Midterm Exam for Natural Language Processing

March 9, 2017

Name: ____
Net ID _____________________

Instructions

There are 7 questions, each will be worth 15 points for a total of 105 points. You will have approximately 1:15 minutes to complete this test.

The test materials will include this printout and, optionally, one blank blue-covered booklet. I suggest that you fill in all answers directly on this printout. The blue booklet is available if you would like scrap paper, or if you run out of space on the test and need somewhere else to write your answers. You also have the option of putting all your answers in the blue booklet, if you prefer. Whatever you do, please make it clear. Write me notes about where to find particular questions if necessary. As long as I can find and understand your answers, it's fine. If you use the blue booklet, please write your name on this as well, so I can identify it as yours if it gets separated from this printout.

This test is an open book/open notes test: Please feel free to bring your text book, your notes, copies of class lectures and other reading material to the test. A calculator is also permitted and it is OK to look at materials on the web in order to read helpful information, being mindful of the time limit. Just don't use a program that solves a problem for you, e.g., do not find a part of speech tagger and run it if asked to manually annotate mark parts of speech – that WOULD be cheating.

Answer all questions on the test. If you show your work and you make a simple arithmetic mistake, but it is clear you knew how to do it, you will get partial credit.

Question 1. Write a regular expression to identify names of kings, queens and other names of people holding royal titles. Your expression should match person names accompanied by an adjacent royal title–the title should either: (a) immediately precede the name; or (b) immediately follow the name and a comma. The expression should match all the examples below and generalize to cover some other royal names. It should also be specific enough that it does not match most non-royal names. In particular, it should not match any part of the strings: a) *My dog Prince*, b) *Disney Princess bedroom set*, or c) *Queens, Prince, and Kings*

1. King David
2. Princess Mononoke
3. Queen Amidala
4. King Yoganarendra Malla
5. Emperor Sun Zhi
6. Empress Dowager Cixi
7. King George III
8. Duke Boleslaw Krzywousty
9. Diana, Princess of Wales
10. Henry Plantagenet, Duke of Normandy
11. Seth, Emperor of Azania
12. Prince Edward, Duke of Kent
13. George I, King of the Hellenes

```
### all on one line with no spaces between lines
    the first two lines are one disjunct and the
    third line is the second disjunct ###
(((King|Queen|Duke|Prince|Princess|Dutchess|Emperor|Empress) )?([A-Z][a-z.]*)
(( [A-Z][a-z]+)|( [IVX]*))*, (King|Queen|Duke|Prince|Princess|Dutchess|
Emperor|Empress)+ *of *(the *)?[A-Z][a-z]+)|
((King|Queen|Duke|Prince|Princess|Dutchess|Emperor|Empress)
( [A-Z][a-z.]*)+([IVX]*))
```

Tag	Description	Tag	Description
CC	Coordinating conjunction	RB	Adverb
CD	Cardinal number	RBR	Adverb, comparative
DT	Determiner	RBS	Adverb, superlative
EX	Existential there	RP	Particle
FW	Foreign word	SYM	Symbol
IN	Preposition or subordinating conjunction	TO	to
JJ	Adjective	UH	Interjection
JJR	Adjective, comparative	VB	Verb, base form
JJS	Adjective, superlative	VBD	Verb, past tense
LS	List item marker	VBG	Verb, gerund or present participle
MD	Modal	VBN	Verb, past participle
NN	Noun, singular or mass	VBP	Verb, non-3rd person singular present
NNS	Noun, plural	VBZ	Verb, 3rd person singular present
NNP	Proper noun, singular	WDT	Wh-determiner
NNPS	Proper noun, plural	WP	Wh-pronoun
PDT	Predeterminer	WP$	Possessive wh-pronoun
POS	Possessive ending	WRB	Wh-adverb
PRP	Personal pronoun	PU	Punctuation
PRP$	Possessive pronoun		

