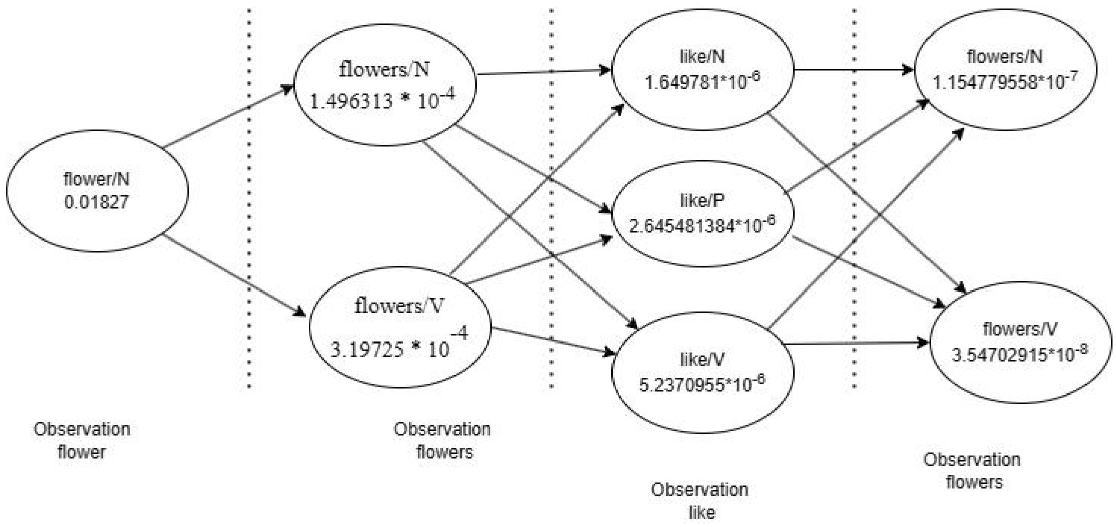
Họ và tên : ĐẶNG THANH HUY
MSSV:20530352
Chapter 6:
EX1:
Flower flowers like flowers.
L = V, N, ART, P
i = 1 to 4
SEQ(1,1) = PROB(flower/V)* PROB(V/O) = 0.05 * 0 = 0
SEQ(2,1) = PROB(flower/N)* PROB(N/O) = 0.063 * 0.29 = 0.01827
t = 2 to 4
i = 1 to 4
SEQ(1,2) = SEQ(2,1)* PROB(V/N)* PROB(flowers/V) = 0.01827 * 0.35 * 0.05 = $3.19725 * 10^{-4}$
SEQ(2,2) = SEQ(2,1)* PROB(N/N)* PROB(flowers/N) = 0.01827*0.13* 0.063 = $1.496313 * 10^{-4}$
t = 3 to 4
i = 1 to 4
SEQ(1,3) = max j=1,4 (SEQ(1,2)* PROB(V/V), SEQ(2,2)* PROB(V/N))* PROB(like/V) = max($3.19725 * 10^{-4}$ * 0, $1.496313 * 10^{-4}$ * 0.35) * 0.1 = $5.2370955 * 10^{-6}$
SEQ(2,3) = max j=1,4 (SEQ(1,2)* PROB(N/V), SEQ(2,2)* PROB(N/N)* PROB(like/N) = max($3.19725 * 10^{-4}$ * 0.43, $1.496313 * 10^{-4}$ * 0.13) * 0.012 = $1.649781 * 10^{-6}$
SEQ(4,3) = max j=1,4 (SEQ(1,2)* PROB(P/V),SEQ(2,2)*PROB(P/N))*PROB(like/P) = max($3.19725 * 10^{-4}$ * 0, $1.496313 * 10^{-4}$ * 0.26) * 0.068 = $2.645481384 * 10^{-6}$
t = 4
i = 1 to 4
SEQ(1,4) = SEQ(2,3)* PROB(V/N) * PROB(flower/V) = $1.649781 * 10^{-6}$ * 0.43 * 0.05 = $3.54702915 * 10^{-8}$
SEQ(2,4) = max j=1,4 (SEQ(1,3)* PROB(N/V), SEQ(2,3)* PROB(N/N), SEQ(4,3)*PROB(N/P)) * PROB(flower/N) = max($5.2370955 * 10^{-6}$ * 0.35, $1.649781 * 10^{-6}$ * 0.13, $2.645481384 * 10^{-6}$ * 0.26) * 0.063 = $1.154779558 * 10^{-7}$

