NLP_PQP_oct_2023

Parsing [3+3=6 Marks]

a) Use the grammar outlined below to parse the following sentence using the top-down parsing method with detailed steps. [3 Marks]

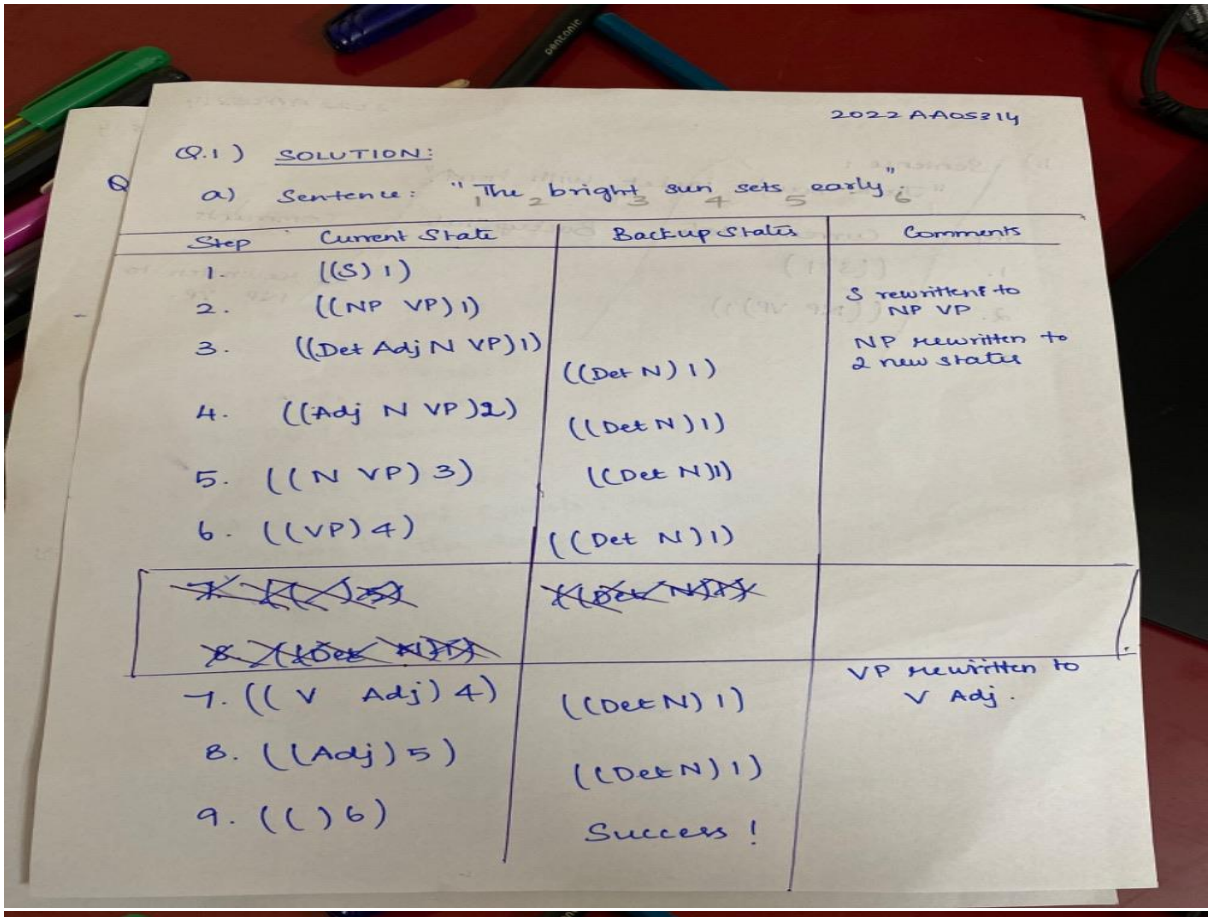
Sentence: "The bright sun sets early"

Rules:
S-> NP VP
NP -> Det Adj N
NP-> Det N
V-> V
VP -> V Adv
The -> ART
Bright -> N, ADJ
Sun-> N
Sets-> V
Early-> Adv

b) The rules for the example sentence " Jack bought jacket with hood" are,

S -> NP VP
VP -> Verb NP
VP -> VP PP
PP -> Preposition NP
NP -> "Jack"
Verb -> "bought"
NP -> "jacket"
Preposition -> "with"
NP -> "hood"
Define CFG with respect to 4-tuple: (N,Σ,R,S). [3 Marks]

NLP_PQP_oct_2023

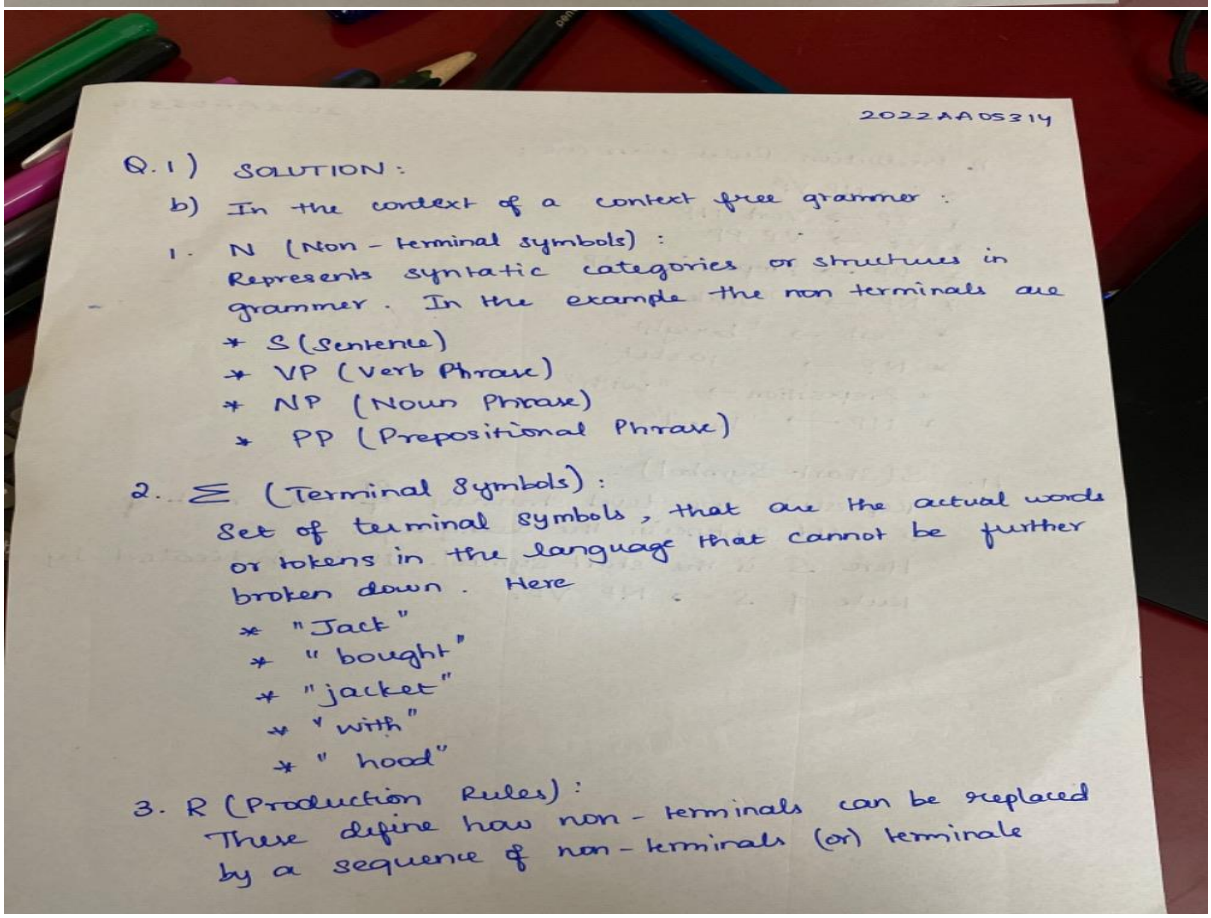


2022 AAOS314

Q.1) SOLUTION:

a) Sentence: "The bright sun sets early"
1 2 3 4 5 6

Step	Current State	Backup States	Comments
1.	((S) 1)		
2.	((NP VP) 1)		S rewritten to NP VP
3.	((Det Adj N VP) 1)	((Det N) 1)	NP rewritten to 2 new states
4.	((Adj N VP) 2)	((Det N) 1)	
5.	((N VP) 3)	((Det N) 1)	
6.	((VP) 4)	((Det N) 1)	
~~7.~~	~~((Det N VP) 4)~~	~~((Det N) 1)~~	
~~8.~~	~~((Det N) 5)~~		
7.	((V Adj) 4)	((Det N) 1)	VP rewritten to V Adj.
8.	((Adj) 5)	((Det N) 1)	
9.	(( ) 6)	Success!	



