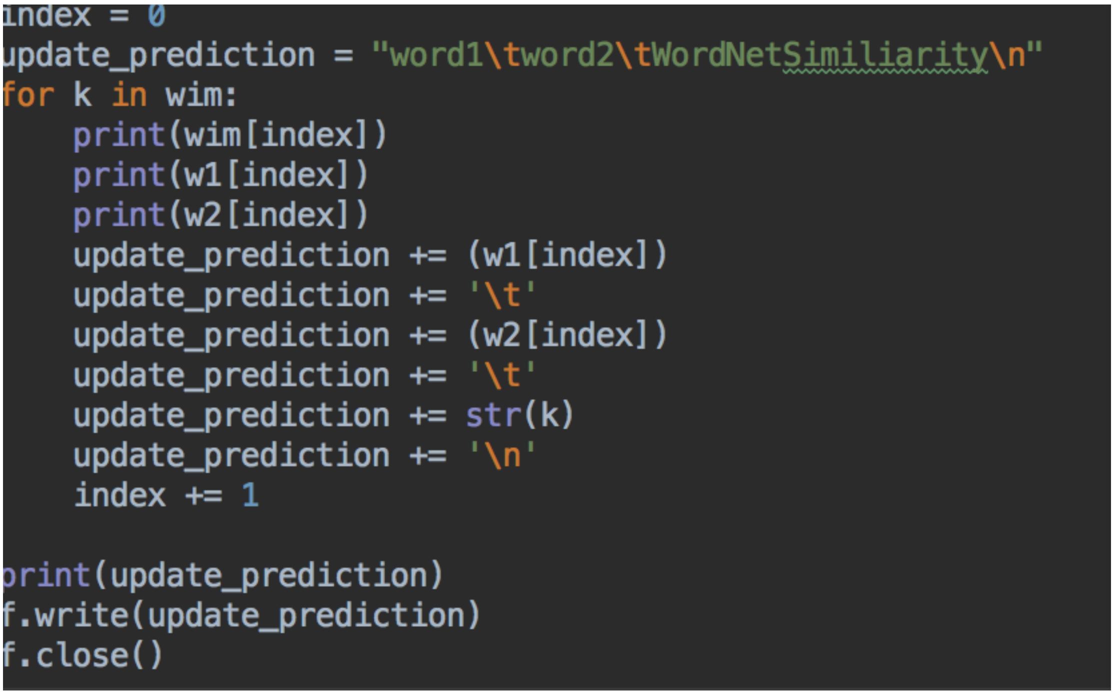
```
for i in range(len(clear_text)):
    word1 = clear_text[i]
    for j in range(len(clear_text)):
        word2 = clear_text[j]
        if word1 == word2:
            continue;
        for s1 in wordnet.synsets(word1):
            for s2 in wordnet.synsets(word2):
                fittest = []
                simialrity  = wordnet.path_similarity( s1, s2 )
                if simialrity is None:
                    simialrity = 0
                fittest.append(simialrity)
        new_fittest = sorted(fittest,reverse=True)[0]
        print(word1, word2, new_fittest)
```

We need to create a file named ‘original-pairs.txt ’ that load the data from the last function.



```
if os.path.exists("original-pairs.txt"):
    f = open("original-pairs.txt","w")
else:
    f = open("original-pairs.txt","x")
    f = open("original-pairs.txt","w")
```