Transition network:



EX2:

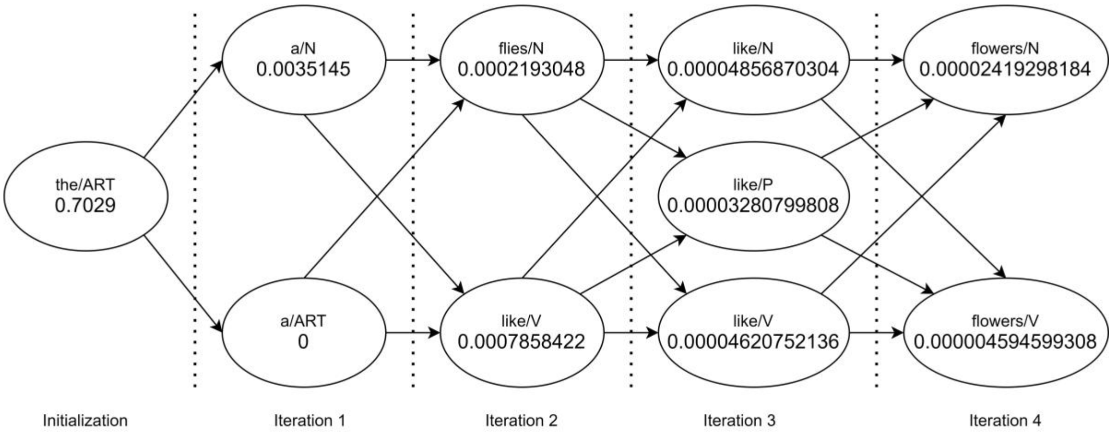
Initialization Step:
SEQSUM(1,1) = PROB(the|V) * PROB(V) = 0 * 0 = 0
SEQSUM(1,2) = PROB(the|N) * PROB(N) = 0.29 * 0 = 0
SEQSUM(1,3) = PROB(the|P) * PROB(P) = 0 * 0 = 0
SEQSUM(1,4) = PROB(the|ART) * PROB(ART) = 0.71 * 0.99 = 0.7029
Computing the Forward Probabilities:
t = 2 (a)
i = 1: SEQSUM(2,1) = (PROB(V|V) * SEQSUM(1,1) + PROB(V|N) * SEQSUM(1,2)
+
PROB(V|P) * SEQSUM(1,3) + PROB(V|ART) * SEQSUM(1,4)) * PROB(a|V)
= 0 * 0 + 0.43 * 0 + 0 * 0 + 0 * 0.7029 * 0
= 0
i = 2: SEQSUM(2,2) = (PROB(N|V) * SEQSUM(1,1) + PROB(N|N) * SEQSUM(1,2)
+
PROB(N|P) * SEQSUM(1,3) + PROB(N|ART) * SEQSUM(1,4)) * PROB(a|N)
= 0.35 * 0 + 0.13 * 0 + 0.26 * 0 + 1 * 0.7029 * 0.005
= 0.0035145
i = 3: SEQSUM(2,3) = (PROB(P|V) * SEQSUM(1,1) + PROB(P|N) * SEQSUM(1,2)
+
PROB(P|P) * SEQSUM(1,3) + PROB(P|ART) * SEQSUM(1,4)) * PROB(a|P)
= 0 * 0 + 0.44 * 0 + 0 * 0 + 0 * 0.7029 * 0
= 0
i = 4: SEQSUM(2,4) = (PROB(ART|V) * SEQSUM(1,1) + PROB(ART|N) *
SEQSUM(1,2)
+ PROB(ART|P) * SEQSUM(1,3) + PROB(ART|ART) * SEQSUM(1,4)) *
PROB(a|ART)
= 0.65 * 0 + 0 * 0 + 0.74 * 0 + 0 * 0.7029 * 0.995
= 0
t = 3 (flies)
i = 1: SEQSUM(3,1) = (PROB(V|V) * SEQSUM(2,1) + PROB(V|N) * SEQSUM(2,2)
+
PROB(V|P) * SEQSUM(2,3) + PROB(V|ART) * SEQSUM(2,4)) * PROB(flies|V)
= 0 * 0 + 0.43 * 0.0035145 + 0 * 0 + 0 * 0 * 0.52
= 0.0007858422

