Natural Language Processing
Final Exam
Nov. 14, 2022
Parker Smith

1. Given the sentences below, build a lexical translation model and show each training step's completion. Assume the model only trains for two iterations.
   1 - this course – este curso
   2 – text processing – procesamiento de texto

   Vocabularies: E = {this, course, text, processing}, S = {este, curso, procesamiento, de, texto}

   Uniform Probabilities:

t(este|this) = 1/5	t(curso|this) = 1/5	t(procesamiento|this) = 1/5	t(de|this) = 1/5	t(texto|this) = 1/5
t(este|course) = 1/5	t(curso|course) = 1/5	t(procesamiento|course) = 1/5	t(de|course) = 1/5	t(texto|course) = 1/5
t(este|text) = 1/5	t(curso|text) = 1/5	t(procesamiento|text) = 1/5	t(de|text) = 1/5	t(texto|text) = 1/5
t(este|processing) = 1/5	t(curso|processing) = 1/5	t(procesamiento|processing) = 1/5	t(de|processing) = 1/5	t(texto|processing) = 1/5

   Step 1a: Compute P(A, F|E)
   t(este|this) x t(curso|course) = 1/25
   t(este|course) x t(curso|this) = 1/25
   t(procesamiento|text) x t(de|processing) = 1/25
   t(procesamiento|text) x t(texto|processing) = 1/25
   t(de|text) x t(procesamiento|processing) = 1/25
   t(texto|text) x t(procesamiento|processing) = 1/25
   t(de|text) x t(texto|processing) = 1/25
   t(texto|text) x t(de|processing) = 1/25

   Step 1b: Normalize
   t(este|this) x t(curso|course) = (1/25)/(2/25) = 1/2
   t(este|course) x t(curso|this) = (1/25)/(2/25) = 1/2
   t(procesamiento|text) x t(de|processing) = (1/25)/(6/25) = 1/6
   t(procesamiento|text) x t(texto|processing) = (1/25)/(6/25) = 1/6
   t(de|text) x t(procesamiento|processing) = (1/25)/(6/25) = 1/6
   t(texto|text) x t(procesamiento|processing) = (1/25)/(6/25) = 1/6
   t(de|text) x t(texto|processing) = (1/25)/(6/25) = 1/6
   t(texto|text) x t(de|processing) = (1/25)/(6/25) = 1/6

Step 2: Compute expected counts of aligned word pairs.

t(este|this) = 1/2	t(curso|this) = 1/2	t(procesamiento|this) = 0	t(de|this) = 0	t(texto|this) = 0	total(this) = 1
t(este|course) = 1/2	t(curso|course) = 1/2	t(procesamiento|course) = 0	t(de|course) = 0	t(texto|course) = 0	total(course) = 1
t(este|text) = 0	t(curso|text) = 0	t(procesamiento|text) = 1/6+1/6	t(de|text) = 1/6+1/6	t(texto|text) = 1/6+1/6	total(text) = 1
t(este|processing) = 0	t(curso|processing) = 0	t(procesamiento|processing) = 1/6+1/6	t(de|processing) = 1/6+1/6	t(texto|processing) = 1/6+1/6	total(processing) = 1

M-Step: Compute the probability parameters by normalizing

t(este|this) = 1/2	t(curso|this) = 1/2	t(procesamiento|this) = 0	t(de|this) = 0	t(texto|this) = 0
t(este|course) = 1/2	t(curso|course) = 1/2	t(procesamiento|course) = 0	t(de|course) = 0	t(texto|course) = 0
t(este|text) = 0	t(curso|text) = 0	t(procesamiento|text) = 1/3	t(de|text) = 1/3	t(texto|text) = 1/3
t(este|processing) = 0	t(curso|processing) = 0	t(procesamiento|processing) = 1/3	t(de|processing) = 1/3	t(texto|processing) = 1/3

ITERATION 2

Step 1a: Compute P(A, F|E)
t(este|this) x t(curso|course) = 1/4
t(este|course) x t(curso|this) = 1/4
t(procesamiento|text) x t(de|processing) = 1/9
t(procesamiento|text) x t(texto|processing) = 1/9
t(de|text) x t(procesamiento|processing) = 1/9
t(texto|text) x t(procesamiento|processing) = 1/9
t(de|text) x t(texto|processing) = 1/9
t(texto|text) x t(de|processing) = 1/9