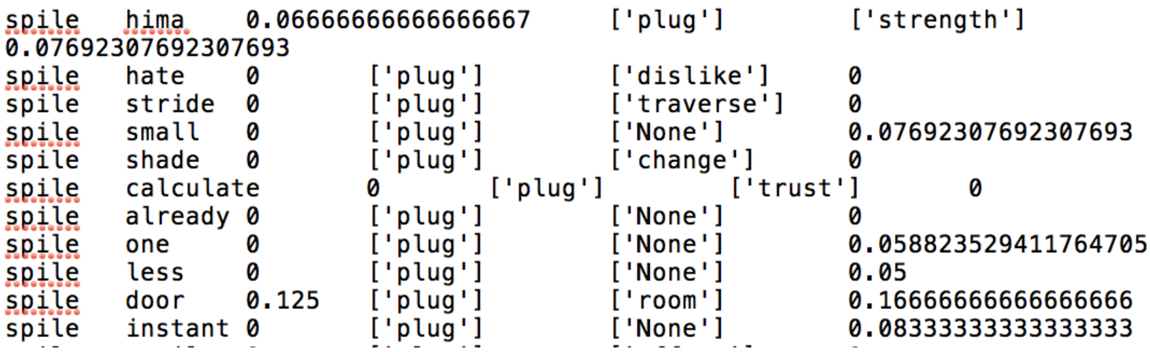
After we generate the file. We create the string and add to the document.



```
index = 0
update_prediction = "word1\tword2\tWordNetSimiliarity\n"
for k in wim:
    print(wim[index])
    print(w1[index])
    print(w2[index])
    update_prediction += (w1[index])
    update_prediction += '\t'
    update_prediction += (w2[index])
    update_prediction += '\t'
    update_prediction += str(k)
    update_prediction += '\n'
    index += 1

print(update_prediction)
f.write(update_prediction)
f.close()
```

Q4 task 3

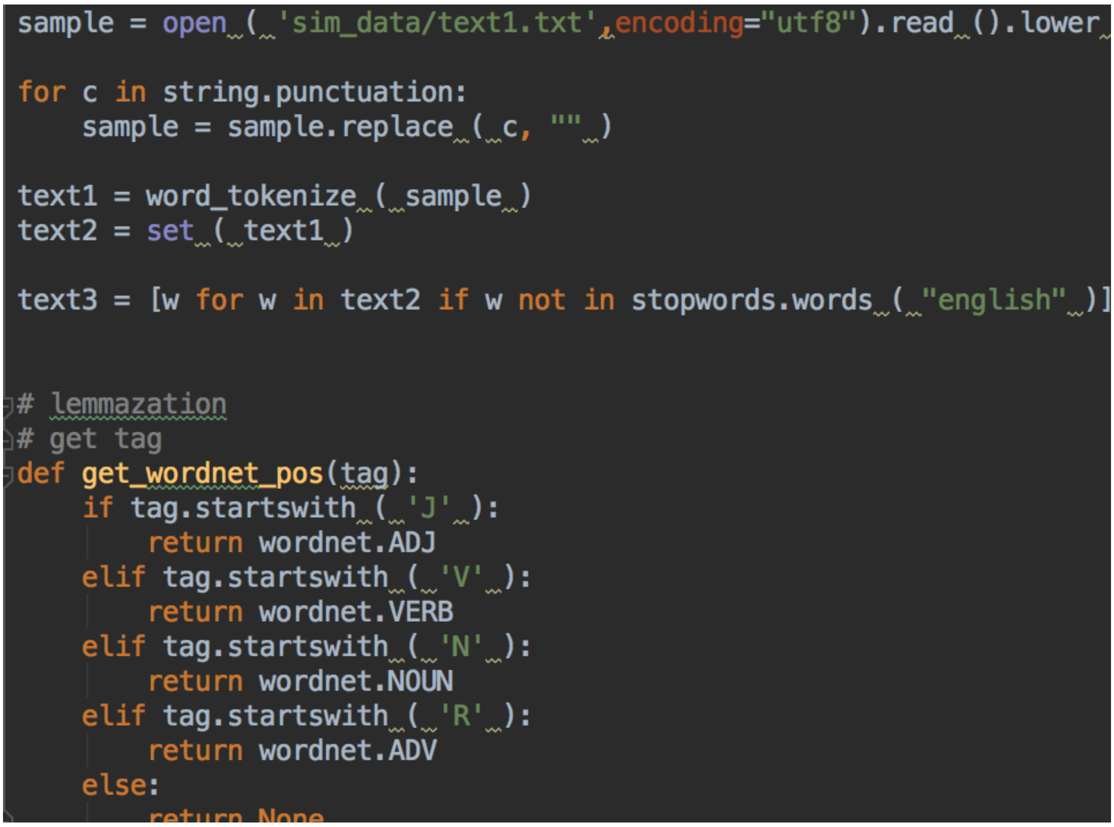
Answer:



```
spile   hima    0.06666666666666667     ['plug']        ['strength']
0.07692307692307693
spile   hate    0       ['plug']        ['dislike']     0
spile   stride  0       ['plug']        ['traverse']    0
spile   small   0       ['plug']        ['None']        0.07692307692307693
spile   shade   0       ['plug']        ['change']      0
spile   calculate       0       ['plug']        ['trust']       0
spile   already 0       ['plug']        ['None']        0
spile   one     0       ['plug']        ['None']        0.058823529411764705
spile   less    0       ['plug']        ['None']        0.05
spile   door    0.125   ['plug']        ['room']        0.16666666666666666
spile   instant 0       ['plug']        ['None']        0.08333333333333333
```