i = 2: SEQSUM(3,2) = (PROB(N|V) * SEQSUM(2,1) + PROB(N|N) * SEQSUM(2,2)
+
PROB(N|P) * SEQSUM(2,3) + PROB(N|ART) * SEQSUM(2,4)) * PROB(flies|N)
= 0.35 * 0 + 0.13 * 0.0035145 + 0.26 * 0 + 1 * 0 * 0.48
= 0.0002193048
i = 3: SEQSUM(3,3) = (PROB(P|V) * SEQSUM(2,1) + PROB(P|N) * SEQSUM(2,2)
+
PROB(P|P) * SEQSUM(2,3) + PROB(P|ART) * SEQSUM(2,4)) * PROB(flies|P)
= 0 * 0 + 0.44 * 0.0035145 + 0 * 0 + 0 * 0 * 0
= 0
i = 4: SEQSUM(3,4) = (PROB(ART|V) * SEQSUM(2,1) + PROB(ART|N) *
SEQSUM(2,2)
+ PROB(ART|P) * SEQSUM(2,3) + PROB(ART|ART) * SEQSUM(2,4)) *
PROB(flies|ART)
= 0.65 * 0 + 0 * 0.0035145 + 0.74 * 0 + 0 * 0 * 0
= 0
t = 4 (like)
i = 1: SEQSUM(4,1) = (PROB(V|V) * SEQSUM(3,1) + PROB(V|N) * SEQSUM(3,2)
+
PROB(V|P) * SEQSUM(3,3) + PROB(V|ART) * SEQSUM(3,4)) * PROB(like|V)
= 0 * 0.0007858422 + 0.43 * 0.0002193048 + 0 * 0 + 0 * 0 * 0.49
= 04620752136
i = 2: SEQSUM(4,2) = (PROB(N|V) * SEQSUM(3,1) + PROB(N|N) * SEQSUM(3,2)
+
PROB(N|P) * SEQSUM(3,3) + PROB(N|ART) * SEQSUM(3,4)) * PROB(like|N)
= 0.35 * 0.0007858422 + 0.13 * 0.0002193048 + 0.26 * 0 + 1 * 0 * 0.16
= 04856870304
i = 3: SEQSUM(4,3) = (PROB(P|V) * SEQSUM(3,1) + PROB(P|N) * SEQSUM(3,2)
+
PROB(P|P) * SEQSUM(3,3) + PROB(P|ART) * SEQSUM(3,4)) * PROB(like|P)
= 0 * 0.0007858422 + 0.44 * 0.0002193048 + 0 * 0 + 0 * 0 * 0.34
= 03280799808
i = 4: SEQSUM(4,4) = (PROB(ART|V) * SEQSUM(3,1) + PROB(ART|N) *
SEQSUM(3,2)
+ PROB(ART|P) * SEQSUM(3,3) + PROB(ART|ART) * SEQSUM(3,4)) *
PROB(like|ART)
= 0.65 * 0.0007858422 + 0 * 0.0002193048 + 0.74 * 0 + 0 * 0 * 0
= 0
t = 5 (flowers)
i = 1: SEQSUM(5,1) = (PROB(V|V) * SEQSUM(4,1) + PROB(V|N) * SEQSUM(4,2)
+
PROB(V|P) * SEQSUM(4,3) + PROB(V|ART) * SEQSUM(4,4)) * PROB(flowers|V)
= 0 * 04620752136 + 0.43 * 04856870304 + 0 * 03280799808 + 0 * 0 * 0.22
= 004594599308
i = 2: SEQSUM(5,2) = (PROB(N|V) * SEQSUM(4,1) + PROB(N|N) * SEQSUM(4,2)
+
PROB(N|P) * SEQSUM(4,3) + PROB(N|ART) * SEQSUM(4,4)) * PROB(flowers|N)
= 0.35 * 04620752136 + 0.13 * 04856870304 + 0.26 * 03280799808 + 1 * 0 * 0.78
= 02419298184

