1.a):

Step	Stack	Buffer	New dep	Transition
0	[ROOT]	[To, ask, those, questions, is, to, answer, them]		
1	[ROOT, To]	[ask, those, questions, is, to, answer, them]		SHIFT
2	[ROOT, To, ask]	[those, questions, is, to, answer, them]		SHIFT
3	[ROOT, ask]	[those, questions, is, to, answer, them]	ask $\overset{mark}{\rightarrow}$ To	LEFT-ARC
4	[ROOT, ask, those]	[questions, is, to, answer, them]		SHIFT

step	stack	buffer	New dep	transition
5	[ROOT, ask, those, questions]	[is, to, answer, them]		SHIFT
6	[ROOT, ask, questions]	[is, to, answer, them]	Questions →(det) those	LEFT-ARC
7	[ROOT, ask]	[is, to, answer, them]	Ask → (dobj) questions	RIGHT-ARC
8	[ROOT, ask, is]	[to, answer, them]		SHIFT
9	[ROOT, is]	[to, answer, them]	Is → (csubj)ask	LEFT_ARC
10	[ROOT, is, to]	[answer, them]		SHIFT
11	[ROOT, is, to, answer]	[them]		SHIFT
12	[ROOT, is, answer]	[them]	Answer →(mark) to	LEFT_ARC
13	[ROOT, is, answer, them]	[]		SHIFT
14	[ROOT, is, answer]	[]	Answer →(dobj) them	RIGHT_ARC
15	[ROOT, is]	[]	Is →(xcomp) answer	RIGHT-ARC
16	[ROOT]	[]	ROOT →(root) is	RIGHT-ARC

b) if we don’t consider ROOT in the word count, then we need 2n steps. There are 9 words in the given sentence and we needed 16 = 2*8 steps to parse it. The formula 2n comes from: 1. Each word would need to be transitioned from the buffer to stack exactly once; 2. Each word would need to perform either a left arc or a right arc to be cleared from the stack.
g)

GPT-4

J

perform dependency parsing for "To ask those questions is to answer them"

Dependency parsing aims to represent a sentence in terms of its grammatical relations between words. Here's a breakdown of the dependencies in the sentence "To ask those questions is to answer them":

1. "To ask those questions" is an infinitival clause that serves as the subject of the main clause.
   - "ask" is the main verb of this clause.
   - "To" is an auxiliary indicating the infinitive form.
   - "those" is a determiner modifying "questions".
   - "questions" is the direct object of "ask".
2. "is" is the main verb of the main clause.
3. "to answer them" is an infinitival clause that serves as the predicate complement.
   - "answer" is the main verb of this clause.
   - "To" is an auxiliary indicating the infinitive form.
   - "them" is the direct object of "answer".

In a dependency tree, this structure would be represented as:

In a dependency tree, this structure would be represented as:

csharp Copy code

```
        is
      /    \
   ask      answer
    |          |
questions    them
    |
 those
```

Note: This is a simplified representation, and an actual dependency tree would include more detailed relations and finer-grained distinctions.

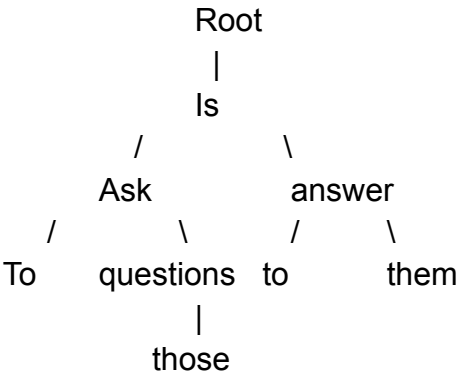
Oct 5, 2023, GPT4.

2 a)
The transition based mechanism is one way - moving words from buffer to the stack, and can not go back to a word once it has been removed from stack due to left or right arcs. Left and right arcs may be insufficient in the case of a non-projective tree, as they only look at the second last and the last elements in the stack.

b)

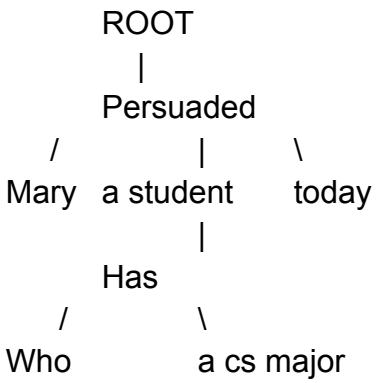
c)