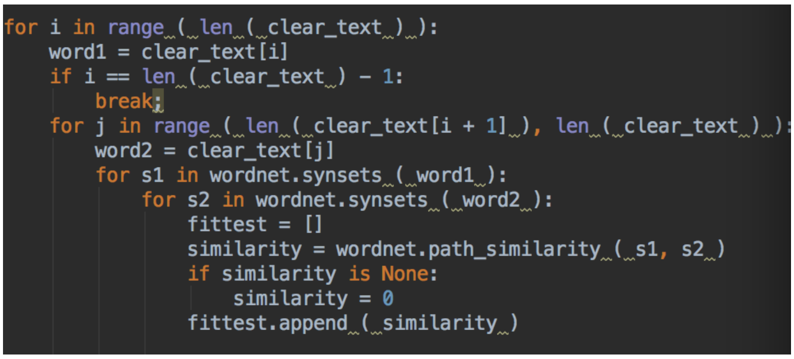
Process:

First of all, we need to get the lower letter, lemmazation, remove the punctuation and the stopwords**.**



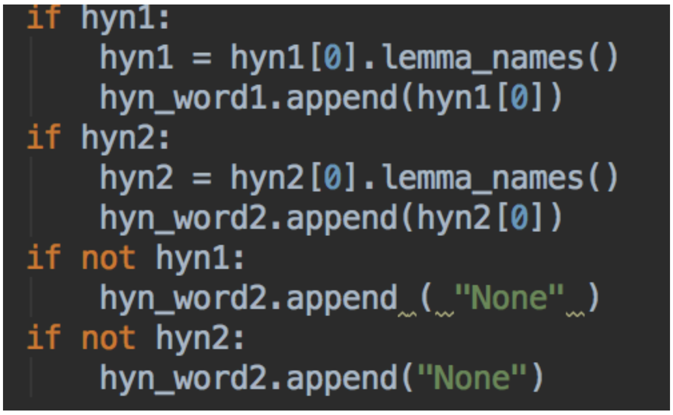
```
sample = open ( 'sim_data/text1.txt',encoding="utf8").read ().lower

for c in string.punctuation:
    sample = sample.replace ( c, "" )

text1 = word_tokenize ( sample )
text2 = set ( text1 )

text3 = [w for w in text2 if w not in stopwords.words ( "english" )]


# lemmazation
# get tag
def get_wordnet_pos(tag):
    if tag.startswith ( 'J' ):
        return wordnet.ADJ
    elif tag.startswith ( 'V' ):
        return wordnet.VERB
    elif tag.startswith ( 'N' ):
        return wordnet.NOUN
    elif tag.startswith ( 'R' ):
        return wordnet.ADV
    else:
        return None
```

Next, we use four loops to compare the similarity, we abstract the left word and right word in the first two loops. Then we get the hypwernyms of left and right word in the next two loops. Then we use the Path_similarity to compute the similarity