i = 3: SEQSUM(5,3) = (PROB(P|V) * SEQSUM(4,1) + PROB(P|N) * SEQSUM(4,2)
+
PROB(P|P) * SEQSUM(4,3) + PROB(P|ART) * SEQSUM(4,4)) * PROB(flowers|P)
= 0 * 04620752136 + 0.44 * 04856870304 + 0 * 03280799808 + 0 * 0 * 0
= 0
i = 4: SEQSUM(5,4) = (PROB(ART|V) * SEQSUM(4,1) + PROB(ART|N) *
SEQSUM(4,2)
+ PROB(ART|P) * SEQSUM(4,3) + PROB(ART|ART) * SEQSUM(4,4)) *
PROB(flowers|ART)
= 0.65 * 04620752136 + 0 * 04856870304 + 0.74 * 03280799808 + 0 * 0 * 0
= 0
Computing the Lexical Probabilities:
Normalization at t = 1:
PROB(the|V) = SEQSUM(1,1) / SEQSUM(1,1) + SEQSUM(1,2) + SEQSUM(1,3) +
SEQSUM(1,4)
= 0 / 0.7029
= 0
PROB(the|N) = SEQSUM(1,2) / SEQSUM(1,1) + SEQSUM(1,2) + SEQSUM(1,3) +
SEQSUM(1,4)
= 0 / 0.7029
= 0
PROB(the|P) = SEQSUM(1,3) / SEQSUM(1,1) + SEQSUM(1,2) + SEQSUM(1,3) +
SEQSUM(1,4)
= 0 / 0.7029
= 0
PROB(the|ART) = SEQSUM(1,4) / SEQSUM(1,1) + SEQSUM(1,2) + SEQSUM(1,3)
+
SEQSUM(1,4)
= 0.7029 / 0.7029
= 1
Normalization at t = 2:
PROB(a|V) = SEQSUM(2,1) / SEQSUM(2,1) + SEQSUM(2,2) + SEQSUM(2,3) +
SEQSUM(2,4)
= 0 / 0.0035145
= 0
PROB(a|N) = SEQSUM(2,2) / SEQSUM(2,1) + SEQSUM(2,2) + SEQSUM(2,3) +
SEQSUM(2,4)
= 0.0035145 / 0.0035145
= 1
PROB(a|P) = SEQSUM(2,3) / SEQSUM(2,1) + SEQSUM(2,2) + SEQSUM(2,3) +
SEQSUM(2,4)
= 0 / 0.0035145
= 0
PROB(a|ART) = SEQSUM(2,4) / SEQSUM(2,1) + SEQSUM(2,2) + SEQSUM(2,3) +
SEQSUM(2,4)
= 0 / 0.0035145
= 0
Normalization at t = 3:
PROB(flies|V) = SEQSUM(3,1) / SEQSUM(3,1) + SEQSUM(3,2) + SEQSUM(3,3) +
SEQSUM(3,4)

= 0.0007858422 / 0.001005147
= 0.7818181818
PROB(flies|N) = SEQSUM(3,2) / SEQSUM(3,1) + SEQSUM(3,2) + SEQSUM(3,3) + SEQSUM(3,4)
= 0.0002193048 / 0.001005147
= 0.2181818182
PROB(flies|P) = SEQSUM(3,3) / SEQSUM(3,1) + SEQSUM(3,2) + SEQSUM(3,3) + SEQSUM(3,4)
= 0 / 0.001005147
= 0
PROB(flies|ART) = SEQSUM(3,4) / SEQSUM(3,1) + SEQSUM(3,2) +
SEQSUM(3,3) +
SEQSUM(3,4)
= 0 / 0.001005147
= 0
Normalization at t = 4:
PROB(like|V) = SEQSUM(4,1) / SEQSUM(4,1) + SEQSUM(4,2) + SEQSUM(4,3) + SEQSUM(4,4)
= 04620752136 / 0.0001275842225
= 0.3621726923
PROB(like|N) = SEQSUM(4,2) / SEQSUM(4,1) + SEQSUM(4,2) + SEQSUM(4,3) + SEQSUM(4,4)
= 04856870304 / 0.0001275842225
= 0.3806795393
PROB(like|P) = SEQSUM(4,3) / SEQSUM(4,1) + SEQSUM(4,2) + SEQSUM(4,3) + SEQSUM(4,4)
= 03280799808 / 0.0001275842225
= 0.2571477683
PROB(like|ART) = SEQSUM(4,4) / SEQSUM(4,1) + SEQSUM(4,2) + SEQSUM(4,3) +
SEQSUM(4,4)
= 0 / 0.0001275842225
= 0
Normalization at t = 5:
PROB(flowers|V) = SEQSUM(5,1) / SEQSUM(5,1) + SEQSUM(5,2) +
SEQSUM(5,3) +
SEQSUM(5,4)
= 004594599308 / 02878758115
= 0.1596035208
PROB(flowers|N) = SEQSUM(5,2) / SEQSUM(5,1) + SEQSUM(5,2) +
SEQSUM(5,3) +
SEQSUM(5,4)
= 02419298184 / 02878758115
= 0.8403964791
PROB(flowers|P) = SEQSUM(5,3) / SEQSUM(5,1) + SEQSUM(5,2) + SEQSUM(5,3) +
SEQSUM(5,4)
= 0 / 02878758115
= 0