Step 1b: Normalize
t(este|this) x t(curso|course) = (1/4)/(2/4) = 1/2
t(este|course) x t(curso|this) = (1/4)/(2/4) = 1/2
t(procesamiento|text) x t(de|processing) = (1/9)/(6/9) = 1/6
t(procesamiento|text) x t(texto|processing) = (1/9)/(6/9) = 1/6
t(de|text) x t(procesamiento|processing) = (1/9)/(6/9) = 1/6
t(texto|text) x t(procesamiento|processing) = (1/9)/(6/9) = 1/6
t(de|text) x t(texto|processing) = (1/9)/(6/9) = 1/6
t(texto|text) x t(de|processing) = (1/9)/(6/9) = 1/6

Step 2: Compute expected counts of aligned word pairs.

t(este|this) = 1/2	t(curso|this) = 1/2	t(procesamiento|this) = 0	t(de|this) = 0	t(texto|this) = 0	total(this) = 1
t(este|course) = 1/2	t(curso|course) = 1/2	t(procesamiento|course) = 0	t(de|course) = 0	t(texto|course) = 0	total(course) = 1
t(este|text) = 0	t(curso|text) = 0	t(procesamiento|text) = 1/6+1/6	t(de|text) = 1/6+1/6	t(texto|text) = 1/6+1/6	total(text) = 1
t(este|processing) = 0	t(curso|processing) = 0	t(procesamiento|processing) = 1/6+1/6	t(de|processing) = 1/6+1/6	t(texto|processing) = 1/6+1/6	total(processing) = 1

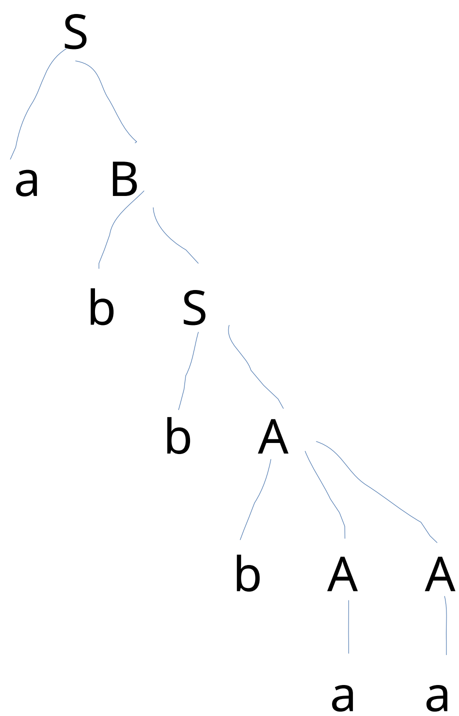
M-Step: Compute the probability parameters by normalizing

t(este|this) = 1/2	t(curso|this) = 1/2	t(procesamiento|this) = 0	t(de|this) = 0	t(texto|this) = 0
t(este|course) = 1/2	t(curso|course) = 1/2	t(procesamiento|course) = 0	t(de|course) = 0	t(texto|course) = 0
t(este|text) = 0	t(curso|text) = 0	t(procesamiento|text) = 1/3	t(de|text) = 1/3	t(texto|text) = 1/3
t(este|processing) = 0	t(curso|processing) = 0	t(procesamiento|processing) = 1/3	t(de|processing) = 1/3	t(texto|processing) = 1/3

2. Is the below gammer ambiguous? Provide and example and show parse tree to validate your answer.

   S→ a B | b A
   A → a | a S | b A A
   B → b | b S | a B B

   No, this grammar is unambiguous. An example is **abbbaa**. A parse tree is shown below. This is the only possible way to parse this sentence, therefore this grammar is unambiguous.



3. Describe the issue of BLEU score: The issue with BLEU score is that it only measures direct word-to-word similarity. There are no considerations of phrases or synonyms, and nonsensical language, if ordered in a way similar to the reference translation, can be scored very highly.