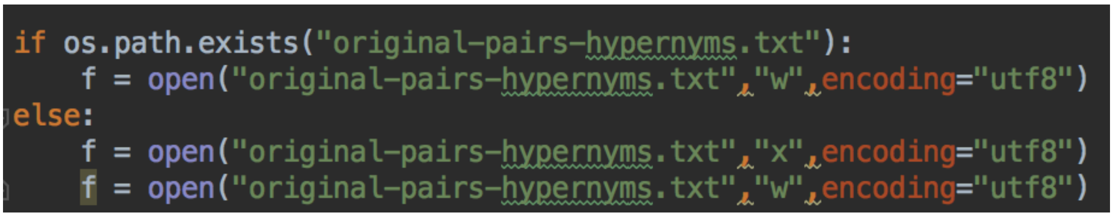
```
for i in range ( len ( clear_text ) ):
    word1 = clear_text[i]
    if i == len ( clear_text ) - 1:
        break;
    for j in range ( len ( clear_text[i + 1] ), len ( clear_text ) ):
        word2 = clear_text[j]
        for s1 in wordnet.synsets ( word1 ):
            for s2 in wordnet.synsets ( word2 ):
                fittest = []
                similarity = wordnet.path_similarity ( s1, s2 )
                if similarity is None:
                    similarity = 0
                fittest.append ( similarity )
```

If the hypernyms of left and right words is ‘None’, we need to exchange to the zero.



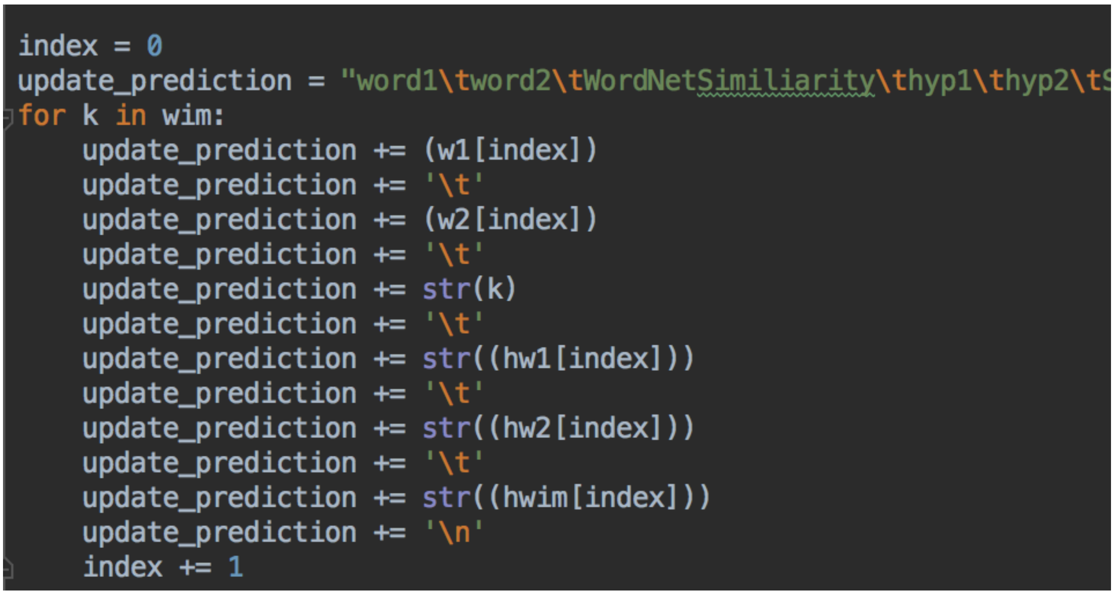
```
if hyn1:
    hyn1 = hyn1[0].lemma_names()
    hyn_word1.append(hyn1[0])
if hyn2:
    hyn2 = hyn2[0].lemma_names()
    hyn_word2.append(hyn2[0])
if not hyn1:
    hyn_word2.append ( "None" )
if not hyn2:
    hyn_word2.append("None")
```