PROB(flowers|ART) = SEQSUM(5,4) / SEQSUM(5,1) + SEQSUM(5,2) + SEQSUM(5,3)
+ SEQSUM(5,4)
= 0 / 02878758115
= 0



Chapter 7:
Ex1:
a) "A man stopped at every truck stop."
This sentence is ambiguous due to its possible syntactic structures. It can be interpreted in two main ways:
1. The man stopped at every truck stop he encountered.
Paraphrase: The man halted at each truck stop along his journey.
2. There was a man at each truck stop who stopped.
Paraphrase: At each truck stop, there was a man who came to a halt.

b) "Several people ate the pizza."
This sentence is ambiguous due to its possible word senses. It can be interpreted in two main ways:
1.Different groups of people consumed the pizza separately.
Paraphrase: Multiple groups of individuals had pizza.
2.A group of people together consumed the pizza.
Paraphrase: A number of people shared in eating the pizza.
c) "We saw her duck."
This sentence is ambiguous due to a combination of syntactic structures and word senses. It can be interpreted in two main ways:
1.We observed the duck belonging to her.
Paraphrase: We witnessed the duck that belongs to her.
2.We observed her perform the action of ducking.
Paraphrase: We witnessed her ducking (bending down quickly).
EX2:
1) George ate a pizza at every road stop.
Ambiguity: The sentence could mean either that George ate one distinct pizza at each road stop, or he ate portions of the same pizza across multiple road stops.
Quasi-Logical Forms:
Distinct Pizzas Interpretation:
$\forall$s(s in RoadStops$\rightarrow\exists$p(p in Pizzas$\land$EAT(George,p,s)))

This interpretation suggests that for each road stop s, there is a different pizza p that George ate.
Same Pizza, Multiple Stops Interpretation:
$\exists p(p \text{ in Pizzas} \wedge \forall s(s \text{ in RoadStops} \rightarrow EAT(George,p,s)))$

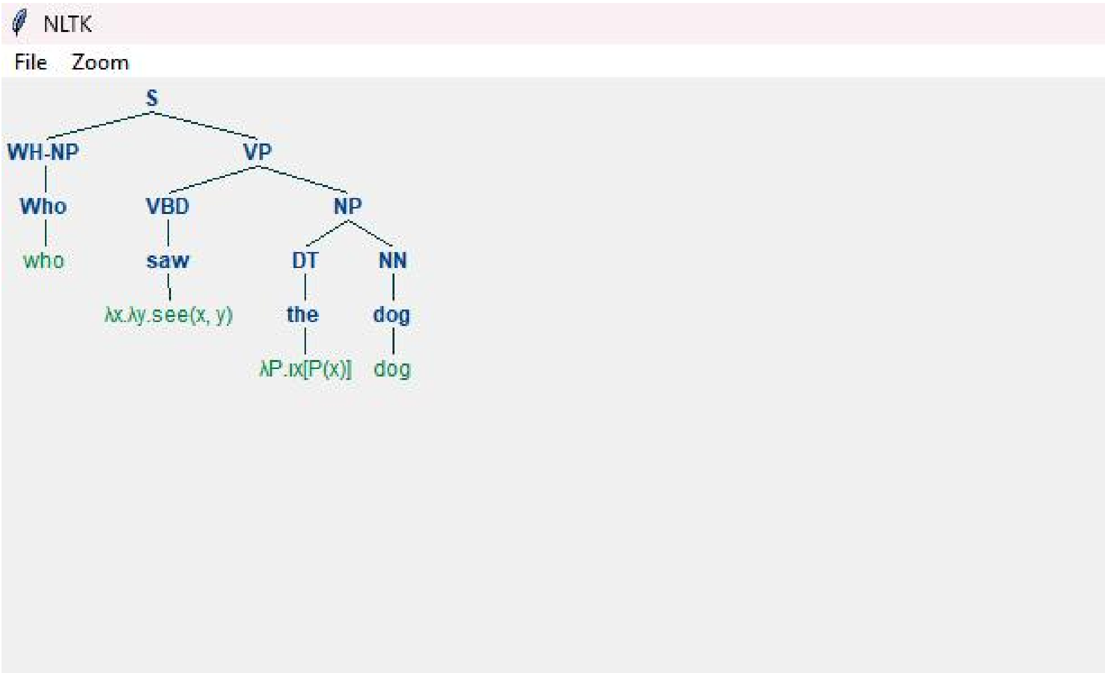
This form suggests that there is one pizza p that George ate across all road stops.
2) Several employees from every company bought a pizza.
Ambiguity: This sentence could mean either that a group of employees from each company collectively purchased one pizza, or each employee from every company purchased a separate pizza.

