Natural Language Processing (COS4861)

Adriaan Louw (53031377)

August 13, 2018

1 Question 1

1.1 Question 1.1

The Markov assumption is that for a model the future state of the model depends only on the current state of the model and not any previous state. This assumption is used in bigrams. Where the probability that a word will appear is dependant only on the previous word i.e. $P(w_n|w_{n-1})$. This is not true for n-grams where $n > 2$ since then the probability will depend on more than just the previous word. For example in the sentence "The cat sat on the mat" we will be using a bigram model if we wanted to determine the probability that the word "on" came after "sat" i.e. P(on—sat). Thus we are only looking at the previous word. Using a trigram P(on—cat sat) we will be looking at 2 previous "states" or words and thus not follow the markov assumption.

1.2 Question 1.2

Please run MLE.py and please ensure that python 3.7 is installed. It can be found at https://www.python.org/downloads/

1.3 Question 1.3

I included the word "tale" in the input set to indicate an example of sparse data. Because we are using a data set there is the possibility that we want the MLE to find the most likely word following a word that is not in the training set.This makes it difficult to use an MLE on a different data set than the one it is trained on. To overcome this limitation we can use Laplace smoothing. Laplace smoothing adds one to the number of each occurrence of a the n-gram. (In this case we will be using a bigram.) Where normally the probability of finding the bigram is

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{1}$$

That is the number of occurrences of the bigram divided by the number of occurrences of w_{n-1}. This becomes

$$P^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V} \tag{2}$$

where V is the total vocabulary size. Coming back to our example of the word "tale" here the vocabulary will be increased by 1 and there will be an occurrence of each word in the vocabulary following "tale" added to the list of bigrams. This way the probability of any word following lame will not be 0.

2 Question 2

For $V_2(1)$:

$$\begin{aligned} V_1(1)P(PPSS \mid PPSS) &= (0.025)(0.00014) \\ &= 0.0000035 \end{aligned} \tag{3}$$

$$\begin{aligned} V_1(2)P(PPSS \mid VB) &= (0)(0.007) \\ &= 0 \end{aligned} \tag{4}$$

$$\begin{aligned} V_1(3)P(PPSS \mid TO) &= (0)(0) \\ &= 0 \end{aligned} \tag{5}$$

$$\begin{aligned} V_1(4)P(PPSS \mid NN) &= (0)(0.0045) \\ &= 0 \end{aligned} \tag{6}$$

$$\begin{aligned} V_2(1) = max(0.0000035, 0, 0, 0)P(want \mid PPSS) \\ = (0.0000035)(0) \\ = 0 \end{aligned} \tag{7}$$

For $V_2(3)$:

$$\begin{aligned} V_1(1)P(TO \mid PPSS) &= (0.025)(0.00079) \\ &= 0.00001975 \end{aligned} \tag{8}$$

$$\begin{aligned} V_1(2)P(TO \mid VB) &= (0)(0.035) \\ &= 0 \end{aligned} \tag{9}$$

$$\begin{aligned} V_1(3)P(TO \mid TO) &= (0)(0) \\ &= 0 \end{aligned} \tag{10}$$

$$\begin{aligned} V_1(4)P(TO \mid NN) &= (0)(0.016) \\ &= 0 \end{aligned} \tag{11}$$

$$\begin{aligned} V_2(3) = max(0.00001975, 0, 0, 0)P(want \mid TO) \\ = (0.00001975)(0) \\ = 0 \end{aligned} \tag{12}$$

For $V_2(4)$:

$$\begin{aligned} V_1(1)P(NN \mid PPSS) &= (0.025)(0.0012) \\ &= 0.00003 \end{aligned} \tag{13}$$

$$\begin{aligned} V_1(2)P(NN \mid VB) &= (0)(0.047) \\ &= 0 \end{aligned} \tag{14}$$

$$\begin{aligned} V_1(3)P(NN \mid TO) &= (0)(0.00047) \\ &= 0 \end{aligned} \tag{15}$$

$$\begin{aligned} V_1(4)P(NN \mid NN) &= (0)(0.087) \\ &= 0 \end{aligned} \tag{16}$$

$$\begin{aligned} V_2(4) &= max(0.00003, 0, 0, 0)P(want \mid NN) \\ &= (0.00003)(0.000054) \\ &= 0.00000000162 \end{aligned} \tag{17}$$