After we generate the data, we need to create the document to load the data and the formatting is UTF-8.



```
if os.path.exists("original-pairs-hypernyms.txt"):
    f = open("original-pairs-hypernyms.txt","w",encoding="utf8")
else:
    f = open("original-pairs-hypernyms.txt","x",encoding="utf8")
    f = open("original-pairs-hypernyms.txt","w",encoding="utf8")
```

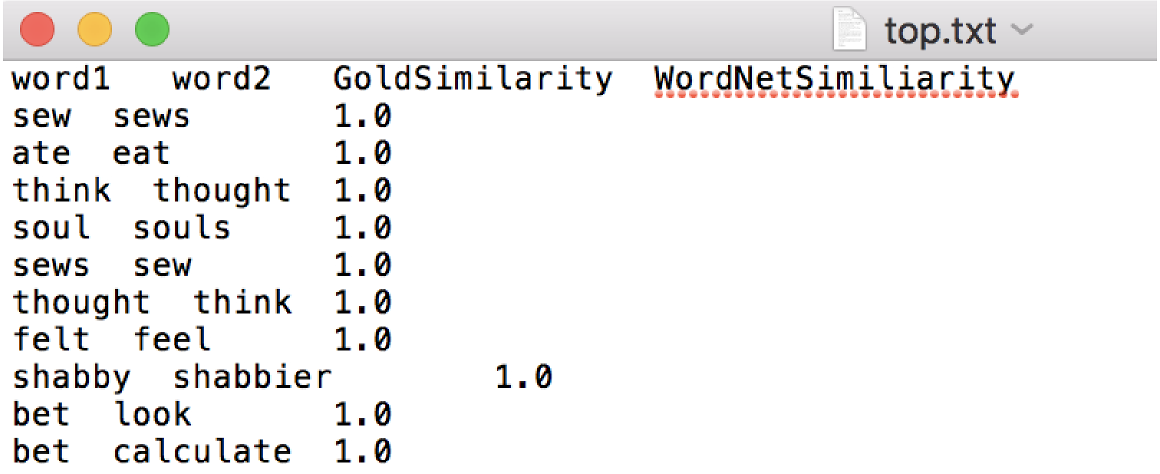
Then we create the string to meet the request and write into the file.



```
index = 0
update_prediction = "word1\tword2\tWordNetSimiliarity\thyp1\thyp2\tS
for k in wim:
    update_prediction += (w1[index])
    update_prediction += '\t'
    update_prediction += (w2[index])
    update_prediction += '\t'
    update_prediction += str(k)
    update_prediction += '\t'
    update_prediction += str((hw1[index]))
    update_prediction += '\t'
    update_prediction += str((hw2[index]))
    update_prediction += '\t'
    update_prediction += str((hwim[index]))
    update_prediction += '\n'
    index += 1
```

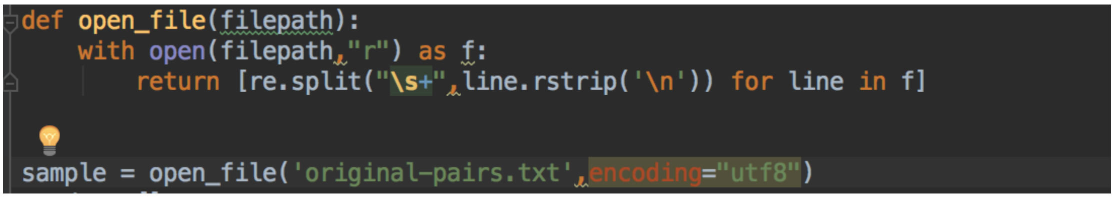
Q4 task 4

Answer:



