Anas Salamah

NLP Assignment 5
10/9/14

***Please see last three pages for code and math work.**

1 Tagging and Tag Sets (10 points)

1.1 When taggers go bad (5 points)

British Left Waffles on Falkland Islands

N V N PREP N N

N N N PREP N N

1.2 Exploring the tag set (5 points)

1.

Number of Tags	Number of Distinct Words
1	47328
2	7186
3	1146
4	265
5	87
6	27
7	12
8	1
9	1
10	2

2.

"that" as a CS-NC: When I have instructions to leave is equivalent in meaning to I have instructions that I am to leave this place, dominant stress is ordinarily on leave.

"that" as a WPS-NC: But when to represents to consciousness in that was the moment that I came to, and similarly in that was the moment I came to, there is much stronger stress on to.

"that" as a DT-NC: Thus to has light stress both in that was the conclusion that I came to and in that was the conclusion I came to.

"that" as a CS-HL: According to the official interpretation of the Charter, a member cannot be penalized by not having the right to vote in the General Assembly for nonpayment of financial obligations to the `` special " United Nations" budgets, and of course cannot be expelled from the Organization which yo suggested in your editorial , due to the fact that there is no provision in the Charter for expulsion.

"that" as a WPO-NC: Thus to has light stress both in that was the conclusion that I came to and in that was the conclusion I came to.

"that" as a NIL: Thus, as a development program is being launched, commitments and obligations must be entered into in a given year which may exceed by twofold or threefold the expenditures to be made in that year.

"that" as a QL: Then, she was back on her feet, winking and smiling that enormous smile she had lots of wonderful big teeth that yo never would have suspected she had when she was not smiling.

"that" as a CS: She was a living doll and no mistake -- the blue-black bang, the wide cheekbones, olive-flushed, that betrayed the Cherokee strain in her Midwestern lineage, and the mouth whose only fault, in the "novelists" carping phrase, was that the lower lip was a trifle too voluptuous.

"that" as a WPS: She was a living doll and no mistake -- the blue-black bang, the wide cheekbones, olive-flushed, that betrayed the Cherokee strain in her Midwestern lineage, and the mouth whose only fault, in the "novelists" carping phrase, was that the lower lip was a trifle too voluptuous.

"that" as a WPS-HL: Withholding of funds to schools that deny children on account of race.

"that" as a DT: " See that guy " ? ?"that" as a WPO: It was nothing that he said or did, but it seemed so natural to her that she should be working for him, looking forward to his eventual proposal.

2 Viterbi Algorithm (30 Points)

2.1 Emission Probability (10 points)

Frequency Table:

	NOUN	VERB	CONJ	PRO
’e	0.1	0.1	0.1	1.1
’eg	0.1	0.1	1.1	0.1

ghaH	0.1	0.1	0.1	1.1
ja’chuqmeH	0.1	1.1	0.1	0.1
Legh	0.1	0.1	0.1	0.1
neH	0.1	1.1	0.1	0.1
pa’Daq	1.1	0.1	0.1	0.1
puq	2.1	0.1	0.1	0.1
qIp	0.1	2.1	0.1	0.1
rajHom	1.1	0.1	0.1	0.1
taH	0.1	1.1	0.1	0.1
tara’ngan	4.1	0.1	0.1	0.1
yaS	0.1	0.1	0.1	0.1

Emission Probability:

	NOUN	VERB	CONJ	PRO
’e	0.0108	0.0159	0.0435	0.3333
’eg	0.0108	0.0159	0.4783	0.0303
ghaH	0.0108	0.0159	0.0435	0.3333
ja’chuqmeH	0.0108	0.1746	0.0435	0.0303
Legh	0.0108	0.0159	0.0435	0.0303
neH	0.0108	0.1746	0.0435	0.0303
pa’Daq	0.1183	0.0159	0.0435	0.0303
puq	0.2258	0.0159	0.0435	0.0303
qIp	0.0108	0.3333	0.0435	0.0303

rajHom	0.1183	0.0159	0.0435	0.0303
taH	0.0108	0.1746	0.0435	0.0303
tara’ngan	0.4409	0.0159	0.0435	0.0303
yaS	0.0108	0.0159	0.0435	0.0303