For $V_3(1)$:

$$\begin{aligned} V_2(1)P(PPSS \mid PPSS) &= (0)(0.00014) \\ &= 0 \end{aligned} \tag{18}$$

$$\begin{aligned} V_2(2)P(PPSS \mid VB) &= (0.000051)(0.007) \\ &= 0.000000357 \end{aligned} \tag{19}$$

$$\begin{aligned} V_2(3)P(PPSS \mid TO) &= (0)(0) \\ &= 0 \end{aligned} \tag{20}$$

$$\begin{aligned} V_2(4)P(PPSS \mid NN) &= (0.00000000162)(0.0045) \\ &= 0 \end{aligned} \tag{21}$$

$$\begin{aligned} V_3(1) &= max(0, 0.000000357, 0, 7.29 * 10^{-12})P(to \mid PPSS) \\ &= (0.000000357)(0) \\ &= 0 \end{aligned} \tag{22}$$

For $V_3(2)$:

$$\begin{aligned} V_2(1)P(VB \mid PPSS) &= (0)(0.23) \\ &= 0 \end{aligned} \tag{23}$$

$$\begin{aligned} V_2(2)P(VB \mid VB) &= (0.000051)(0.0038) \\ &= 1.938 * 10^{-7} \end{aligned} \tag{24}$$

$$\begin{aligned} V_2(3)P(VB \mid TO) &= (0)(0.83) \\ &= 0 \end{aligned} \tag{25}$$

$$\begin{aligned} V_2(4)P(VB \mid NN) &= (1.62 * 10^{-9})(0.0045) \\ &= 6.48 * 10^{-12} \end{aligned} \tag{26}$$

$$\begin{aligned} V_3(2) &= max(0, 1.938 * 10^{-7}, 0, 6.48 * 10^{-12})P(to \mid VB) \\ &= (1.938 * 10^{-7})(0) \\ &= 0 \end{aligned} \tag{27}$$

For $V_3(3)$:

$$\begin{aligned} V_2(1)P(TO \mid PPSS) &= (0)(0.00079) \\ &= 0 \end{aligned} \tag{28}$$

$$\begin{aligned} V_2(2)P(TO \mid VB) &= (0.000051)(0.0035) \\ &= 0.000001785 \end{aligned} \tag{29}$$

$$\begin{aligned} V_2(3)P(TO \mid TO) &= (0)(0) \\ &= 0 \end{aligned} \tag{30}$$

$$\begin{aligned} V_2(4)P(TO \mid NN) &= (1.62 * 10^{-9})(0.016) \\ &= 2.592 * 10^{-11} \end{aligned} \tag{31}$$

$$\begin{aligned} V_3(3) &= max(0, 0.000001785, 0, 2.592 * 10^{-11})P(to \mid TO) \\ &= (0.000001785)(0.99) \\ &= 0.00000176715 \end{aligned} \tag{32}$$

For $V_3(4)$:

$$\begin{aligned} V_2(1)P(TO \mid PPSS) &= (0)(0.0012) \\ &= 0 \end{aligned} \tag{33}$$

$$\begin{aligned} V_2(2)P(TO \mid VB) &= (0.000051)(0.047) \\ &= 0.000002397 \end{aligned} \tag{34}$$

$$\begin{aligned} V_2(3)P(TO \mid TO) &= (0)(0.00047) \\ &= 0 \end{aligned} \tag{35}$$

$$\begin{aligned} V_2(4)P(TO \mid NN) &= (1.62 * 10^{-9})(0.087) \\ &= 1.4094 * 10^{-10} \end{aligned} \tag{36}$$

$$\begin{aligned} V_3(4) &= max(0, 0.000002397, 0, 1.4094 * 10^{-10})P(to \mid NN) \\ &= (0.000002397)(0) \\ &= 0 \end{aligned} \tag{37}$$

For $V_4(1)$:

$$\begin{aligned} V_3(1)P(PPSS \mid PPSS) &= (0)(0.00014) \\ &= 0 \end{aligned} \tag{38}$$

$$\begin{aligned} V_3(2)P(PPSS \mid VB) &= (0)(0.007) \\ &= 0 \end{aligned} \tag{39}$$

$$\begin{aligned} V_3(3)P(PPSS \mid TO) &= (0.00000176715)(0) \\ &= 0 \end{aligned} \tag{40}$$

$$\begin{aligned} V_3(4)P(PPSS \mid NN) &= (0.00000000162)(0.0047) \\ &= 0 \end{aligned} \tag{41}$$

$$\begin{aligned} V_4(1) &= max(0,0,0,0)P(race \mid PPSS) \\ &= (0)(0) \\ &= 0 \end{aligned} \tag{42}$$