Quasi-Logical Forms:
Collective Purchase Interpretation:
$\forall c(c \text{ in Companies} \rightarrow \exists e(e \text{ in Employees of } c \wedge \exists p(p \text{ in Pizzas} \wedge BUY(e,p))))$
This interpretation means that for each company c, there is a group of employees e who collectively buy at least one pizza p.
Individual Purchase Interpretation:
$\forall c(c \text{ in Companies} \rightarrow \forall e(e \text{ in Employees of } c \rightarrow \exists p(p \text{ in Pizzas} \wedge BUY(e,p))))$
This form implies that every employee e from each company c buys their own individual pizza p.
3) We saw John in the park by the beach.
Ambiguity: The sentence might mean that John was seen in a park located next to the beach, or it might mean that John was seen at both the park and also independently by the beach.
Quasi-Logical Forms:
Park Adjacent to the Beach Interpretation:
SEE(We,John,IN(Park adjacent to Beach))
This implies that the park where John was seen is next to the beach.
Separate Locations Interpretation:
$SEE(We,John,IN(Park) \wedge IN(Beach))$
This interpretation suggests that John was seen both in the park and, separately, at the beach.
Chapter 8:
EX1:
$(\lambda x(Px)A)$simplifies to PA
$((\lambda x(xA))(\lambda y(Qy)))$ simplifies to QA
$((\lambda x((\lambda y(Py))x))A)$ simplifies to PA
EX2:
Syntactic Analysis:
NP1 (The man)
VP (gave the apple to Bill)
    V (gave)
    NP2 (the apple)
    PP (to Bill)
        P (to)
        NP3 (Bill)
Lexical Entries:
man: λm (MAN m1)
apple: λa (APPLE a1)