Tree 1:



word	subsequent		Gaps (k+1)	Gap degree
To	[]	[to]	0	0
Ask	[To, questions, those]	[To], [Ask those questions]	2	1
those	[]	[those]	0	0
questions	[those]	[those questions]	0	0
is	[To, ask, questions, those, answer, to, them]	[To ask those questions], [is to answer them]	2	1
to	[]	[to]	0	0
answer	[to, them]	[to], [answer them]	2	1
them	[]	[them]	0	0

Tree2:



word	subsequent	Split to	Gaps
Mary	[]	[Mary]	0
Persuaded	[Mary, today, a student, has, who, a cs major]	[Mary], [persuaded a student today who has a cs major]	1
A student	[has, who, a cs major]	[a student who has a cs major]	0
today	[]	[today]	0
has	[who, a cs major]	[who], [has a cs major]	1
who	[]	[who]	0
A cs major	[]	[a cs major]	0

Gap degree for both trees is 1.

e)

(used latex for this part as it's too messy to type)

From wikipedia, uniform distribution has expected value $\mu = \frac{1}{2}(a+b)$, variance $\sigma^2 = \frac{1}{12}(b-a)^2$.
from μ :

$$\frac{1}{2}(a+b) = 0$$

$$a+b = 0$$

$$a = -b$$

from σ^2 :

$$\frac{1}{12}(b-a)^2 = \frac{2}{m}$$

substitute $a = -b$:

$$\frac{1}{12}(b-(-b))^2 = \frac{2}{m}$$

$$\frac{1}{12}(2b)^2 = \frac{2}{m}$$

$$4b^2 = \frac{24}{m}$$

$$b = \sqrt{\frac{6}{m}}$$

find a with $a = -b$:

$$a = -\sqrt{\frac{6}{m}}$$

```
From wikipedia, uniform distribution has expected value
$\mu$ = $\frac{1}{2} (a+b)$, variance $\sigma^2 =
\frac{1}{12}(b-a)^2$.
from $\mu :$
$$\frac{1}{2} (a+b) = 0$$
$$ a+b = 0$$
$$ a = -b$$
from $\sigma^2 :$
$$\frac{1}{12}(b-a)^2 = \frac{2}{m}$$
substitute $a = -b$:
$$\frac{1}{12}(b-(-b))^2 = \frac{2}{m}$$
$$\frac{1}{12}(2b)^2 = \frac{2}{m}$$
$$4b^2 = \frac{24}{m}$$
$$b = \sqrt{\frac{6}{m}}$$
find a with $a = -b$:
$$a = -\sqrt{\frac{6}{m}}$$
\vspace{\baselineskip}
```

g) The reason that D_A can not be left alone in arc scoring is that, in arc scoring we are trying to decide if there is a relationship between any possible heads and possible dependencies. Thus we need to look at the relationship between H_A and D_A, and add bias based on the head, H_A. Having a term just for D_A would mean assigning a score to a word based solely on its potential as a dependent, without looking at other words that it might depend on.

In label scoring, we are trying to decide the type of an existing relationship. Aside from the interaction between D_L and H_L, it is better to calculate the score for each term D_A and H_A separately, as they each may provide independent information on the type of their relationship.

h) Dependency parsing is different from standard classification problems, since we are predicting the syntax relationship between words in a sentence, instead of only determining the classification of the input. We need to capture the interaction between all words to determine pairs of heads and deps. To do so we need Biaffine transformation, to 1. Get weighted features for each input (D_A, H_A, etc), 2. consider pairwise interactions between all features of deps and heads and get scoring. This is where the multiplications come from.

j) A MST tree structure ensures that there is only a single root, there's no loop, and all nodes are connected, which are essential constraints for dependency parsing. If we use argmax (without other careful regulations), it may find more than 1 root, or connect words in a loop, or leave some words out. When finding labels, we could use argmax as the arcs are already found, and labels don't have such constraints, they don't affect each other as arcs do.

CSC 485H/2501H, Fall 2023: Assignment 1

Family name: Zhai

First name: Xueqing

Student #: 1006962413

Date: Oct 6, 2023

I declare that this assignment, both my paper and electronic submissions, is my own work, and is in accordance with the University of Toronto Code of Behaviour on Academic Matters and the Code of Student Conduct.

Signature: 

Note: I used the chu-liu-edmonds algorithm implementation found on an online resource, and I labeled the part and put on the link in the file. I wanted to make sure other parts of my code works (otherwise they'll show error). I have no intention to commit academic offence, please deduct mark for that part.