For $V_4(2)$:

$$\begin{aligned} V_3(1)P(VB \mid PPSS) &= (0)(0.023) \\ &= 0 \end{aligned} \tag{43}$$

$$\begin{aligned} V_3(2)P(VB \mid VB) &= (0)(0.0038) \\ &= 0 \end{aligned} \tag{44}$$

$$\begin{aligned} V_3(3)P(VB \mid TO) &= (0.00000176715)(0.83) \\ &= 0.0000014667345 \end{aligned} \tag{45}$$

$$\begin{aligned} V_3(4)P(VB \mid NN) &= (0)(0.0040) \\ &= 0 \end{aligned} \tag{46}$$

$$\begin{aligned} V_4(2) &= max(0,0,0.0000014667345,0)P(race \mid VB) \\ &= (0.0000014667345)(0.00012) \\ &= 1.7600814 * 10^{-10} \end{aligned} \tag{47}$$

For $V_4(3)$:

$$\begin{aligned} V_3(1)P(TO \mid PPSS) &= (0)(0.00079) \\ &= 0 \end{aligned} \tag{48}$$

$$\begin{aligned} V_3(2)P(TO \mid VB) &= (0)(0.035) \\ &= 0 \end{aligned} \tag{49}$$

$$\begin{aligned} V_3(3)P(TO \mid TO) &= (0.00000176715)(0) \\ &= 0 \end{aligned} \tag{50}$$

$$\begin{aligned} V_3(4)P(TO \mid NN) &= (0)(0.016) \\ &= 0 \end{aligned} \tag{51}$$

$$\begin{aligned} V_4(3) &= max(0,0,0,0)P(race \mid TO) \\ &= (0)(0) \\ &= 0 \end{aligned} \tag{52}$$

For $V_4(4)$:

$$\begin{aligned} V_3(1)P(NN \mid PPSS) &= (0)(0.0012) \\ &= 0 \end{aligned} \tag{53}$$