2022 AAOS314

Q.1) SOLUTION:

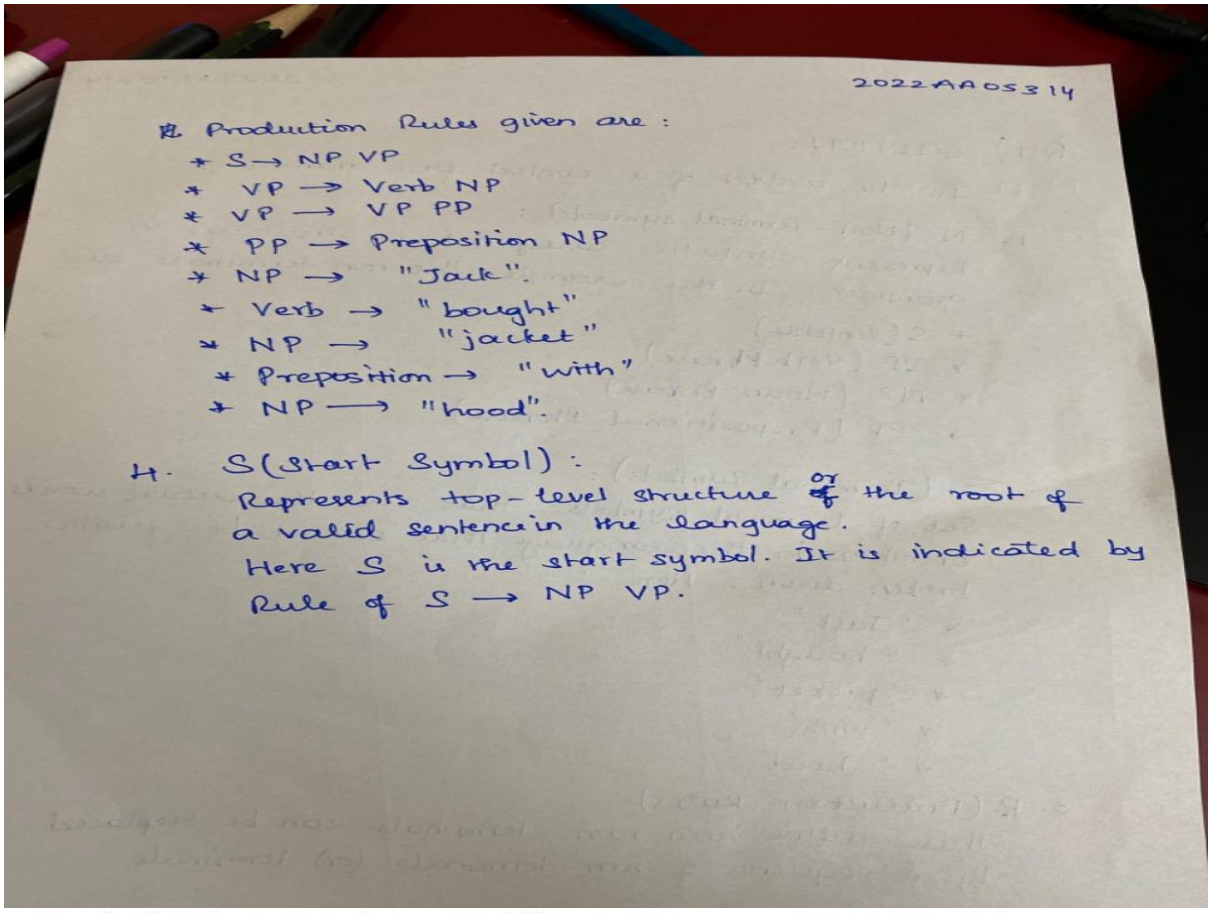
b) In the context of a context free grammer:

1. N (Non-terminal symbols):
Represents syntatic categories or structures in grammer. In the example the non terminals are
* S (Sentence)
* VP (Verb Phrase)
* NP (Noun Phrase)
* PP (Prepositional Phrase)

2. Σ (Terminal Symbols):
Set of terminal symbols, that are the actual words or tokens in the language that cannot be further broken down. Here
* "Jack"
* "bought"
* "jacket"
* "with"
* "hood"

3. R (Production Rules):
These define how non-terminals can be replaced by a sequence of non-terminals (or) terminals

NLP_PQP_oct_2023



2022AA05314

Production Rules given are :

* S → NP VP
* VP → Verb NP
* VP → VP PP
* PP → Preposition NP
* NP → "Jack".
* Verb → "bought"
* NP → "jacket"
* Preposition → "with"
* NP → "hood".

4. S (Start Symbol) :
Represents top-level structure of the root of a valid sentence in the language.
Here S is the start symbol. It is indicated by Rule of S → NP VP.

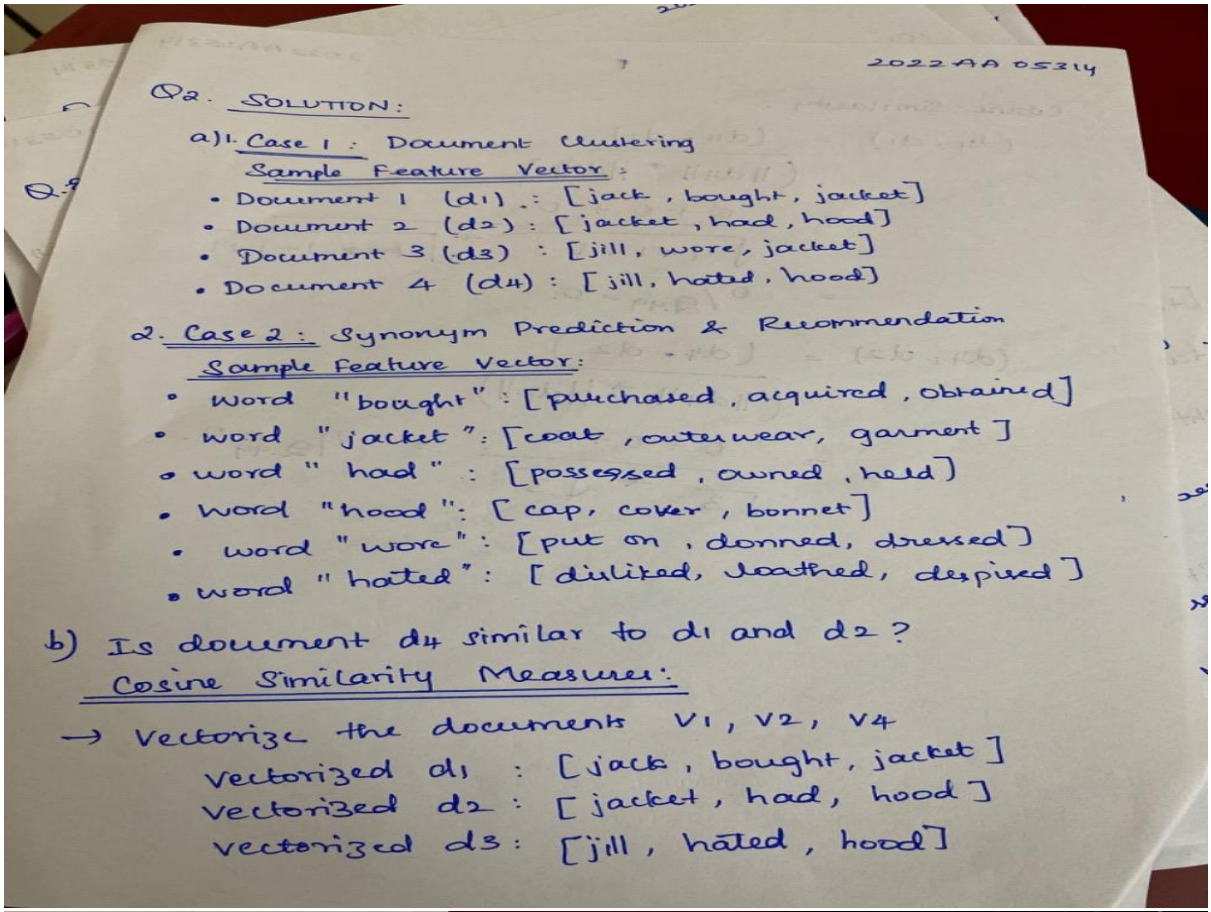
Consider the below input and answer the following questions. **[1+1+2 = 4 Marks]**

Document 1 (d1)	***Jack bought jacket***
Document 2 (d2)	***Jacket had hood***
Document 3 (d3)	***Jill wore jacket***
Document 4 (d4)	***Jill hated hood***

a) List only one sample feature vector for each of the below use cases using the above training data:
   1. Case 1: Requirement is to automate document clustering
   2. Case 2: Requirement is to model a system that can predict & recommend synonym of words

b) Is document d4 similar to d1 or d2? Justify your answer using cosine similarity measure. Note: Ignore punctuations & treat small case vs capital case as same but retain all the tokens, including stop words if any.