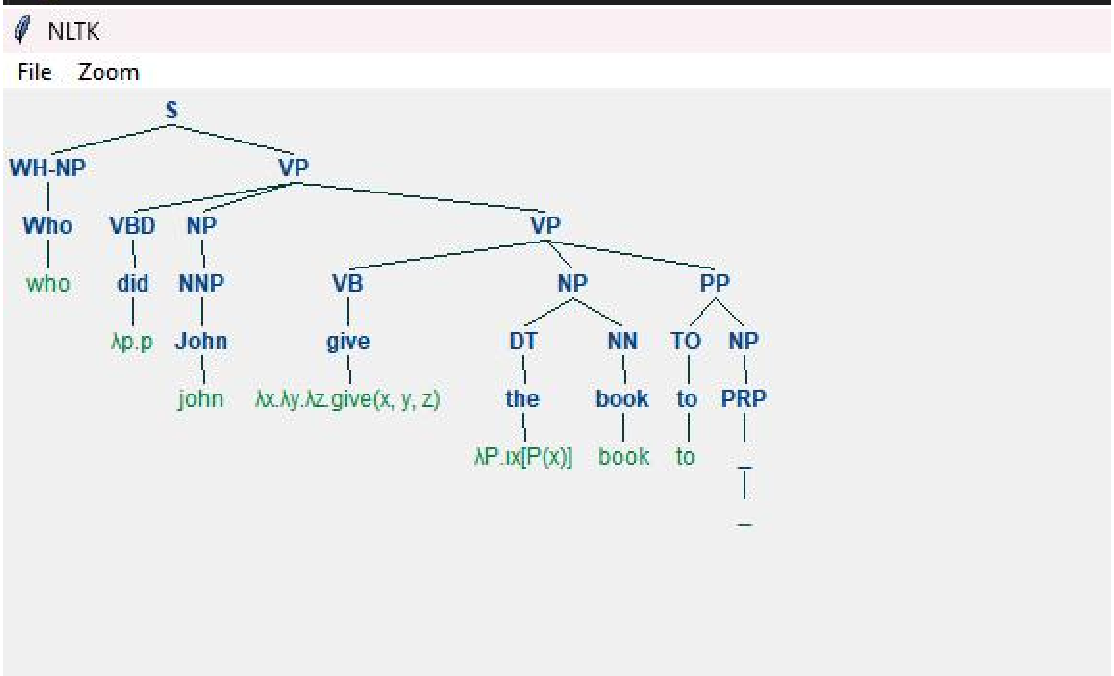
Bill: λb (NAME b1 "Bill")
gave: λx λy λz (PAST (GIVE x y z)) - where x is the giver, y is the thing given, and z is
the receiver.
to: λt (TO t1)
Semantic Interpretation:
NP1:
the: λP λx (THE x (P x))
man: λm1 (MAN m1)
SEM(NP1) = (λP λx (THE x (P)x))) (λm1 (MAN m1)) = λm1 (THE m1 (MAN m1))
NP2:
the: λP λx (THE x (P x))
apple: λa1 (APPLE a1)
SEM(NP2) = (λP λx (THE x (Px))) (λa1 (APPLE a1)) = λa1 (THE a1 (APPLE a1))
PP:
to: λt (TO t1)
Bill: λb1 (NAME b1 "Bill")
SEM(PP) = λt (TO t1) (λb1 (NAME b1 "Bill")) = TO (NAME "Bill")
VP:
gave: λx λy λz (PAST (GIVE x y z))
NP2: λa1 (THE a1 (APPLE a1))
PP: TO (NAME "Bill")
SEM(VP) = λx λy λz ((TO (NAME "Bill")) PAST (GIVE x y z)) (λa1 (THE a1 (APPLE a1))) (NAME "Bill") = λx ((TO (NAME "Bill")) PAST (GIVE x (λa1 (THE a1 (APPLE a1))) (NAME "Bill")))
S:
NP1: λm1 (THE m1 (MAN m1))
VP: λx (PAST (GIVE x (λa1 (THE a1 (APPLE a1))))TO (NAME "Bill"))
SEM(S) = λx ((TO (NAME "Bill")) PAST (GIVE x (λa1 (THE a1 (APPLE a1)))) (NAME "Bill") (λm1 (THE m1 (MAN m1))) = (TO (NAME "Bill")) PAST (GIVE λm1 (THE m1 (MAN m1)) (λa1 (THE a1 (APPLE a1))) (NAME "Bill"))
EX3:
I use python to draw .

BTCN.py > create_trees

```python
import nltk
from nltk import Tree

# Define the function to display a parse tree
def draw_parse_tree(tree):
    tree.draw()

# Define the function to create parse trees with semantic annotations
def create_trees():
    # Tree for "Who saw the dog?"
    t1 = Tree('S', [
        Tree('WH-NP', [Tree('Who', ['who'])]),
        Tree('VP', [
            Tree('VBD', [Tree('saw', ['λx.λy.see(x, y)'])]),
            Tree('NP', [
                Tree('DT', [Tree('the', ['λP.ιx[P(x)]'])]),
                Tree('NN', [Tree('dog', ['dog'])])
            ])
        ]),
    ])

```



```python
# Tree for "Who did John give the book to?"
t2 = Tree('S', [
    Tree('WH-NP', [Tree('Who', ['who'])]),
    Tree('VP', [
        Tree('VBD', [Tree('did', ['λp.p'])]),
        Tree('NP', [Tree('NNP', [Tree('John', ['john'])])]),
        Tree('VP', [
            Tree('VB', [Tree('give', ['λx.λy.λz.give(x, y, z)'])]),
            Tree('NP', [
                Tree('DT', [Tree('the', ['λP.ιx[P(x)]'])]),
                Tree('NN', [Tree('book', ['book'])])
            ]),
            Tree('PP', [
                Tree('TO', [Tree('to', ['to'])]),
                Tree('NP', [Tree('PRP', [Tree('_', ['_'])])])
            ])
        ])
    ])
])
```