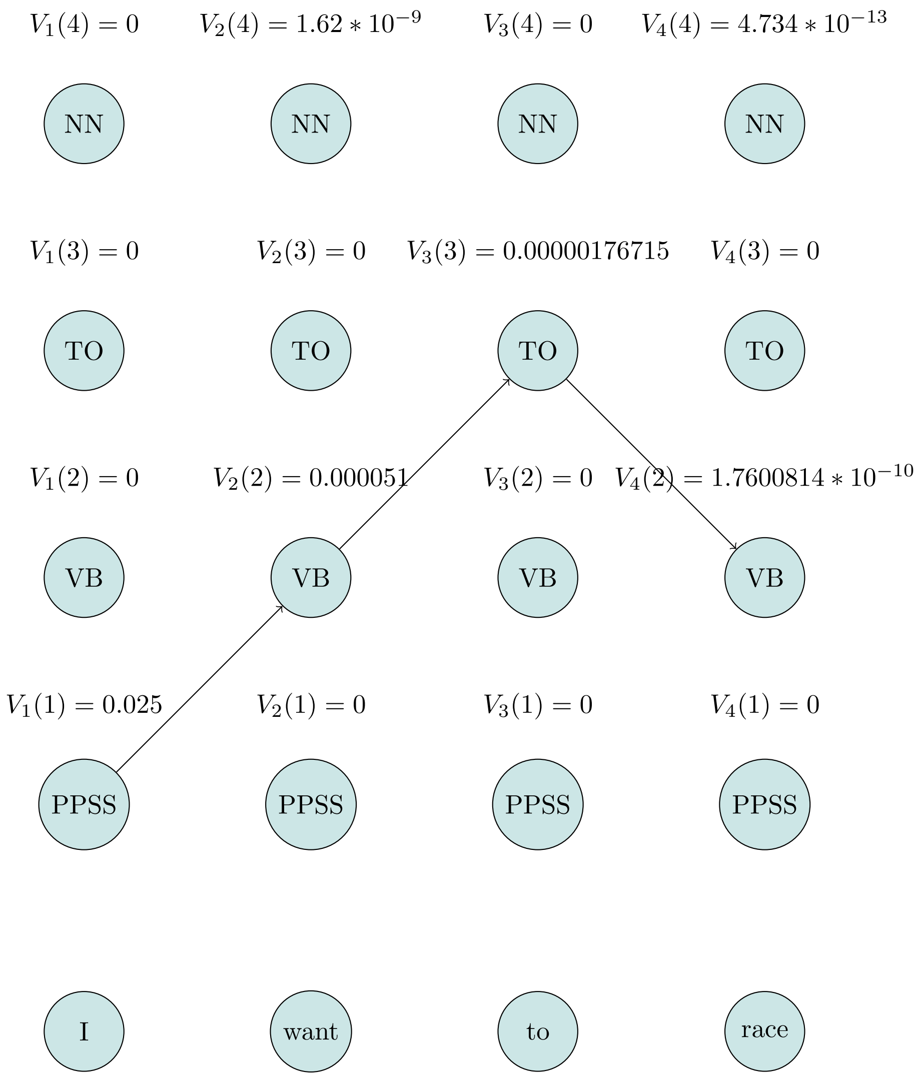

Figure 1: Showing path for question 2

$$\begin{aligned} V_3(2)P(NN \mid VB) &= (0)(0.047) \\ &= 0 \end{aligned} \tag{54}$$

$$\begin{aligned} V_3(3)P(NN \mid TO) &= (0.00000176715)(0.00047) \\ &= 8.305605 * 10^{-10} \end{aligned} \tag{55}$$

$$\begin{aligned} V_3(4)P(NN \mid NN) &= (0)(0.087) \\ &= 0 \end{aligned} \tag{56}$$

$$\begin{aligned} V_4(4) &= max(0, 0, 8.305605 * 10^{-10}, 0)P(race \mid NN) \\ &= (8.305605 * 10^{-10})(0.00057) \\ &= 4.73419 * 10^{-13} \end{aligned} \tag{57}$$

The path can be seen in Figure 1. In each step going to the node with the highest probability. Thus the pat is PPSS VB TO VB

3 Question 3

Please run POS.py and please ensure that python 3.7 is installed. It can be found at https://www.python.org/downloads/.

Output file for question 3.1b is out.txt and the confusion matrix for question 3.1c will be named confusionMatrix.txt

4 Question 4

4.1 Question 4.1

The parse tree for this question can be found in Figure 2.

The following will be added to the lexicon. The verbs: ”would”, ”like” and ”ride”. The infinitive marker ”to”. The preposition ”with”. The proper noun ”Golden Arrow”.

The following grammar rules will also be added:

VP → VP VP

VP → Infinitive marker VP

4.2 Question 4.2

The parse tree for this question can be found in Figure 3.

We add the following words to the lexicon. The pronoun ”What”, noun ”fare”, and Propen nouns ”Cape Town” and ”Bloemfontein”.

The following grammar rules will also be added:

S → NP VP ?

VP → Verb NP PP PP

4.3 Question 4.3

The parse tree for this question can be found in Figure 3.

We add the following words to the lexicon. The adverb ”there”, Nouns ”Greyhound” and ”route” and proper nouns ”Durban” and ”Bela-Bela”.

The following grammar rules will also be added:

S → VP NP PP PP ?

VP → VP Adverb

5 Question 5

5.1 Question 5.1

[Rapunzel]$_{NP}$[let]$_{VP}$down[her goldern hair]$_{NP}$

5.2 Question 5.2

How[do]$_{VP}$[I]$_{NP}$[get]$_{VP}$to[Mozambique]$_{NP}$?

5.3 Question 5.3

[Can]$_{VP}$[the manager]$_{NP}$[give]$_{VP}$me[another room]$_{NP}$?