2.2 Start and Transition Probability (5 points)

Frequency Table:

	NOUN	VERB	CONJ	PRO
START	2.1	1.1	0.1	0.1
N	0.1	3.1	1.1	2.1
V	5.1	0.1	0.1	0.1
CONJ	1.1	0.1	0.1	0.1
PRO	0.1	1.1	0.1	0.1

Initial and transition probabilities:

	NOUN	VERB	CONJ	PRO
START	.618	0.324	0.029	0.029
N	0.016	0.484	0.172	0.328
V	0.944	0.019	0.019	0.019
CONJ	.786	0.071	0.071	0.071
PRO	0.071	.786	0.071	0.071

2.3 Viterbi Decoding (15 points)

1.

= (πN * βN,”tara’ngan”)* (σN,V * βV,”legh”) * (σV,N * βN,”yaS”)

= (0.618 * 0.4409) * (0.484 * 0.0159) * (0.944 * 0.0108)

=2.1e-5

2.

POS	n = 1	n = 2	n = 3
z = N	-1.9	-14.2	-15.5
z = V	-7.6	-8.9	-14.9
z = CONJ	-9.6	-9.0	-16.9
z = PRO	-10.2	-8.6	-17.5

3.

NOUN, PRO, VERB

4.

(a) If we log the probability of sequence NOUN, VERB, NOUN, we would get, log(3.2e-5) = -15.5. The log probability for NOUN, VERB, NOUN is a little bit lower than the sequence NOUN, PRO, VERB which is -14.9