NLP_PQP_oct_2023



2022AA05314

Q2. SOLUTION:

a) 1. Case 1 : Document Clustering

Sample Feature Vector :

- Document 1 (d1) : [jack, bought, jacket]
- Document 2 (d2) : [jacket, had, hood]
- Document 3 (d3) : [jill, wore, jacket]
- Document 4 (d4) : [jill, hated, hood]

2. Case 2 : Synonym Prediction & Recommendation

Sample Feature Vector:

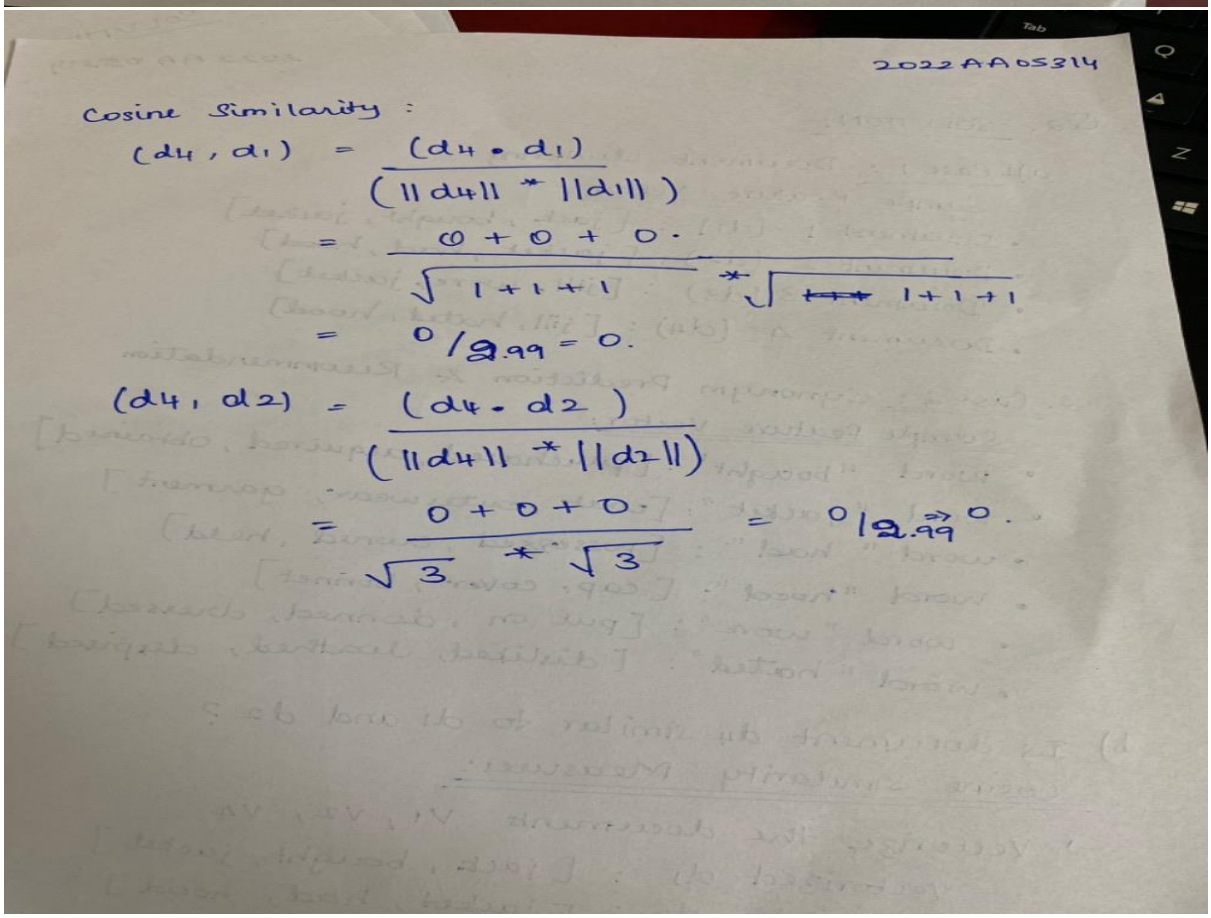
- Word "bought" : [purchased, acquired, obtained]
- word "jacket" : [coat, outerwear, garment]
- word "had" : [possessed, owned, held]
- word "hood" : [cap, cover, bonnet]
- word "wore" : [put on, donned, dressed]
- word "hated" : [disliked, loathed, despised]

b) Is document d4 similar to d1 and d2 ?

Cosine Similarity Measure:

→ Vectorize the documents V1, V2, V4

Vectorized d1 : [jack, bought, jacket]
Vectorized d2 : [jacket, had, hood]
vectorized d3 : [jill, hated, hood]



2022AA05314

Cosine Similarity :

$$(d4, d1) = \frac{(d4 \cdot d1)}{(\|d4\| * \|d1\|)}$$

$$= \frac{0 + 0 + 0}{\sqrt{1+1+1} * \sqrt{1+1+1}}$$

$$= 0/2.99 = 0.$$

$$(d4, d2) = \frac{(d4 \cdot d2)}{(\|d4\| * \|d2\|)}$$

$$= \frac{0 + 0 + 0}{\sqrt{3} * \sqrt{3}} = 0/2.99 \Rightarrow 0.$$

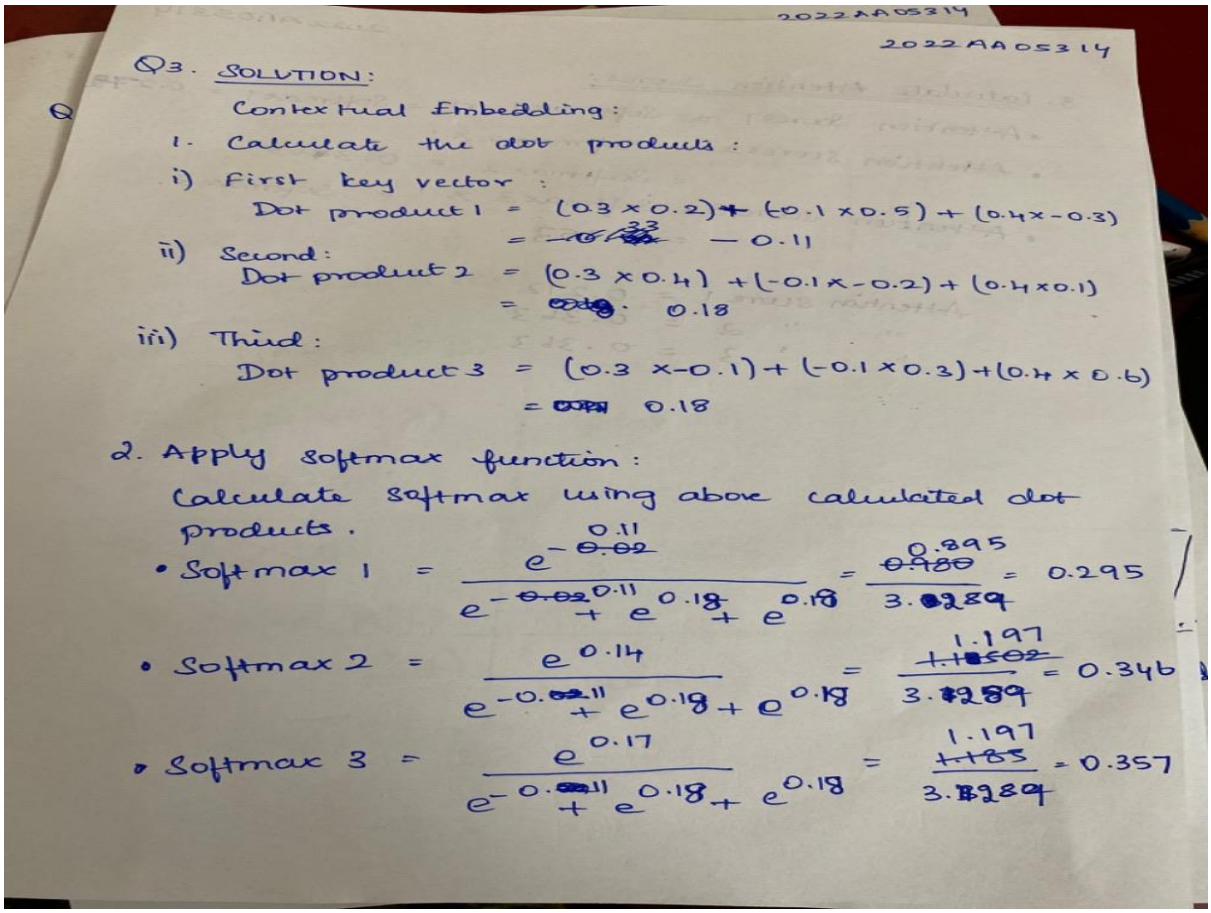
NLP_PQP_oct_2023

Contextual Embedding [5 Marks]

Consider a simplified attention mechanism in a transformer-based NLP model. Given the following vectors:

- Query Vector (Q): Q=[0.3,−0.1,0.4]Q=[0.3,−0.1,0.4]
- Key Vectors (K): K=[[0.2,0.5,−0.3],[0.4,−0.2,0.1],[−0.1,0.3,0.6]]K=[[0.2,0.5,−0.3],[0.4,−0.2,0.1],[−0.1,0.3,0.6]]
- Value Vectors (V): V=[[1,0,0],[0,1,0],[0,0,1]]V=[[1,0,0],[0,1,0],[0,0,1]]

Calculate the attention scores for the given query vector Q. Clearly show your calculations.