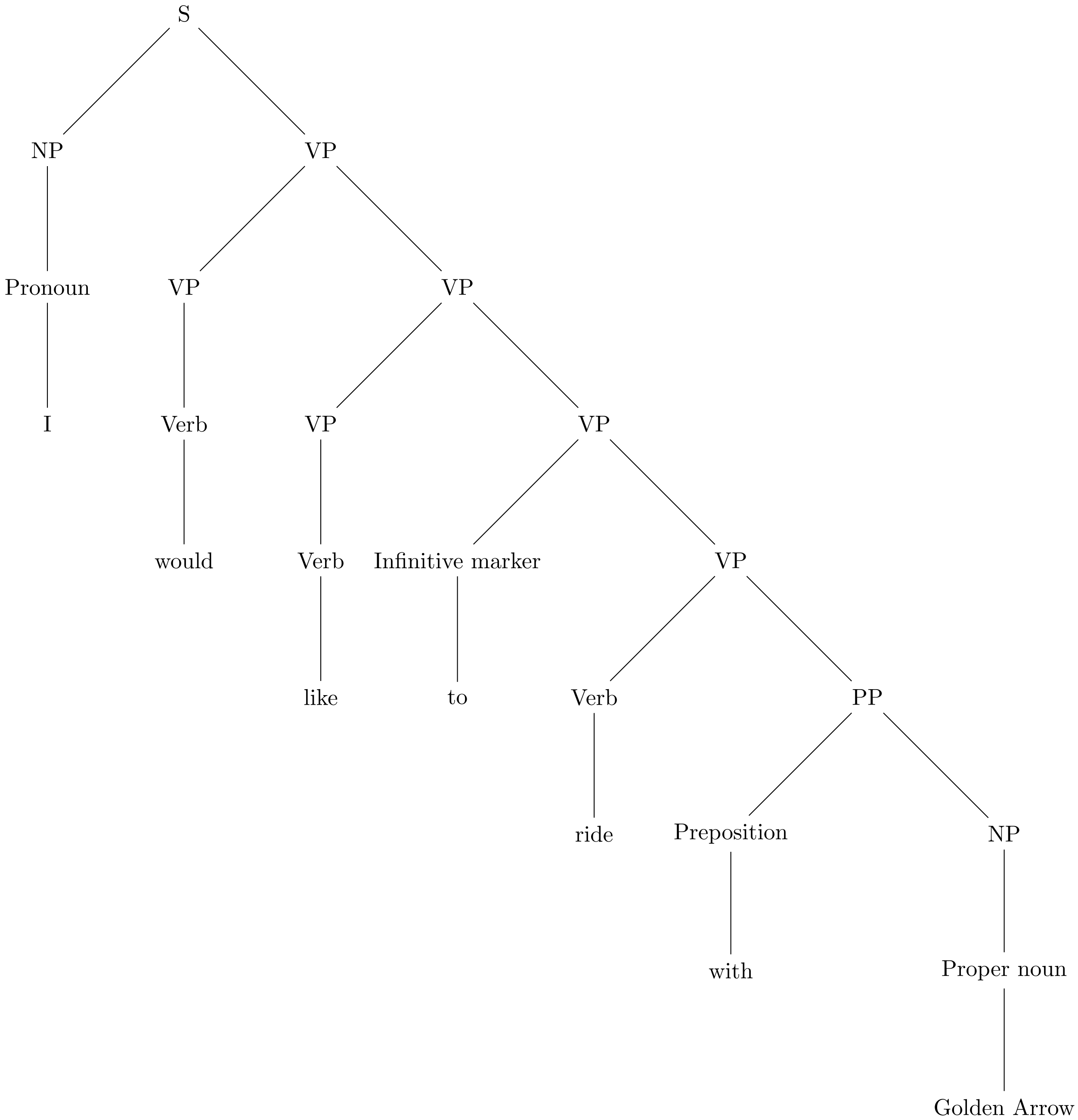

Figure 2: Parse tree for question 4.1

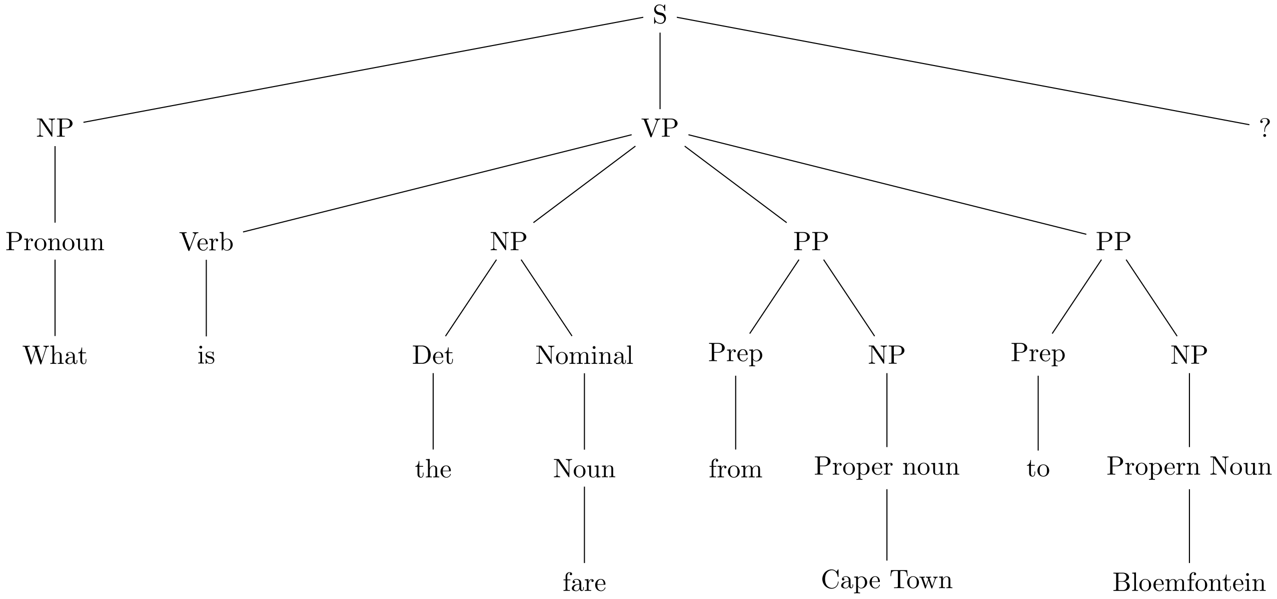