Table 1: **Penn Treebank POS tags**

Question 2. Manually process the following sentence in two ways, filling in the columns in the chart below:

> It was a shy nocturnal creature with the appearance of a medium-size dog, except for its abdominal pouch and dark stripes.[1]

I have tokenized this sentence and placed the tokens in the first column in the table below. Fill in the second column with PENN TREEBANK parts of speech (POS) tags, as per Table 1 (unlike conventional Penn Treebank Tags, all punctuation is marked *PU*). In the third column, enter a BIO tag indicating whether a token is beginning a noun group (B), inside a noun group (I) or outside of a noun group (O). Remember not to include right modifiers as noun groups are not full NPs. If you are uncertain about any part of speech assignment, include a short note why you chose the tag you did.

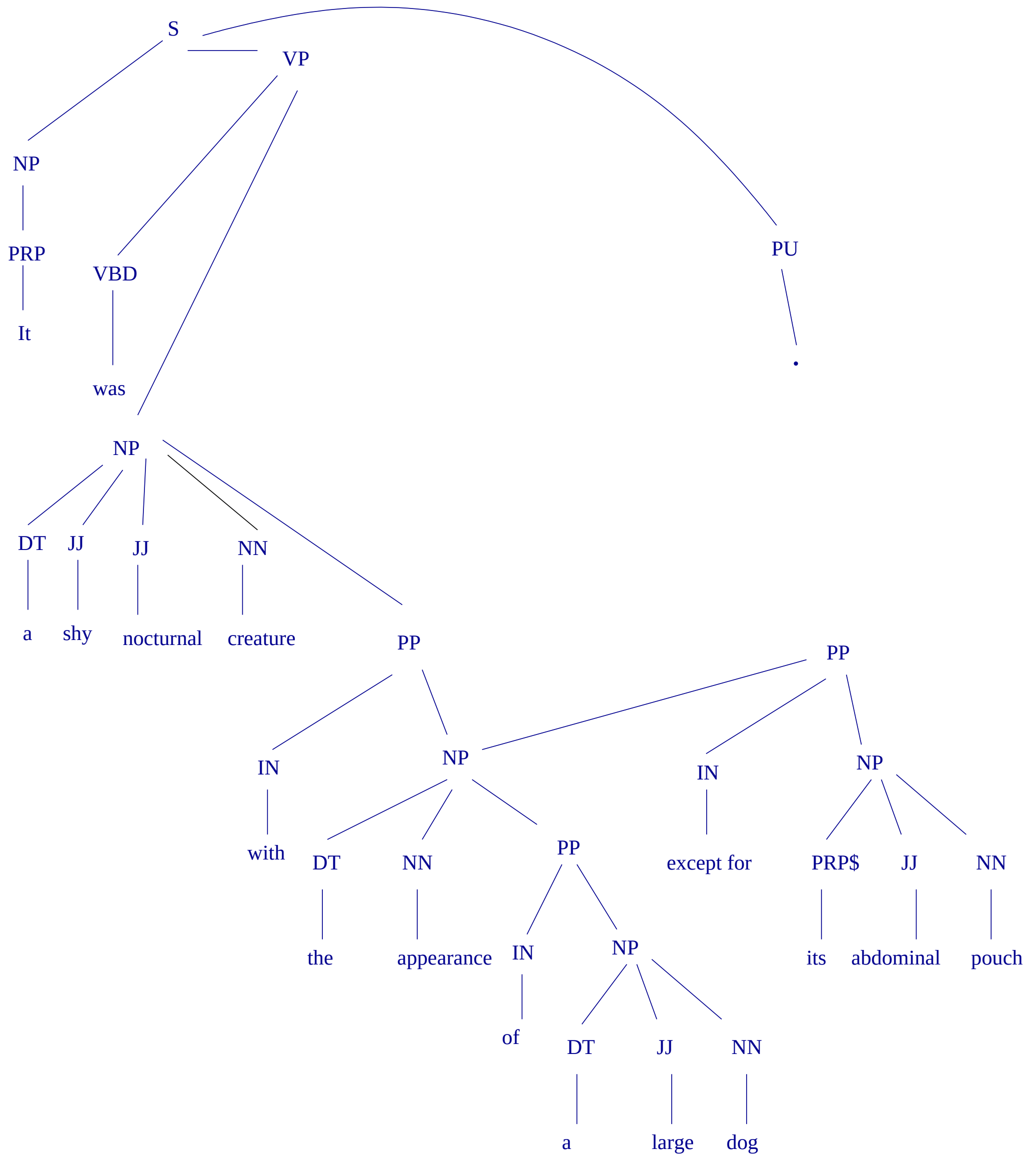
[1]Adapted from Wikipedia article about creatures known as a thylacines, aka, "Tasmanian Tigers."