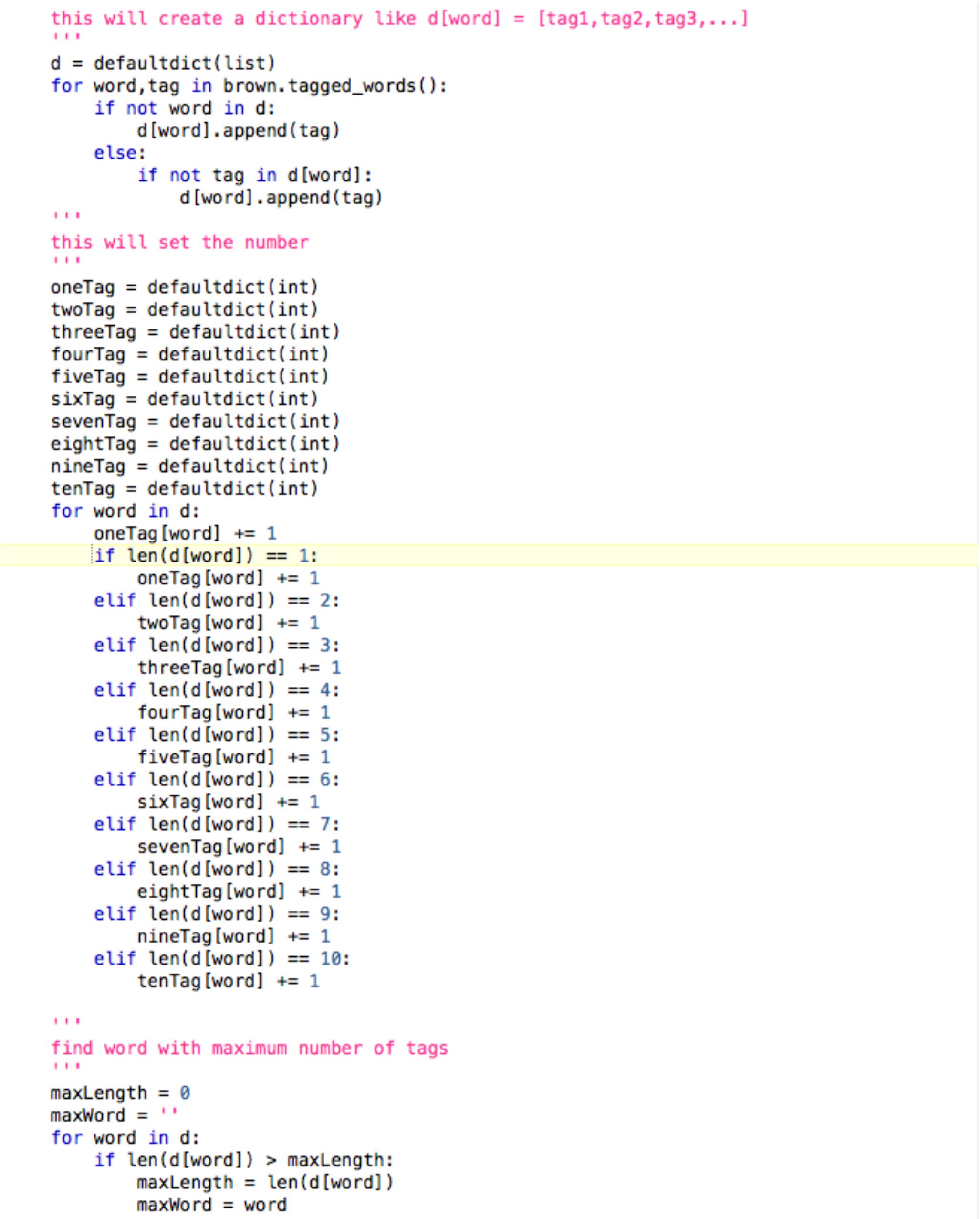
(b) NOUN, VERB, NOUN is more plausible linguistically because it can translate to Subject, VERB, Object. However, I do not believe that logic follows in every other language. So since the most likely sequence of POS is NOUN, PRO, VERB, This sequence must be somehow more plausible in the Klingon language.

(c) I believe it does. They both make Markov assumptions that next state only depends on the current state and independent of previous history.

5.

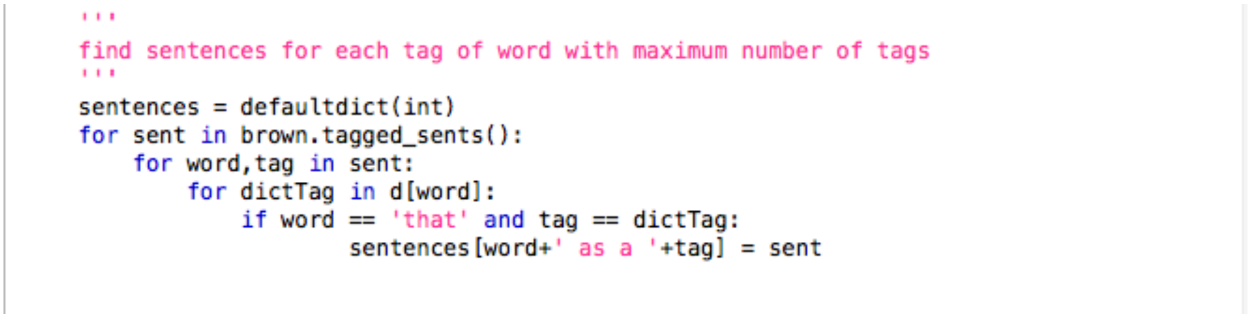
Based on researching the words, legh means “to see” and Yas means “Officer” but that doesn’t translate to my POS tag which is confusing. So the sentence might translate to “Human/Child watches officer”. The subject would be human/child.