Figure 3: Parse tree for question 4.2

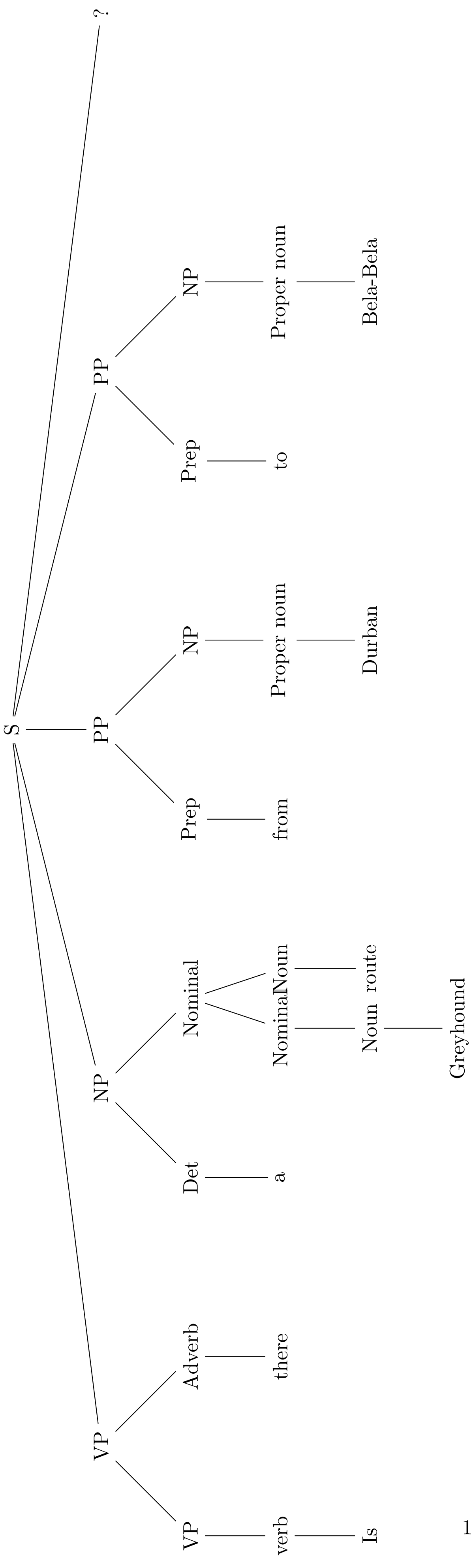

Figure 4: Tree for question 4.3