Token	**POS Tag**	**BIO Tag**
It	PRP	B
was	VBD	O
a	DT	B
shy	JJ	I
nocturnal	JJ	I
creature	NN	I
with	IN	O
the	DT	B
appearance	NN	I
of	IN	O
a	DT	B
medium	JJ	I
size	NN	I
dog	NN	I
except	IN	O
for	IN	O
its	PRP$	B
abdominal	JJ	I
pouch	NN	I
and	CC	O
dark	JJ	B
stripes	NNS	I
.	PU	O

Question 3. Draw a Phrase Structure Tree analyzing the same sentence you analyzed in Question 2, i.e.,

> It was a shy nocturnal creature with the appearance of a medium-size dog, except for its abdominal pouch.

You should assume the same POS tags you used in Question 2. Assume that the words *except* and *for* together form a unit that "acts" like a single preposition. Note that the sentence has been shortened slightly (leaving out the words "*and dark stripes*")



Question 4: Fill in the CKY chart below for sentence

Hope springs eternal

assuming the rules below. Remember that the rows of the chart represent start positions and the columns represent end positions.

1. **S $\rightarrow$ NP VP**
2. **NP $\rightarrow$ NN**
3. **NP $\rightarrow$ NNS**
4. **NP $\rightarrow$ NNP**
5. **NP $\rightarrow$ NN NN**
6. **NP $\rightarrow$ NN NNS**
7. **VP $\rightarrow$ VB**
8. **VP $\rightarrow$ VBZ**
9. **VP $\rightarrow$ VBZ JJ**
10. **NN $\rightarrow$ hope**
11. **NNP $\rightarrow$ hope**
12. **VB $\rightarrow$ hope**
13. **NNS $\rightarrow$ springs**
14. **VBZ $\rightarrow$ springs**
15. **JJ $\rightarrow$ eternal**

	hope	*springs*	*eternal*
	1	2	3
0	NN, NNP, VB, NP, VP	NP, S	S
1	XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX	NNS, VBZ, NP, VP	VP
2	XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX	JJ

Question 5. For this question, compute the probability of the sentence

There was an old person from Rome

assuming the following language model, using frequencies of words in a copy of *Edward Lear's Book of Nonsense* available through Project Gutenberg:

- The probability of a sentence is the product of the probability of all the tokens in that sentence and the probability that the sentence ends (probability of an end_sentence token). Thus if N is the length of the sentence, multiply N+1 factors for the total probability.
- The probability of each token is computed based on the bigram, unigram and out of vocabulary frequencies found in the training corpus.
- The following backoff model is assumed:
  - Use the bigram probability of each token, given the previous token, if available. If the previous token is OOV, take the bigram of the current token given that the previous token is OOV.
  - Otherwise, use the unigram probability of the token, if available.
  - Otherwise, use the unigram OOV probability.