2022AA05314

2022AA05314

Q3. SOLUTION:

Contextual Embedding:

1. Calculate the dot products:

i) First key vector:

Dot product 1 = (0.3 × 0.2) + (0.1 × 0.5) + (0.4 × −0.3)

= −0.11

ii) Second:

Dot product 2 = (0.3 × 0.4) + (−0.1 × −0.2) + (0.4 × 0.1)

= 0.18

iii) Third:

Dot product 3 = (0.3 × −0.1) + (−0.1 × 0.3) + (0.4 × 0.6)

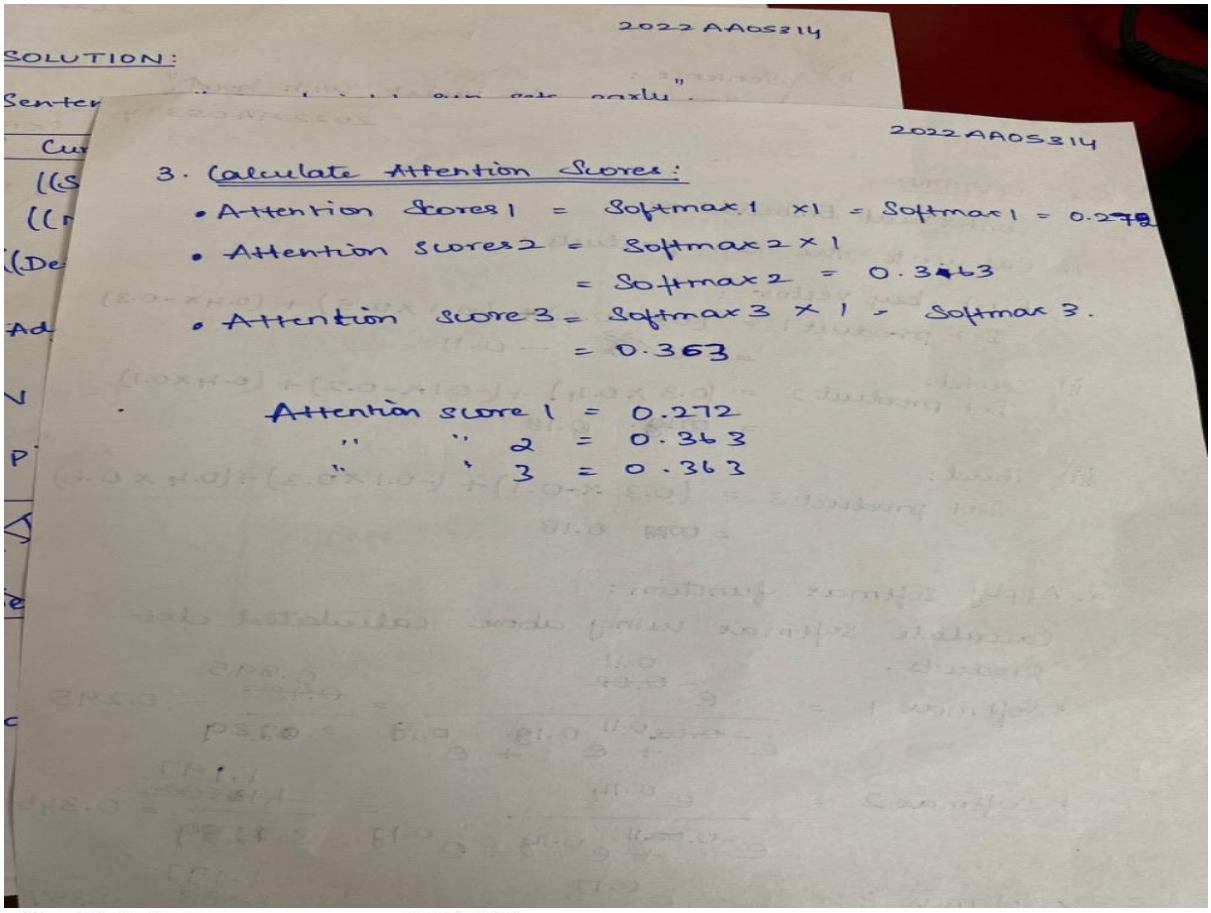
= 0.18

2. Apply softmax function:

Calculate softmax using above calculated dot products.

- Softmax 1 = $\frac{e^{0.11}}{e^{0.11} + e^{0.18} + e^{0.18}} = \frac{0.895}{3.289} = 0.295$
- Softmax 2 = $\frac{e^{0.14}}{e^{0.11} + e^{0.18} + e^{0.18}} = \frac{1.197}{3.289} = 0.346$
- Softmax 3 = $\frac{e^{0.17}}{e^{0.11} + e^{0.18} + e^{0.18}} = \frac{1.197}{3.289} = 0.357$

NLP_PQP_oct_2023



Consider below use case: **[6 Marks]**

Jack buys a jacket for Jill. Jill wears the cloth and expresses her emotion about the jacket by the either "frowning" or "smiling". Assume an Artificial Intelligence system detects Jill's expressions as observation and uses it in the Hidden Morkov Model to generate the natural language labels of the scenario. Two sample outputs look like: "Happy Jill", "Sad Jill" etc.,

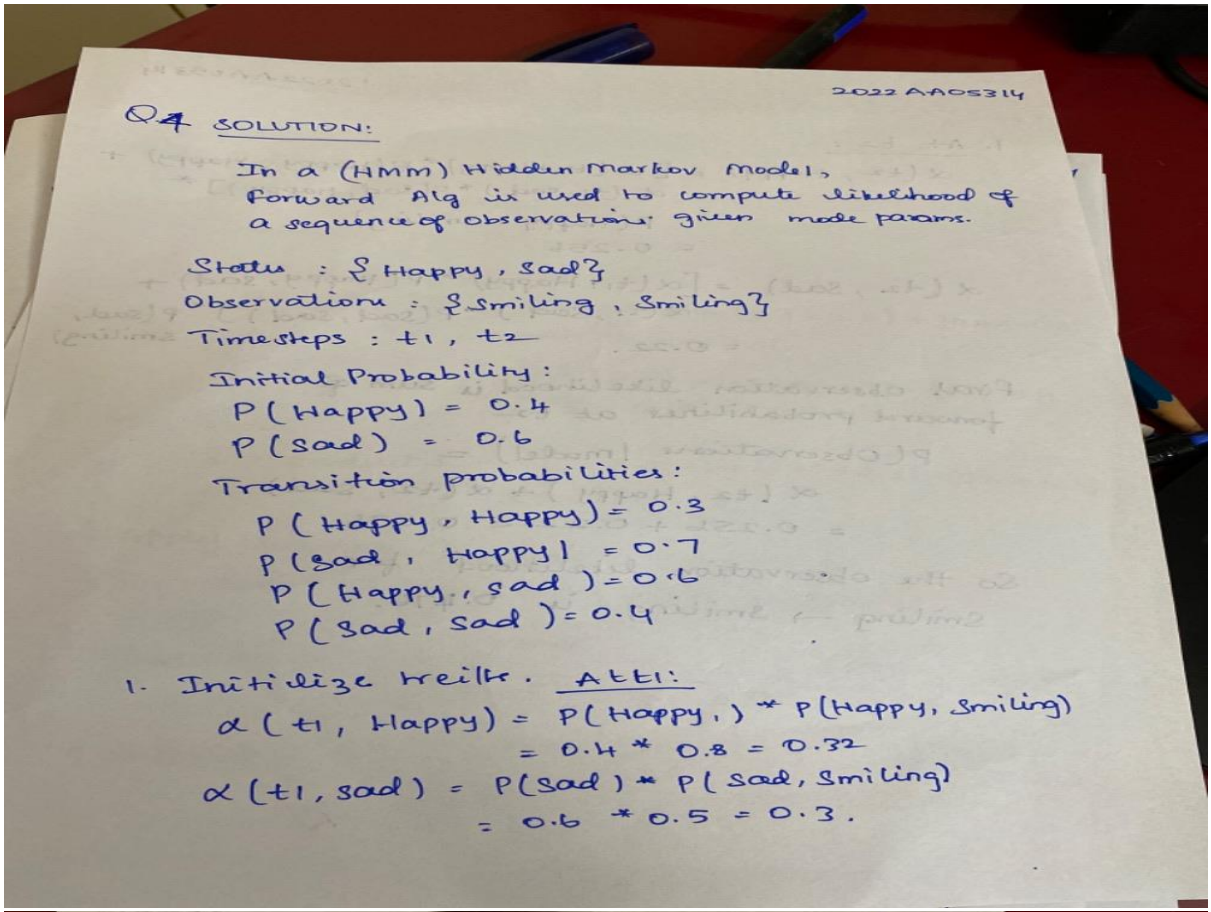
For the above scenario, use the following transition, initial probability model & observation probability models to answer the below question.
If Jill's reaction in two consecutive expression detections are ***Smiling*** → ***Smiling*** , Compute the observation likelihood using forward propagation. Show the trellis as illustrated in the prescribed text book.

Next state → Transition model	Happy	Sad
Happy	0.3	0.7
Sad	0.6	0.4

From Start state to →	Happy	Sad
Initial Probability	0.4	0.6

state →	Happy	Sad
frowning	0.2	0.5
smiling	0.8	0.5
Above depicts the Observations		

NLP_PQP_oct_2023



2022AA05314

Q4 SOLUTION:

In a (HMM) Hidden Markov Model, Forward Alg is used to compute likelihood of a sequence of observations given model params.