Chapter 9 :

EX1:

- For "bridge/STRUCTURE1":
  - Count (the/bridge/STRUCTURE1) = 5500
  - Count (suspension/bridge/STRUCTURE1) = 200
  - Count (bridge/STRUCTURE1) = 5651
  - Count (construction/bridge/ STRUCTURE1) = 1
  - Count (the) in the corpus = 500,000
  - Count (suspension) in the corpus = 2000
  - Count (construction) in the corpus = 1000
- For "bridge/DENTAL-DEV37":
  - Count (the/bridge/DENTAL-DEV37) = 180
  - Count (suspension/bridge/DENTAL-DEV37) = 1

- Count (bridge/DENTAL-DEV37) = 194
- Count (construction/bridge/DENTAL-DEV37) = presumed low 0.1 due to it being rare

- STRUCTURE1:

Cn(the/bridge/STRUCTURE1)=10^7*5500/5651*500000≈1.94

Cn(suspension/bridge/STRUCTURE1)10^7× 200/5651×2000≈17.7

Cn(construction/bridge/STRUCTURE1)=10^7× 1/5651×1000≈1.8

- DENTAL-DEV37:

Cn(the/bridge/DENTAL-DEV37)=10^7× 180/194×500000≈1.84

Cn(suspension/bridge/DENTAL-DEV37)=10^7× 1/194×2000≈2.6

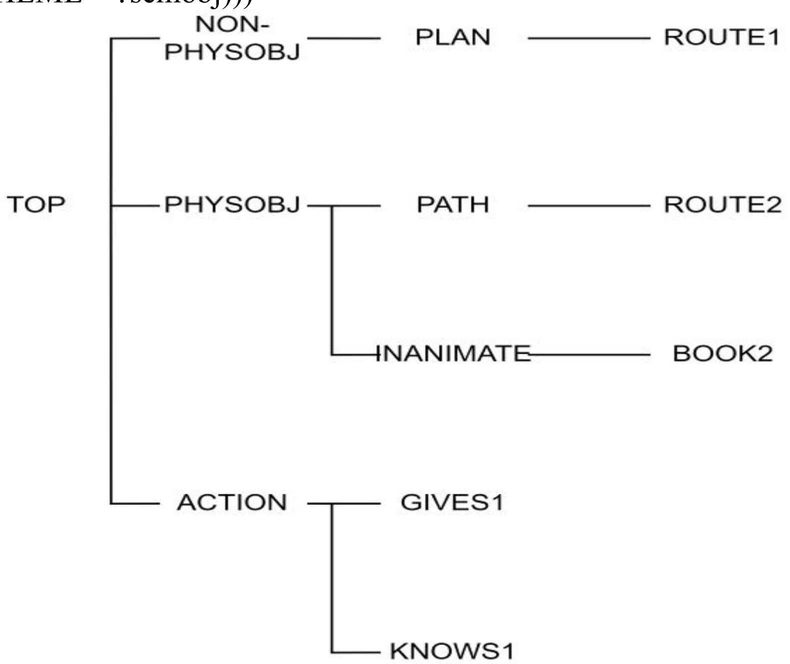
Cn(construction/bridge/DENTAL-DEV37)=10^7× 0.1/194×1000≈0.005

EX2:

The given grammar is enough to appropriately interpret these sentences.

Extend the lexicon and sense hierarchy:

route (N AGR 3s SEM ROUTE1)
route (N AGR 3s SEM ROUTE2)
book (N AGR 3s SEM BOOK2)
gave (V SUBCAT _np VFORM past SUBJ ?subj OBJ? SEM (& (GIVES1 *)
(AGENT * ? subj)
(THEME * ?semobj)))
knows (V SUBCAT _np VFORM pres SUBJ ? subj OBJ? SEM (& (KNOWS1 *)
(AGENT * ?subj)
(THEME * ?semobj)))



Extend the selectional restrictions:

(AGENT GIVES1 PERSON)
(THEME GIVES1 PHYSOBJ)
(AGENT KNOWS1 PERSON)
(THEME KNOWS1 NON-PHYSOBJ || PHYSOBJ)