Use the following information to calculate these probabilities:

- Bigram frequencies for bigrams ending with tokens in the sentence. Each bullet lists bigrams beginning with the same first token. B_Sentence represents the beginning of the sentence and *oov* represents out of vocabulary words.[2]
  - **B_Sentence** + *oov* → 124; **B_sentence** + and → 35; **B_sentence** + but → 35; **B_sentence** + he → 37; **B_sentence** + she → 20; **B_sentence** + so → 19; **B_sentence** + that → 46; **B_sentence** + there → 113; **B_sentence** + till → 11; **B_sentence** + to → 11; **B_sentence** + when → 26; **B_sentence** + which → 25; **B_sentence** + who → 65; **B_sentence** + whose → 30
  - **there** + was → 113
  - **are** + *oov* → 3; **are** + of → 1; **are** + you → 2
  - **old** + *oov* → 3; **old** + derry → 1; **old** + lady → 4; **old** + man → 91; **old** + person → 42
  - **person** + of → 51; **person** + whose → 1
  - **from** + *oov* → 2; **from** + his → 1; **from** + the → 2; **from** + this → 1; **from** + turkey → 1
  - ***oov*** + ! → 13; ***oov*** + ***oov*** → 67; ***oov*** + , → 147; ***oov*** + . → 117; ***oov*** + ; → 69; ***oov*** + ? → 5; ***oov*** + a → 7; ***oov*** + all → 5; ***oov*** + and → 15; ***oov*** + by → 9; ***oov*** + E_sentence → 5; ***oov*** + from → 5; ***oov*** + her → 5; ***oov*** + his → 12; ***oov*** + in → 6; ***oov*** + of → 26; ***oov*** + old → 24; ***oov*** + that → 17; ***oov*** + the → 9; ***oov*** + to → 17; ***oov*** + was → 9; ***oov*** + with → 9; ***oov*** + young → 6;
- The unigram frequencies of each token, including *oov* and E_**sentence** (end of sentence). Unigram frequencies are: **B_sentence** → 688; **there** → 113; **are** → 6; **old** → 141; **person** → 52; **from** → 7; ***oov*** → 701; E_**sentence** → 688
- **There are a total of 5884 words (tokens) in the corpus.**

[2]For bigrams such that the first item is B_Sentence, only bigrams with frequency greater than 10 are included. For bigrams where the first item is *oov*, only bigrams with frequency greater than 5 are included.

Answer to Question 5

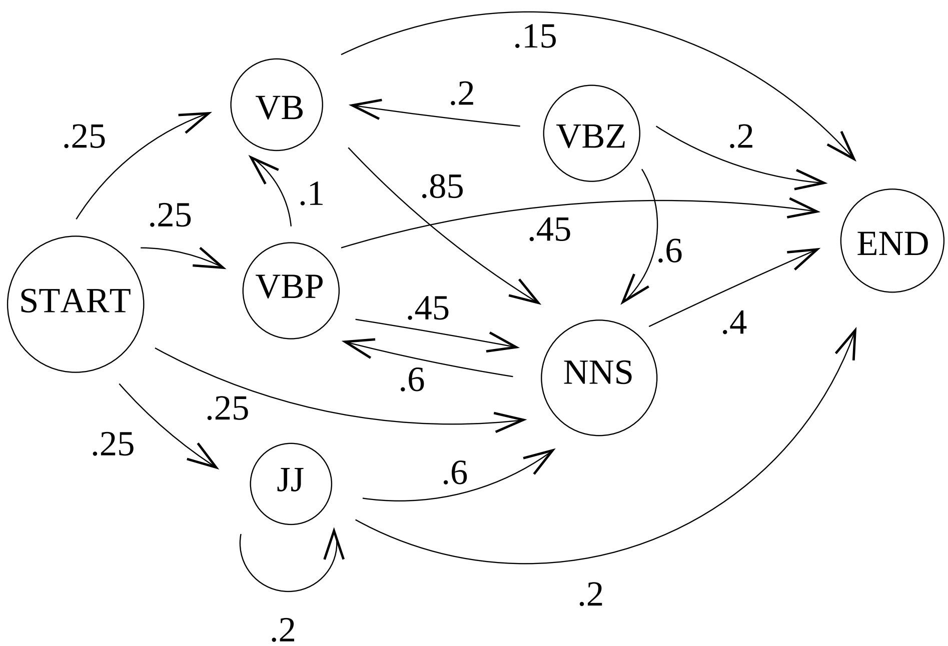
- B_sentence + there – 113/688 (sentence-initial *there* divided by number of sentences)
- there + was – 113/113 (all instances of *there* are followed by *was*)
- was + an – 67/701 (both are OOV, bigram of *oov* when followed byr *oov*)
- an + old – 24/701 (24 of the 701 instances of *oov* are followed by *old*)
- old + person – 42/141 (42 of the 141 instances of *old* are followed by *person*)
- person + from – 7/5884 (unigram for *from*)
- from + Rome – 2/7 (2 out of 7 of the instances of *from* are followed by **oov**)
- Rome + E_Sentence – 5/701 (5 out of 701 instances of *oov* are followed by E_sentence).