   Evaluate using BLEU score where N = 2.
   Candidate 1: It is a guide to action which ensures that the military always obeys the commands of the party.
   Candidate 2: It is to ensure the troops forever hearing the activity guidebook that party

direct.

Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.
Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.
Reference 3: It is the practical guide for the army always to heed the directions of the party.

Candidate 1 unigram precision: 17/18
Candidate 1 bigram precision: 10/17
Candidate 1 BLEU score: $(17/18*10/17)^{1/2}=0.7454=74.54\%$

Candidate 2 unigram precision: 8/14
Candidate 2 bigram precision: 1/13
Candidate 2 BLEU score: $(8/14*1/13)^{1/2}=0.2097=20.97\%$

Candidate 1 is clearly the better translation with a 74.54% BLEU score vs a 20.97% BLEU score for Candidate 2.

4. Given the grammar below, parse the given sentences using the CKY algorithm and show your derived table indicating the sentence symbol, S. Also, show the parse tree for both sentences.

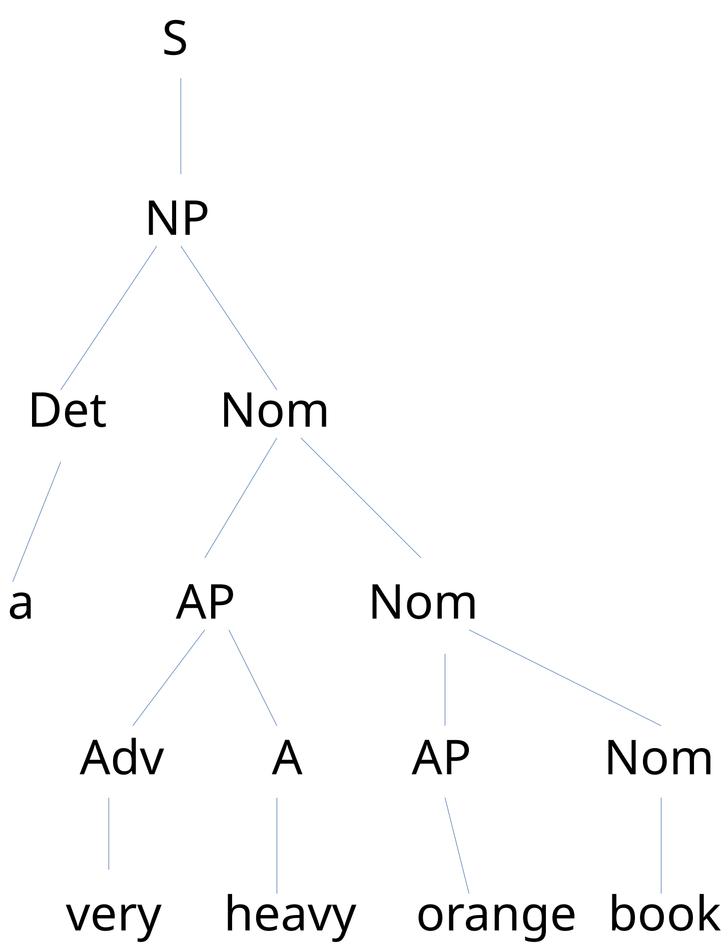
   S → NP
   NP → Det Nom
   Nom → AP Nom
   AP → Adv A
   Det → a | an
   Adv → very | extremely
   AP → heavy | orange | tall
   A → heavy | orange | tall | muscular
   Nom → book | orange | man

   Sentence 1: a very heavy orange book
   Sentence 2: a very tall extremely muscular man

Sentence 1 CKY Derived Table:

	1	2	3	4	5
0	Det (a)				S → NP → Det Nom
1		Adv (very)	AP → Adv A		Nom → AP Nom
2			AP, A (heavy)		
3				AP, A, Nom (orange)	Nom → AP Nom
4					Nom (book)