States : {Happy, Sad}
Observations : {Smiling, Smiling}
Timesteps : t_1, t_2

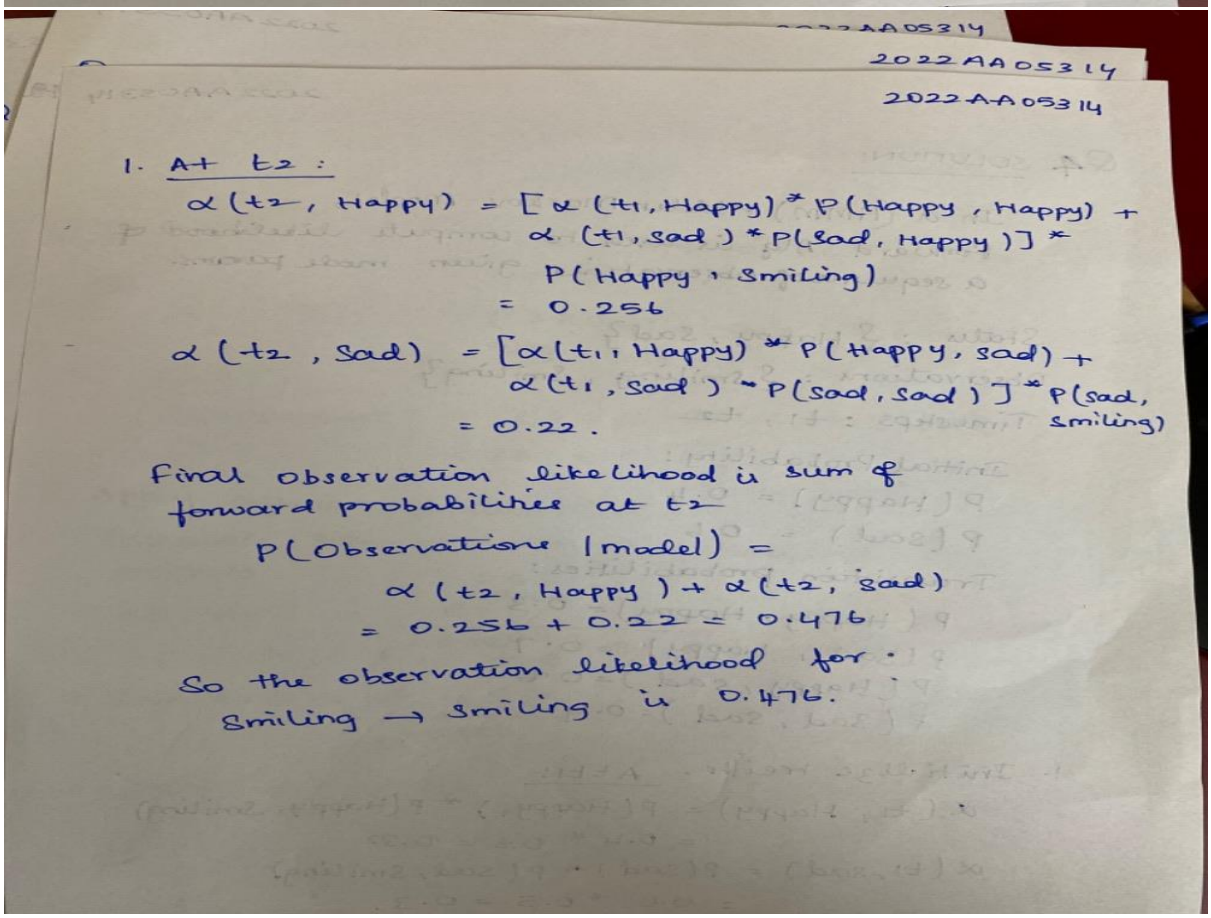
Initial Probability:
P(Happy) = 0.4
P(Sad) = 0.6

Transition probabilities:
P(Happy, Happy) = 0.3
P(Sad, Happy) = 0.7
P(Happy, Sad) = 0.6
P(Sad, Sad) = 0.4

1. Initilize trellis. At t_1:

$\alpha(t_1, Happy) = P(Happy,) * P(Happy, Smiling)$
$= 0.4 * 0.8 = 0.32$

$\alpha(t_1, Sad) = P(Sad) * P(Sad, Smiling)$
$= 0.6 * 0.5 = 0.3.$



2022AA05314

2022AA05314

2022AA05314

1. At t_2 :

$\alpha(t_2, Happy) = [\alpha(t_1, Happy) * P(Happy, Happy) + \alpha(t_1, Sad) * P(Sad, Happy)] * P(Happy, Smiling)$
$= 0.256$

$\alpha(t_2, Sad) = [\alpha(t_1, Happy) * P(Happy, Sad) + \alpha(t_1, Sad) * P(Sad, Sad)] * P(Sad, Smiling)$
$= 0.22.$

Final observation likelihood is sum of forward probabilities at t_2

$P(Observations | model) = \alpha(t_2, Happy) + \alpha(t_2, Sad)$
$= 0.256 + 0.22 = 0.476$

So the observation likelihood for Smiling → Smiling is 0.476.

NLP_PQP_oct_2023

[2.5+1+1.5 = 5 Marks]

For the generic text summarization, given below 5 documents in the collection, answer the following questions sequentially.

Document 1 (d1)	***Jack bought jacket***
Document 2 (d2)	***Jacket had hood***
Document 3 (d3)	***Jill wore jacket***
Document 4 (d4)	***Jill hated hood***

a) Compute the saliency score of every sentence using the average weights of all the words in the sentences. Use only the new simplified formula to compute word's weightage. Note: Ignore punctuations & treat small case vs capital case as same but retain all the words, including stop words if any.

Weight (word W_i) = N/n_i

Where,

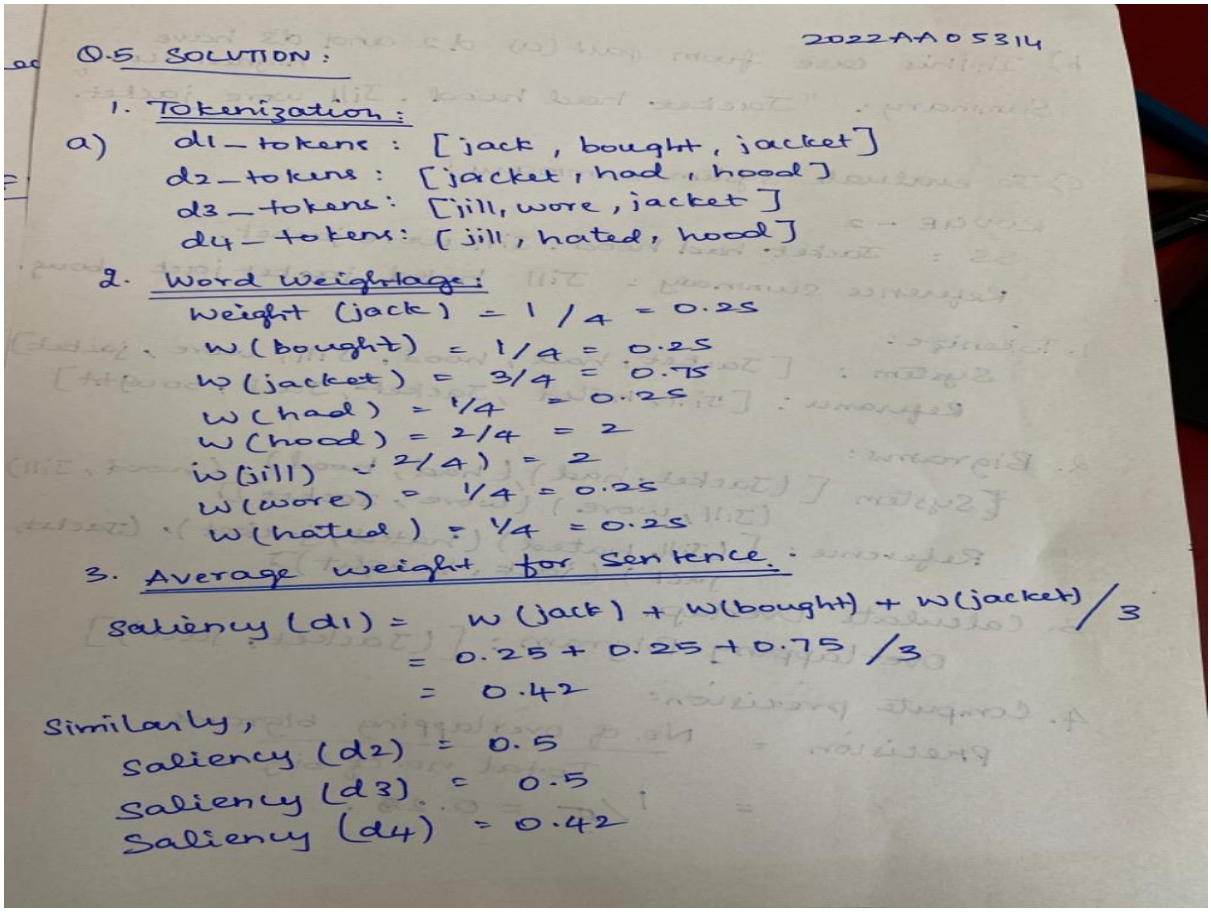
N = Number of document given in the collection

n_i = Number of documents in the collection that has the word W_i

b) Extract summary with 2 most informative documents from part a)'s result.

c) Use ROUGE-2 to evaluate the results of the above system summary obtained in part b) w.r.t the below reference summary.