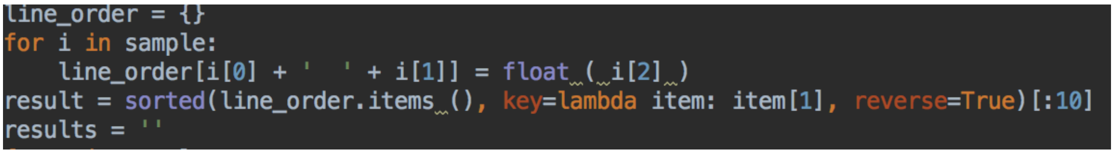
```
word1	word2	GoldSimilarity	WordNetSimiliarity
sew	sews		1.0
ate	eat		1.0
think	thought	1.0
soul	souls		1.0
sews	sew		1.0
thought	think	1.0
felt	feel		1.0
shabby	shabbier		1.0
bet	look		1.0
bet	calculate	1.0
```

Process:
In this task, we use the re and os.path packages. First of all, we open the file we generated in the task2 by using our own open_file function



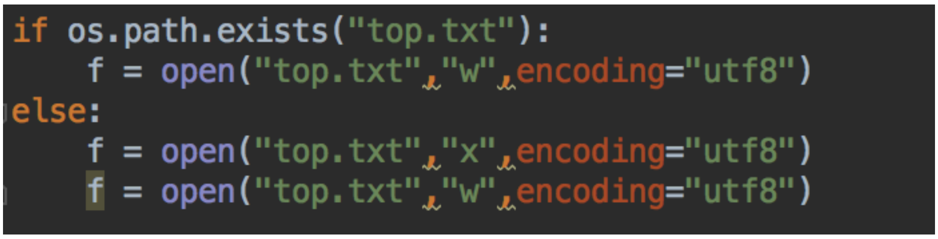
```
def open_file(filepath):
    with open(filepath,"r") as f:
        return [re.split("\s+",line.rstrip('\n')) for line in f]

sample = open_file('original-pairs.txt',encoding="utf8")
```

In this case we are going to use the dictionary function and set the similarity as the index of the dictionary. As the question asked us, we need to get the top 10 similarity



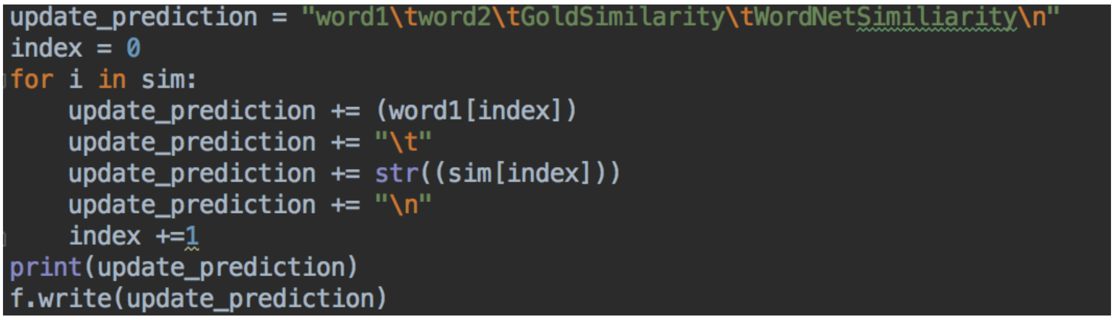
```
line_order = {}
for i in sample:
    line_order[i[0] + '  ' + i[1]] = float(i[2])
result = sorted(line_order.items(), key=lambda item: item[1], reverse=True)[:10]
results = ''
```

Then we create the file and upload the data we generate in the last function.



```
if os.path.exists("top.txt"):
    f = open("top.txt","w",encoding="utf8")
else:
    f = open("top.txt","x",encoding="utf8")
    f = open("top.txt","w",encoding="utf8")
```

After we finished the String, we add to the documents.



```
update_prediction = "word1\tword2\tGoldSimilarity\tWordNetSimiliarity\n"
index = 0
for i in sim:
    update_prediction += (word1[index])
    update_prediction += "\t"
    update_prediction += str((sim[index]))
    update_prediction += "\n"
    index +=1
print(update_prediction)
f.write(update_prediction)
```