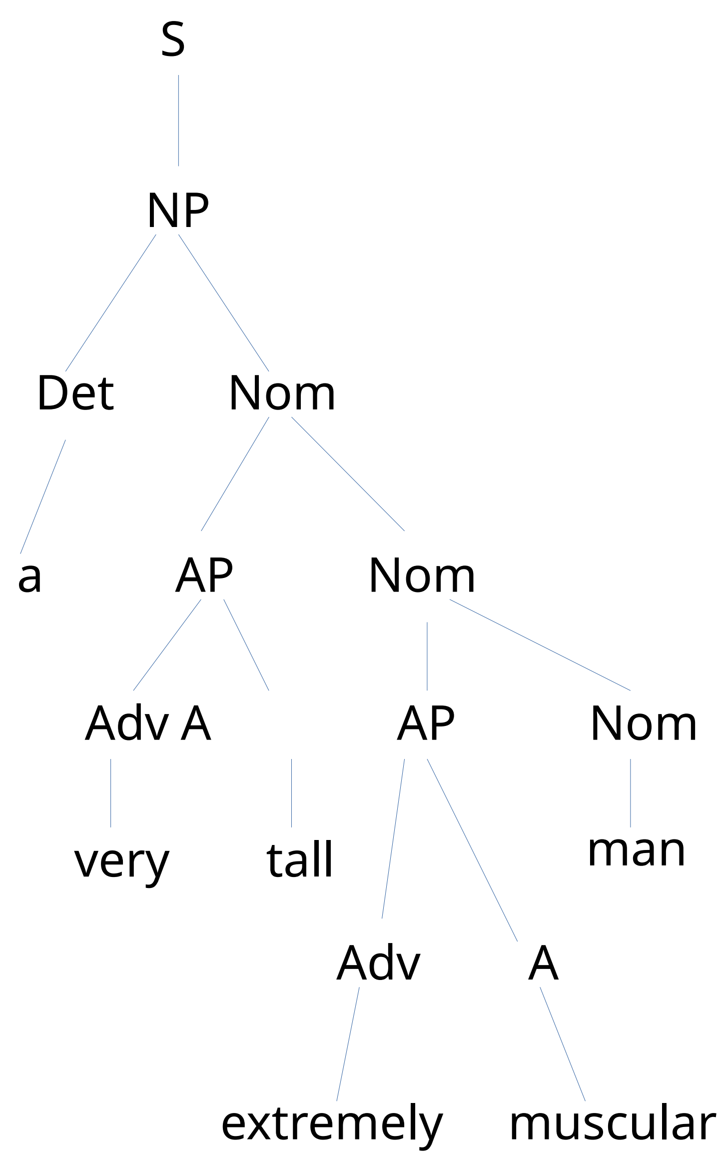
Sentence 1 Parse Tree:



Sentence 2 CKY Derived Table:

	1	2	3	4	5	6
0	Det (a)					S → NP → Det Nom
1		Adv (very)	AP → Adv A			Nom → AP Nom
2			AP, A (tall)			
3				Adv (extremely)	AP → Adv A	Nom → AP Nom
4					A (muscular)	
5						Nom (man)

Sentence 2 Parse Tree:



5. Write down the differences between attachment ambiguity and coordination ambiguity.

   Attachment ambiguity is where a parse tree node can be attached to more than one location while still staying within the rules of the tree, therefore changing the parsed meaning of the sentence. Coordination ambiguity is where different phrases connected with a coordinating conjunction can create trees with different meanings depending on the way the sentence is parsed.

6. Compute the performance of two summarizer systems using ROUGE-1 and ROGUE-2 precision, recall, and f1 score metrics. Based on the comparison, which one of the ROUGE metrics would you use to evaluate your system performance?

   S1 Summary: neymar scored his side's second goal with a curling free kick, and 15 minutes to play in the 2-2 draw at sevilla on saturday night, according to reports in spain.

   S2 Summary: barcelona's neymar substituted in 2-2 draw at sevilla on saturday night, spain's kamui kobayashi claims a late free kick in the champions league after his second goal with the score.

