

CS 6320.002: Natural Language Processing
Fall 2020

Homework 3 — 45 points
Issued 28 Sept. 2020
Due 11:59pm 12 Oct. 2020

Deliverables: Answers can be typed directly into Gradescope. LaTeX can be hand-typed or generated using Mathpix Snip. See the assignment guide for more details.

What does it mean to “show your work?” Write out the math step-by-step; we should be able to clearly follow your reasoning from one step to another. (You can combine “obvious” steps like simplifying fractions or doing basic arithmetic.) The point of showing your work is twofold: to get partial credit if your answer is incorrect, and to show us that you worked the problem yourself and understand it. We will deduct points if steps are missing.

1 Constituent Parsing — 20 points

The problems in this section are based on the material covered in Week 6.

1.1 Context Free Grammars

A. Suppose we have the following CFG:

S	→	NP VP
VP	→	VB NP
NP	→	NP and NP
NP	→	JJ NP
NP	→	NN
JJ	→	black
NN	→	I
NN	→	cats
NN	→	dogs
VB	→	like

How many possible parse trees are there for the sentence “I like black cats and dogs”? Show the derivations for each possible parse, ie. the rules used and the order they are used in.

B. Suppose we have the following CFG:

S	→	NP VP
NP	→	DT NN
NP	→	DT NNS
VP	→	VB NP
DT	→	the
NN	→	cat
NN	→	dog
NNS	→	cats
NNS	→	dogs
VB	→	see
VB	→	sees

This grammar can generate ungrammatical sentences, such as “the dog see the cat” and “the dogs sees the cat.” What rule(s) would you need to add or change so that the grammar generates only grammatical sentences? List all the rules in your new grammar.

1.2 Probabilistic Context Free Grammars

A. Suppose we have the following PCFG:

S	→	NP VP	1.0
NP	→	DT JJ NN	1.0
VP	→	VB	1.0
DT	→	the	1.0
JJ	→	happy	0.5
JJ	→	ε	0.5
NN	→	cat	1.0
VB	→	sat	1.0

Convert this PCFG into Chomsky normal form. If you need new non-terminal symbols, use X , Y , and Z . Make sure you do not change the probabilities of sentences generated by this grammar.

B. Suppose we have the following PCFG:

S	→	NP VP	1.0
NP	→	DT NN	1.0
VP	→	MD VP	0.5
VP	→	VB NP	0.5
DT	→	the	1.0
NN	→	cat	0.5
NN	→	can	0.5
MD	→	can	1.0
VB	→	see	1.0

Use the CKY algorithm to find the highest probability parse of the sentence “The cat can see the can”. Show your work, ie. the dynamic programming table. (Note that CKY for probabilistic PCFGs uses a three-dimensional dynamic programming table. Instead

of trying to format a 3D table, simply use a 2D table, as in CKY for non-probabilistic CFGs, and show both the rule(s) being used and the associated probability in each table entry, as we do in lecture.)

2 Dependency Parsing — 15 points

The problems in this section are based on the material covered in Week 7.

Consider the sentence “I think I like cats.”

A. Parse this sentence by hand and list the dependency relations. You can use arrows, as we do in lecture, or simply list pairs of (head, dependent).

B. Parse this sentence using shift-reduce parsing, the arc-standard transition system, and no arc labels. You can assume you have a perfect oracle that tells you the correct action to take at each time step. Show the stack, buffer, action, and relation set at each time step (you can use a table, as we do in lecture, or simply show a series of lists).

C. Parse this sentence again, this time using the arc-eager transition system. Show your work as before.

3 Machine Translation — 10 points

The problems in this section are based on the material covered in Week 8.

Suppose we have a training corpus of three sentences:

1.	the	dog	2.	little	dog	3.	black	dog
	le	chien		petit	chien		chien	noir

Now suppose we have a word-level statistical machine translation model with the parameters $q(j|i, l, m)$ and $t(f_i|e_j)$ initialized uniformly using this corpus.

A. Perform one iteration of expectation maximization to update the parameters using the training corpus. State the values of the q 's and t 's at the end of this iteration, plus the intermediate values you calculated for $p(a_i = j|f_i, e_j)$. You can simply state the values; no need to show the arithmetic.

B. Perform a second iteration of expectation maximization to update the parameters again. As before, state the values of the q 's and t 's at the end of this iteration, plus the intermediate values you calculated for $p(a_i = j|f_i, e_j)$.