"Jill hated Jacket jack bought"



2022AA05314

Q.5 SOLUTION:

1. Tokenization:

a) d1_tokens : [jack, bought, jacket]
d2_tokens : [jacket, had, hood]
d3_tokens : [jill, wore, jacket]
d4_tokens : [jill, hated, hood]

2. Word weightage:

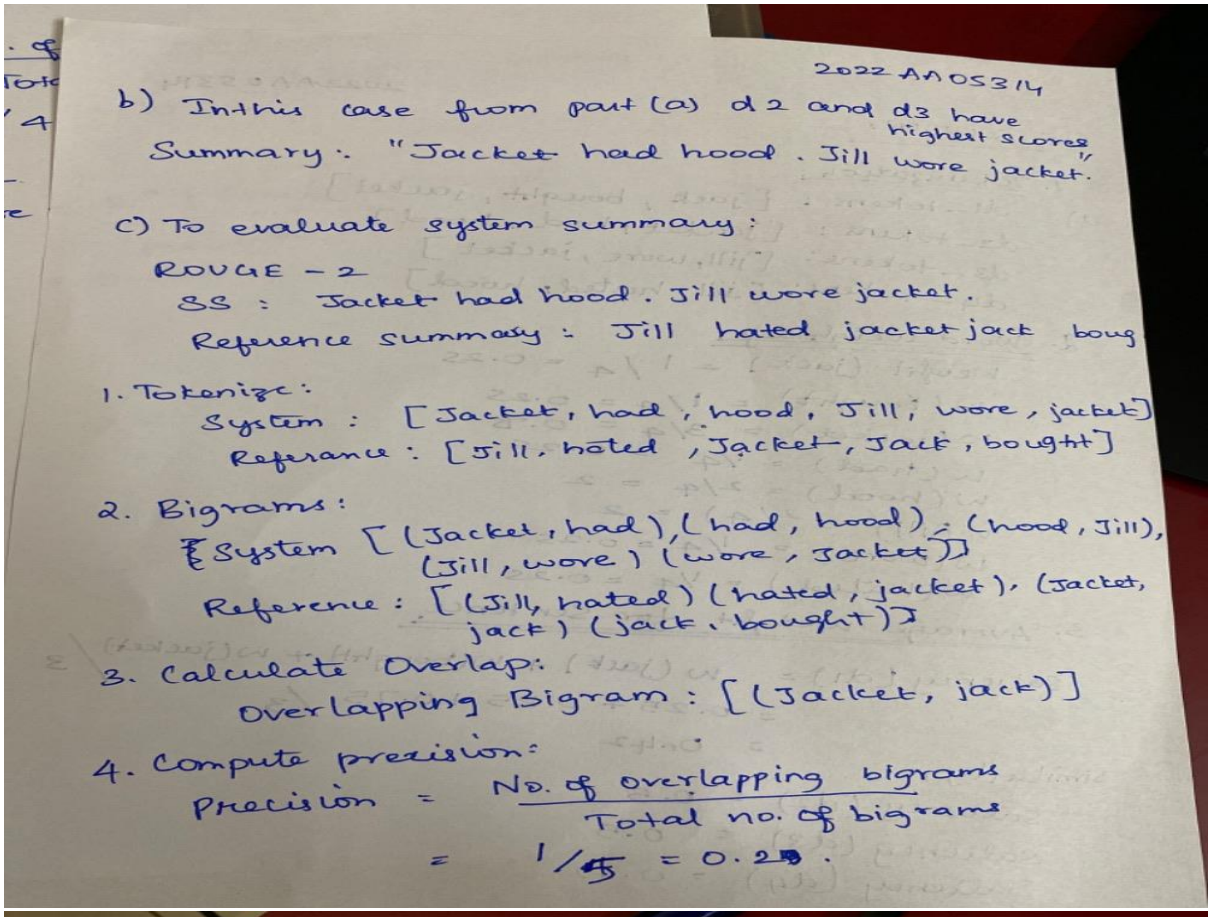
Weight (jack) = 1/4 = 0.25
W(bought) = 1/4 = 0.25
W(jacket) = 3/4 = 0.75
W(had) = 1/4 = 0.25
W(hood) = 2/4 = 2
W(jill) = 2/4 = 2
W(wore) = 1/4 = 0.25
W(hated) = 1/4 = 0.25

3. Average weight for sentence:

Saliency (d1) = W(jack) + W(bought) + W(jacket) / 3
= 0.25 + 0.25 + 0.75 / 3
= 0.42

Similarly,
Saliency (d2) = 0.5
Saliency (d3) = 0.5
Saliency (d4) = 0.42

NLP_PQP_oct_2023



2022AA05314

b) In this case from part (a) d2 and d3 have highest scores

Summary: "Jacket had hood. Jill wore jacket."

c) To evaluate system summary:

ROUGE - 2

SS : Jacket had hood. Jill wore jacket.

Reference summary : Jill hated jacket jack boug

1. Tokenize:

System : [Jacket, had, hood, Jill, wore, jacket]

Referance : [Jill, hated, Jacket, Jack, boughtt]

2. Bigrams:

System [(Jacket, had), (had, hood), (hood, Jill), (Jill, wore) (wore, Jacket)]

Reference : [(Jill, hated) (hated, jacket), (Jacket, jack) (jack, bought)]

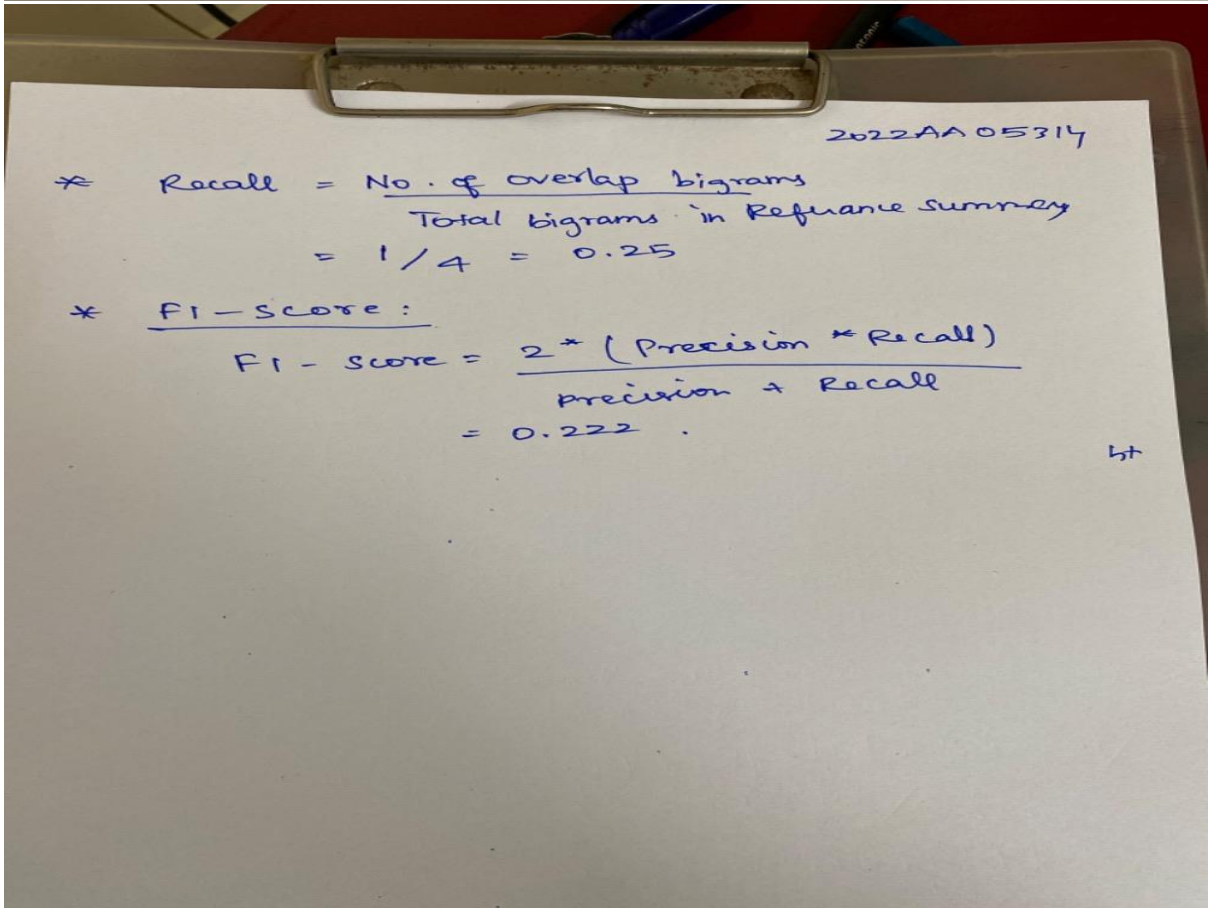
3. Calculate Overlap:

Overlapping Bigram : [(Jacket, jack)]

4. Compute precision:

$$\text{Precision} = \frac{\text{No. of overlapping bigrams}}{\text{Total no. of bigrams}}$$

$$= 1/5 = 0.2$$



2022AA05314

* Recall = $\frac{\text{No. of overlap bigrams}}{\text{Total bigrams in Refrance summary}}$

$= 1/4 = 0.25$

* F1 - score:

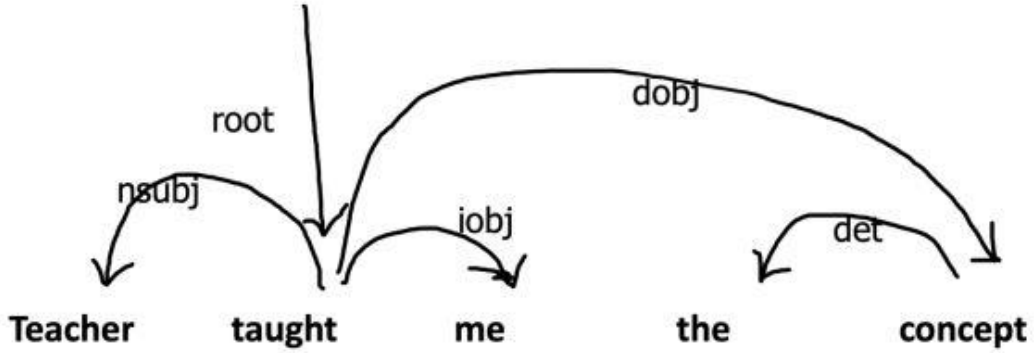
$$\text{F1 - score} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

$= 0.222$.

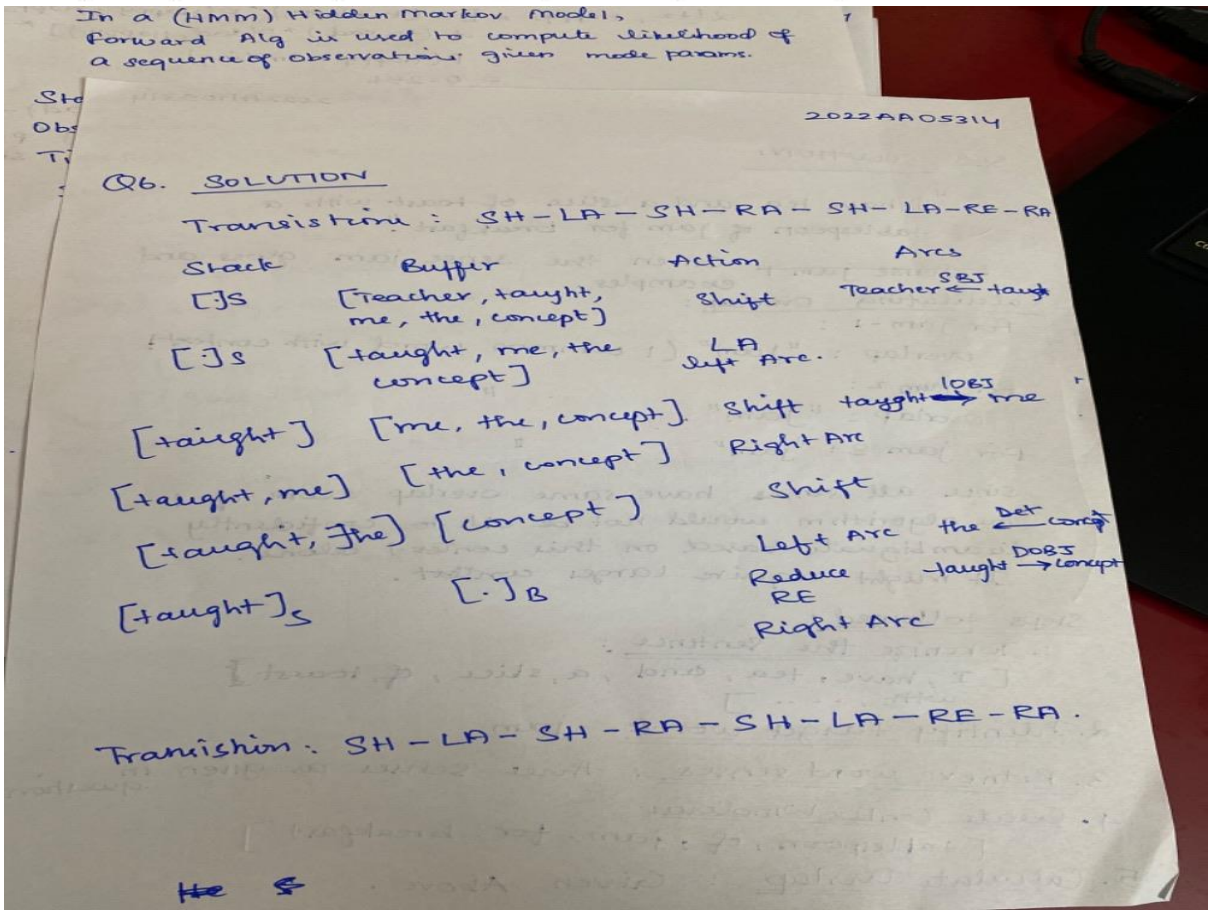
NLP_PQP_oct_2023

Dependency Parsing **[5 Marks]**

Give the correct sequence for Dependency Parsing operations for the sentence given below by filling up the table below:



Step No.	Stack	Word List (Buffer)	Action	Relation Added (Arc)
0		[Teacher, taught, me, the, concept]		



NLP_PQP_oct_2023

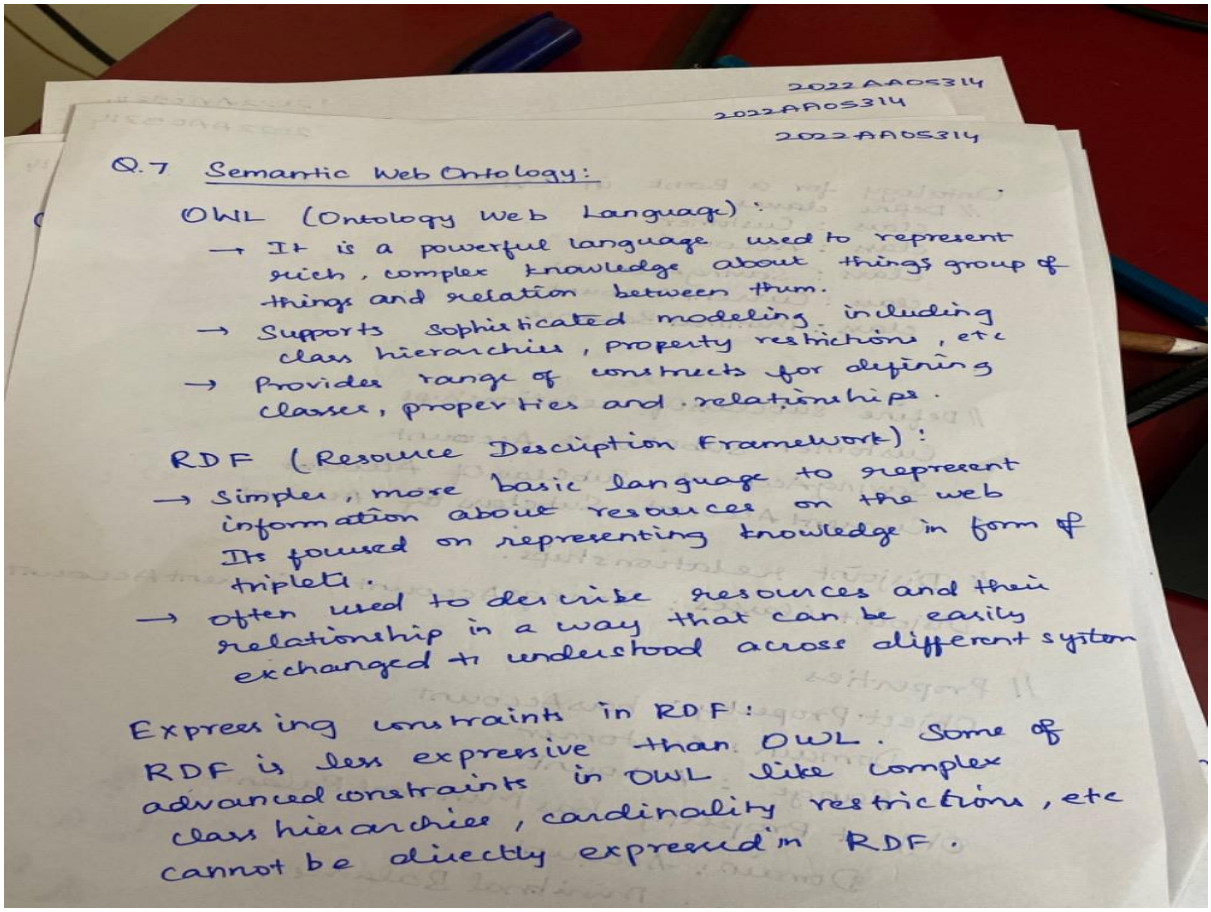
Semantic Web Ontology

a) How are the ontology languages OWL and RDF different from each other. Can you express the same constraints using RDF? If not which one cannot be expressed using RDF? **[2 marks]**
Build a part of ontology for a Bank in OWL syntax with following concepts **[2 Marks]**