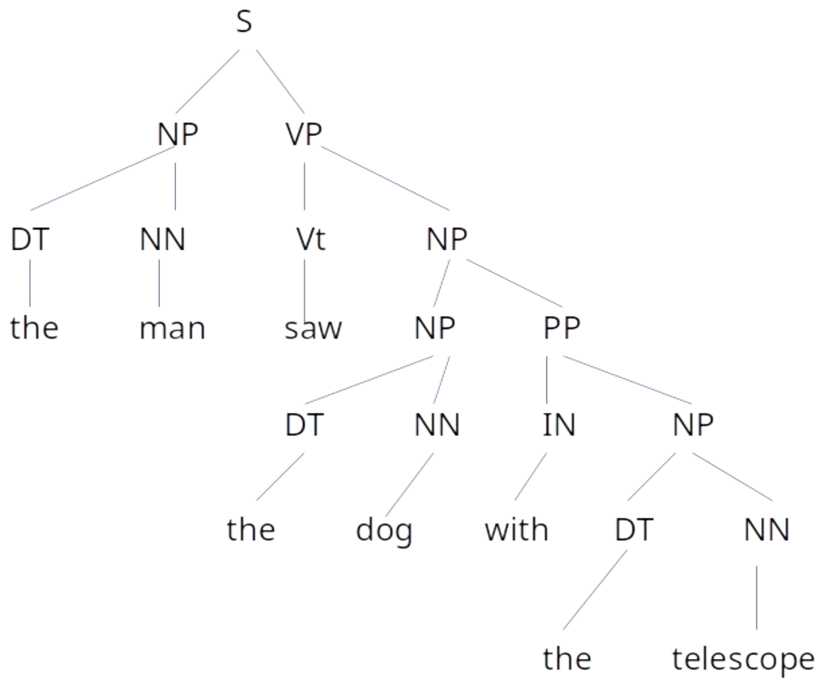
   Reference summary: neymar was taken off with barcelona 2-1 up against sevilla. the brazil captain was visibly angry, and barca went on to draw 2-2. neymar has been replaced 15 times in 34 games this season. click here for all the latest barcelona news.

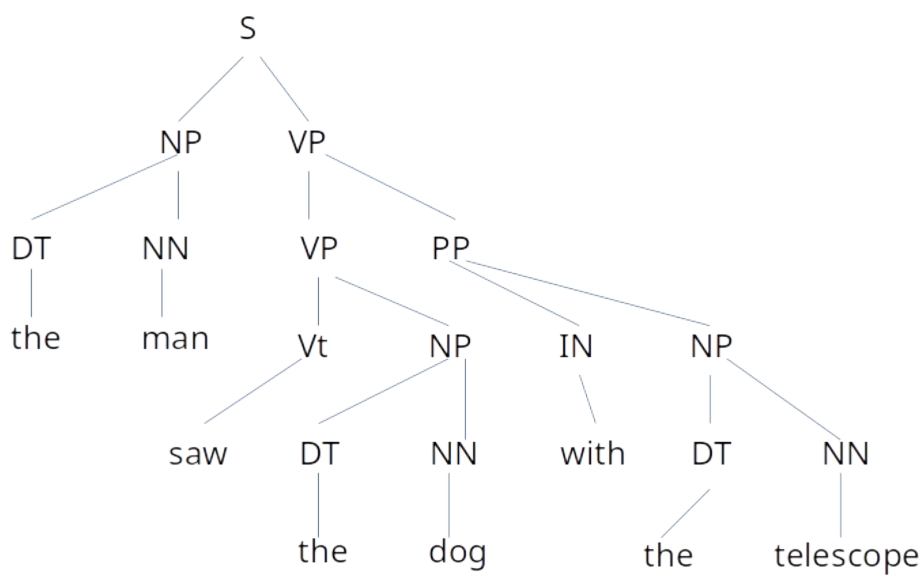
   ROUGE-1:
   S1 recall: 13 / 42 = 0.3095
   S1 precision: 13 / 30 = 0.4333
   S1 F1-score: 0.3611
   S2 recall: 10 / 42 = 0.2381
   S2 precision: 10 / 30 = 0.3333
   S2 F1-score: 0.2778

   ROUGE-2:
   S1 recall: 0/ 41
   S1 precision: 0/ 29
   S1 F1-score: 0
   S2 recall: overlapping 0 / 41
   S2 precision: 0 / 29
   S2 F1-score: 0

   Based on this comparison, I would prefer ROUGE-1 as my metric to evaluate my system's performance. I would choose this metric because this summarization system is an abstract summarizer and therefore does not always contain the same words as the reference summary. This makes finding bi-grams much less likely.