Total: $113/688 \times 1 \times 67/701 \times 24/701 \times 42/141 \times 7/5884 \times 2/7 \times 5/701 = 3.88 \times 10^{-10}$

Question 6. Using the Viterbi algorithm with the transition and likelihood probabilities below: (a) calculate the probability that the sequence of words *free ducks* will be assigned the parts of speech: VB NNS (as in the command telling someone to let some ducks free); (b) calculate the probability that the sequence will be assigned the parts of speech: JJ NNS, as in the noun phrase that refers to ducks that do not cost any money. **Hint: You do not have to fill out the entire table to calculate these two probabilities.**

Likelihood Probabilities		
POS	**free**	**ducks**
NNS	0	.0000835
VB	.000151	0
VBP	.0000800	0
VBZ	0	.0000461
JJ	.00158	0

Transition Probabilities



Viterbi Table				
	START	**free**	**ducks**	**END**
START				
NNS				
VB				
VBP				
VBZ				
JJ				
END				

Answer to Question 6

6a. Transitions: start $\rightarrow$ VB = .25; VB $\rightarrow$ NNS = .85; NNS $\rightarrow$ end = . 4
Likelihoods: VB *free* = .000151; NNS *ducks* = .0000835.
Total = $.25 \times .000151 \times .85 \times .0000835 \times .4 = 1.072 * 10^{-09}$

6b. Transitions: start $\rightarrow$ JJ = .25; JJ $\rightarrow$ NNS = .6; NNS $\rightarrow$ end = . 4
Likelihoods: JJ *free* = 00158; NNS *ducks* = .0000835.
Total = $.25 \times .00158 \times .6 \times .0000835 * .4 = 7.92 \times 10^{-09}$

Question 7. Computer precision, recall and F-measure given the following answer and system output:

Answer Key

1. King David
2. Princess Mononoke found
3. Queen Amidala found
4. King Yoganarendra Malla
5. Emperor Sun Zhi found
6. Empress Dowager Cixi found
7. King George III
8. Duke Boleslaw Krzywousty found
9. Diana, Princess of Wales found
10. Henry Plantagenet, Duke of Normandy
11. Seth, Emperor of Azania
12. Prince Edward, Duke of Kent found
13. George I, King of the Hellenes

System Output

1. Princess Mononoke correct
2. King Kong
3. Queen Amidala correct
4. Disney Princess Bedroom Set
5. Emperor Sun Zhi correct
6. Empress Dowager Cixi correct
7. Duke Boleslaw Krzywousty correct
8. Diana, Princess of Wales correct
9. Prince Edward, Duke of Kent correct
10. King Kullen Supermarket

Answer Question 7:

- 10 system answers, 13 items in answer key, 7 correct (intesection of system and answer key)
- Recall = 7/13 = 53.8
- Precision = 7/10 = 70.0
- F-measure = 2/((13/7) + (10/7)) = 60.9