code for 1.2:



```
this will create a dictionary like d[word] = [tag1,tag2,tag3,...]
'''
d = defaultdict(list)
for word,tag in brown.tagged_words():
    if not word in d:
        d[word].append(tag)
    else:
        if not tag in d[word]:
            d[word].append(tag)
'''
this will set the number
'''
oneTag = defaultdict(int)
twoTag = defaultdict(int)
threeTag = defaultdict(int)
fourTag = defaultdict(int)
fiveTag = defaultdict(int)
sixTag = defaultdict(int)
sevenTag = defaultdict(int)
eightTag = defaultdict(int)
nineTag = defaultdict(int)
tenTag = defaultdict(int)
for word in d:
    oneTag[word] += 1
    if len(d[word]) == 1:
        oneTag[word] += 1
    elif len(d[word]) == 2:
        twoTag[word] += 1
    elif len(d[word]) == 3:
        threeTag[word] += 1
    elif len(d[word]) == 4:
        fourTag[word] += 1
    elif len(d[word]) == 5:
        fiveTag[word] += 1
    elif len(d[word]) == 6:
        sixTag[word] += 1
    elif len(d[word]) == 7:
        sevenTag[word] += 1
    elif len(d[word]) == 8:
        eightTag[word] += 1
    elif len(d[word]) == 9:
        nineTag[word] += 1
    elif len(d[word]) == 10:
        tenTag[word] += 1

'''
find word with maximum number of tags
'''
maxLength = 0
maxWord = ''
for word in d:
    if len(d[word]) > maxLength:
        maxLength = len(d[word])
        maxWord = word
```

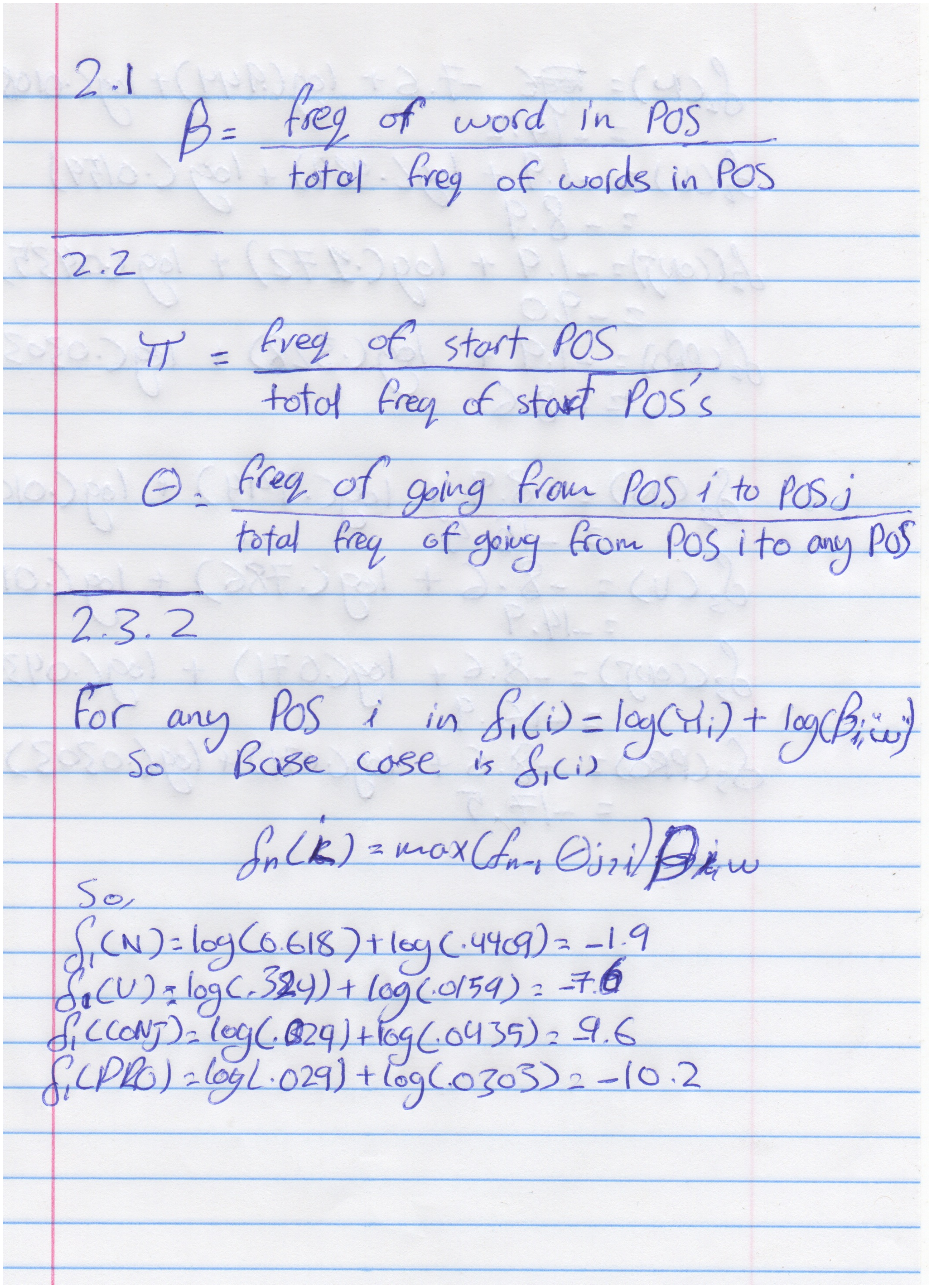


```
'''
find sentences for each tag of word with maximum number of tags
'''
sentences = defaultdict(int)
for sent in brown.tagged_sents():
    for word,tag in sent:
        for dictTag in d[word]:
            if word == 'that' and tag == dictTag:
                sentences[word+' as a '+tag] = sent
```