7. Given a grammar, show that the grammar is ambiguous and produced two different parse trees for the following sentence: **the man saw the dog with the telescope**





8. Evaluate and provide precision, recall, and F1 scores.

Precision: 3 / 7 = 0.4286
Recall: 3 / 8 = 0.3750
F1 scores: 0.3997

9. In transition-based parsing, we see dependency structures were provided, then why do we need to parse the sentence while given the structures?

   If we parse the sentence while given the structure, we can use the result of the parse as a form of training for the parsing algorithm. Afterword, if a sentence needs to be parsed which does not have a structure provided, we can use the trained parser to predict which transitions need to be applied to accurately parse the sentence.

10. What is CNF? Why is Chomsky's Normal Form used?

    CNF stands for Chomsky's Normal Form. It is a form of context free grammar where all rules are in one of two forms: A → BC where A, B, and C are non-terminals, or A → a where A is a non-terminal and a is a terminal. CNF is used because all grammar rules in CNF are context-free. In addition, grammar in CNF can use the CKY parsing algorithm to efficiently parse the grammar.

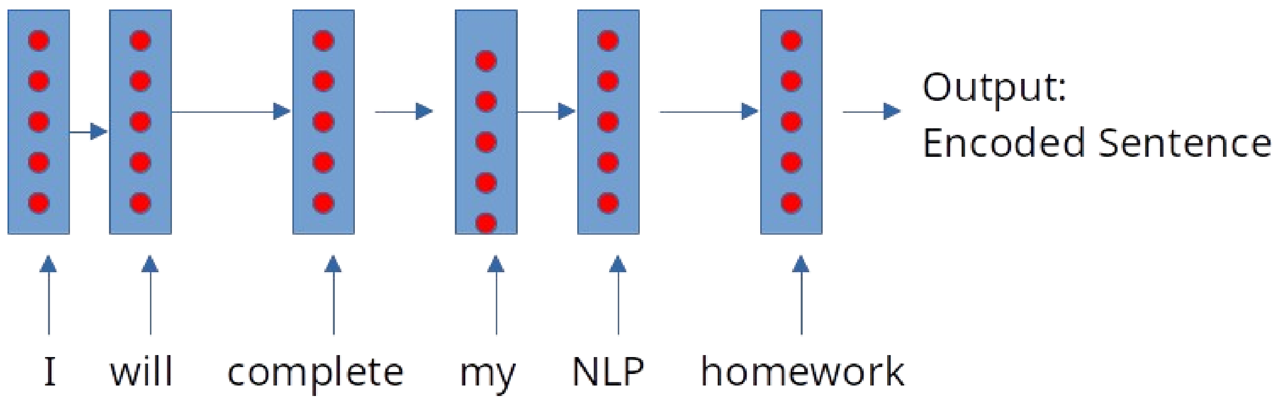
    Convert the following CFG to CNF:
    S → a X b X
    X → a Y | b Y | null
    Y → X | c

    Answer:
    S → AX BX
    X → A Y
    X → B Y
    X → null
    Y → A Y
    Y → B Y
    Y → c
    Y → null
    A → a
    B → b
    AX → A X
    BX → B X

11. How would you encode a sentence using a deep neural network? Show details architecture of your network with an example sentence. What is the bottleneck problem of the sequence-to-sequence model, and how would you get rid of it?

I would use a Recurrent Neural Network to encode a sentence using a deep neural network. This results in the sequence-to-sequence model of encoding. The recurrent neural network is flexible and can work on any length sequence to provide an encoded source input.

Example: **I will complete my NLP homework**



The bottleneck problem of the sequence-to-sequence model is that each sequence of the RNN needs to capture all of the information from the source sentence. This is difficult to do without help, therefore a way to solve this problem is through attention. Attention emphasizes focus on the most important part of the input sentence for each part of the sequence.