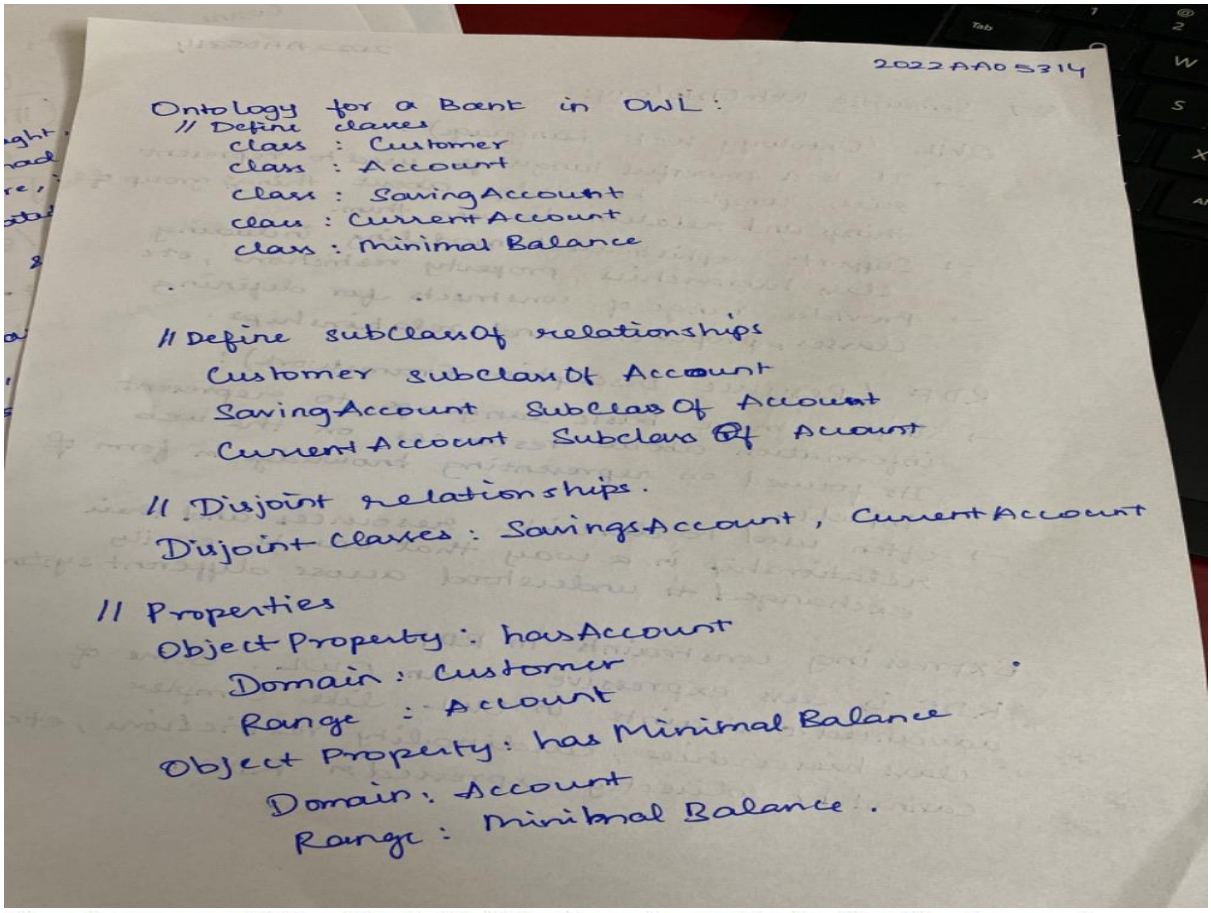
- Customer
- Account
    a. Saving account and
    b. Current account
- Minimum balance
- Bank Employee
- Bank Manager

Also include following relations/constraints:

- subClassOf
- disjointWith
- Domain
- Range



NLP_PQP_oct_2023



Show how you would disambiguate the following sentence using the Simple Lesk approach. Describe the algorithm and show how it would apply in this instance. **[5 Marks]**

I have tea and a slice of toast with a tablespoon of jam for breakfast.

Sense jam-1
Gloss: a crowded mass that impedes or blocks <a traffic jam>

Example: Trucks sat in a jam for ten hours waiting to cross the bridge.

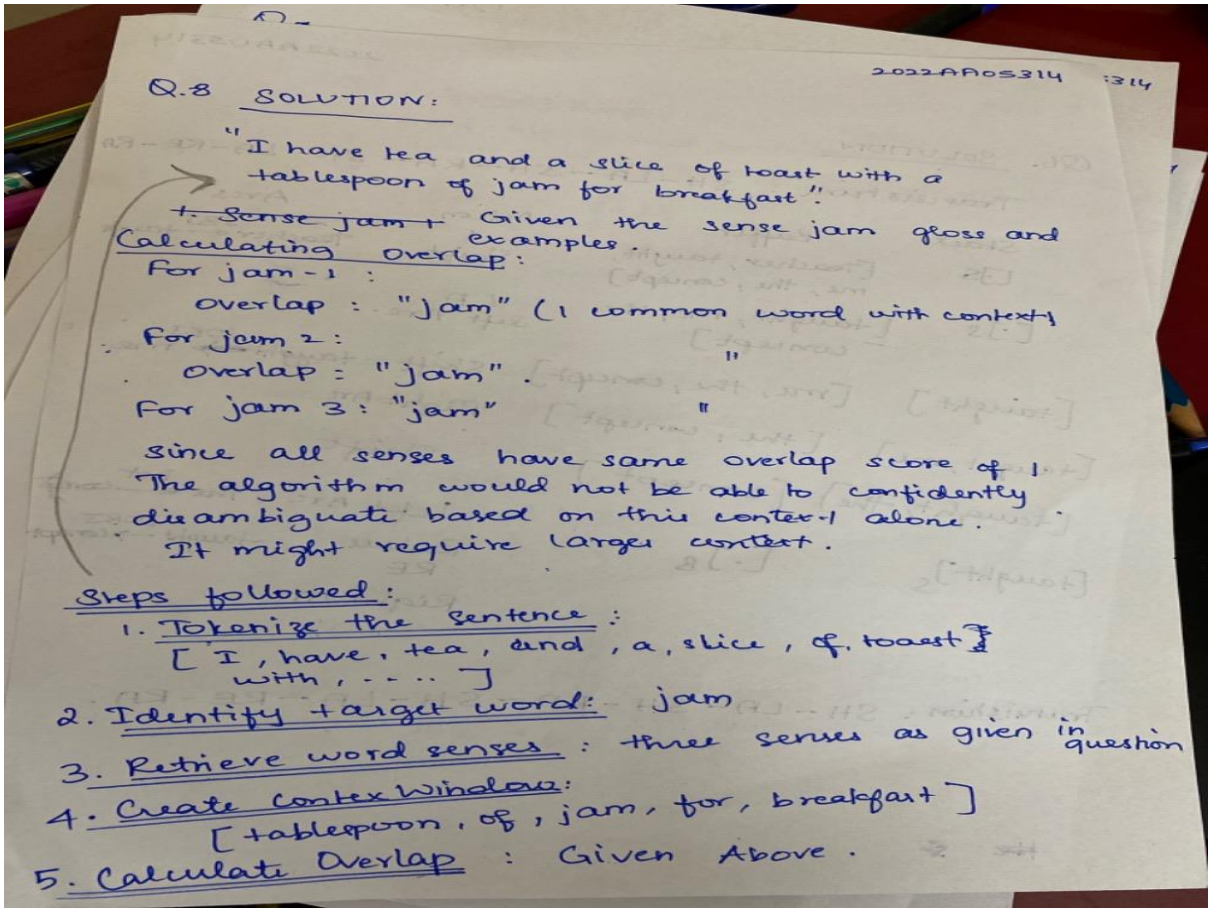
Sense jam-2
Gloss: an often impromptu performance by a group especially of jazz musicians that is characterized by improvisation

Example: The saxophone players took part in a free-form jazz jam

Sense jam-3
Gloss: a food made by boiling fruit and sugar to a thick consistency

Example: He spread home-made jam on his toast.

NLP_PQP_oct_2023



Q.8 SOLUTION:

2022AA05314 :314

"I have tea and a slice of toast with a tablespoon of jam for breakfast".

~~1. Sense jam~~ Given the sense jam gloss and examples.

Calculating Overlap:

For jam - 1 :

Overlap : "jam" (1 common word with context)

For jam 2 :

Overlap : "jam" . "

For jam 3: "jam" "

Since all senses have same overlap score of 1. The algorithm would not be able to confidently disambiguate based on this context alone. It might require larger context.

Steps followed:

1. Tokenize the sentence :
   [I, have, tea, and, a, slice, of, toast, with, ...]
2. Identify target word: jam
3. Retrieve word senses : three senses as given in question
4. Create context window:
   [tablespoon, of, jam, for, breakfast]
5. Calculate Overlap : Given Above.