Anas Salamah

NLP Assignment 5
10/9/14

Steps for parts 2.1, 2.2 and 2.3.2



2.1

$$\beta = \frac{\text{freq of word in POS}}{\text{total freq of words in POS}}$$

2.2

$$\pi = \frac{\text{freq of start POS}}{\text{total freq of start POS's}}$$

$$\Theta = \frac{\text{freq of going from POS } i \text{ to POS } j}{\text{total freq of going from POS } i \text{ to any POS}}$$

2.3.2

For any POS i in $\delta_1(i) = \log(\pi_i) + \log(\beta_{i,\text{"}w\text{"}})$

So Base case is $\delta_1(i)$

$$\delta_n(k) = \max(\delta_{n-1}\, \Theta_{j,i})\, \beta_{k,w}$$

So,

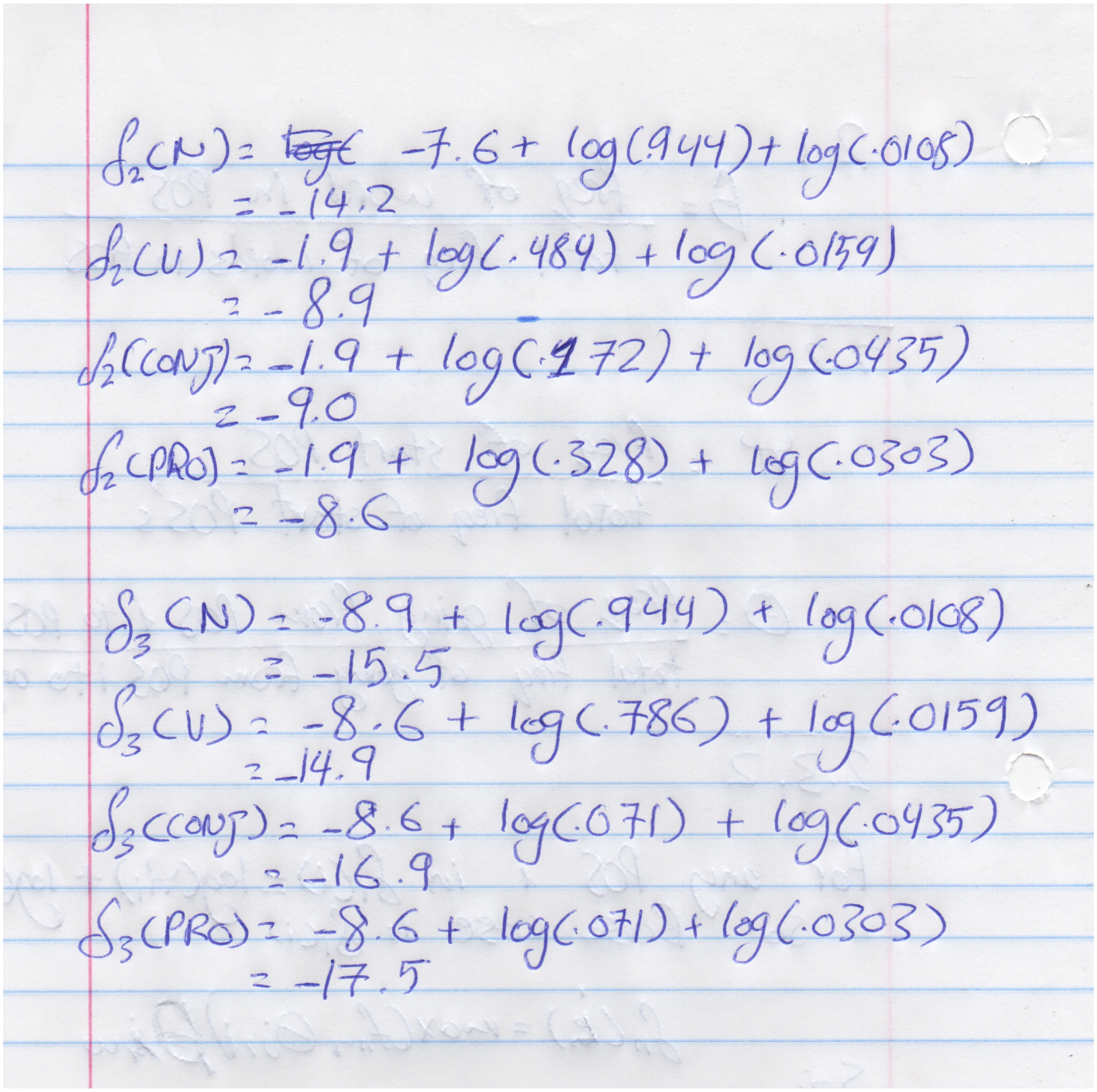
$\delta_1(N) = \log(0.618) + \log(.4409) = -1.9$

$\delta_1(V) = \log(.324) + \log(.0159) = -7.6$

$\delta_1(CONJ) = \log(.029) + \log(.0435) = -9.6$

$\delta_1(PRO) = \log(.029) + \log(.0303) = -10.2$

Anas Salamah NLP Assignment 5
10/9/14



$\delta_2(N) =$ ~~log(~~ $-7.6 + \log(.944) + \log(.0108)$
$= -14.2$

$\delta_2(V) = -1.9 + \log(.484) + \log(.0159)$
$= -8.9$

$\delta_2(CONJ) = -1.9 + \log(.172) + \log(.0435)$
$= -9.0$

$\delta_2(PRO) = -1.9 + \log(.328) + \log(.0303)$
$= -8.6$

$\delta_3(N) = -8.9 + \log(.944) + \log(.0108)$
$= -15.5$

$\delta_3(V) = -8.6 + \log(.786) + \log(.0159)$
$= -14.9$

$\delta_3(CONJ) = -8.6 + \log(.071) + \log(.0435)$
$= -16.9$

$\delta_3(PRO) = -8.6 + \log(.071) + \log(.0303)$
$= -17.5$