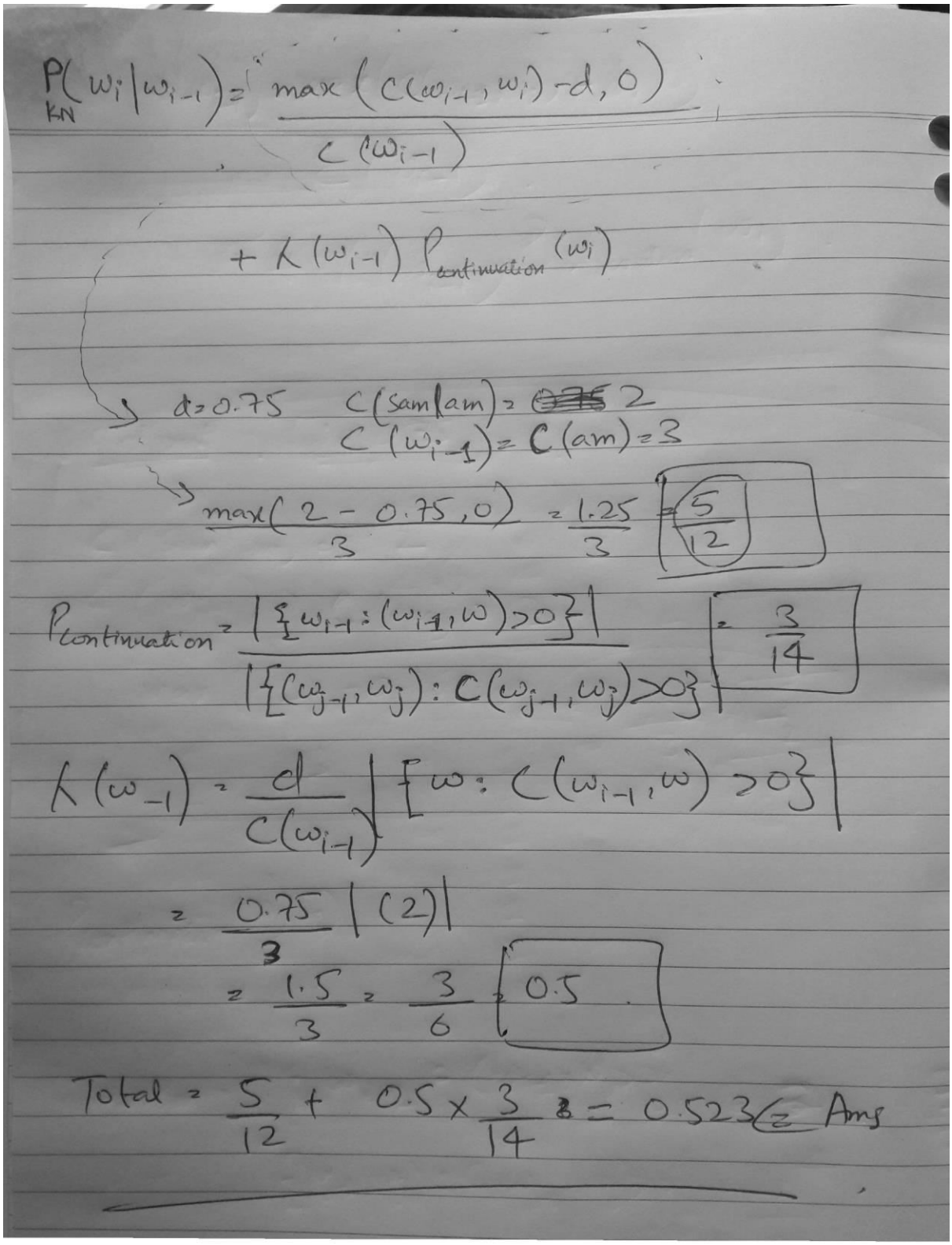
Assignment 3 NLP

Document Analysis

Q2.



Q3.

ORIGINAL GRAMMAR	**Corrected Grammar**
PRP$->my|his|her|its	Unchanged
PNP->nounEndWithS’	not required
Nominal-> PNP	PNP->Nominal
DET Nominal ->Det Noun	Nominal-> Det Noun
Nominal->PRP$ Nominal	Unchanged
Nominal-> Nominal Noun	Unchanged
Nominal->Noun	Unchanged

Q4

1. Firstly we will find the frequency of all the different words in the training corpus.

2. We will extract only the words with frequency 1 and save it in a map called RARE.

3. Then we will go through the corpus again replacing every word that exists in the RARE map to a new string called 'UNK$'. We will name it as New_Corpus;

4. We will learn word embeddings for all the distinct words in the new corpus.

5. We will train another weight matrix ahead of word embedding matrix for word prediction purposes.

6. Use gradient descent to optimize weigthts of both matrices.

7. Once the model is finished training. It is ready for testing.

5. While testing, in the test set, we replace all unseen words to the string 'UNK$'.

Q5.

Part 1

All arcs in this algorithms are made either by Right-arc or left-arc action. So both of these make sure that there are no cycles created in our dependency parsing.

Assuming the transition left arc- adds an arc from n’->n from the next input token n’ to the node n on the top of the stack. The reason that dependant node is immediately removed afterwards is to eliminate the possibility of ever having an arc from n->n’ because that would create cycle in the graph.

The reason for adding the left most element of the que in the stack after right-arc is the same as above i.e. to prevent creating cycles.

Part 2

The space complexity of Nivre's Algorithm is O(n). Because the parser is initialized at <nil,W,theta> and ends at (S,nil,A). The aggregate space needed by S(stack) and I(list) is never greater than the number of tokens(words) because [shift,right arc] (push) pop the word from the I(input list) and [left arc,reduce] (pop) pop the stack. pop transitions are bounded by the number of push transitions. So eventually the memory required can never exceed memory required to store n.