POS Tagging Master Guide

HMM, Viterbi & Combinatorics

Complete Exam Preparation Set

The "Golden Formulas" Cheat Sheet

1. **HMM Point-wise Score (Disambiguation):**

$$\text{Score}(t_i) = \underbrace{P(t_i|t_{i-1})}_{\text{Trans (Row=Prev)}} \times \underbrace{P(w_i|t_i)}_{\text{Emit (Row=Tag)}}$$

Tip: In Transition Tables, usually Row = Previous Tag, Column = Current Tag.

2. **Viterbi Recursion (The "Max" Rule):**

$$v_t(j) = \max_i \left[v_{t-1}(i) \times P(t_j|t_i) \right] \times P(w_t|t_j)$$

3. **Combinatorics (Counting Sequences):**

$$\text{Total Paths} = \text{Count}(W_1) \times \text{Count}(W_2) \times \cdots \times \text{Count}(W_n)$$

4. **Log-Probability (Avoids Underflow):**

$$\text{LogScore} = \log(\text{Previous}) + \log(\text{Trans}) + \log(\text{Emit})$$

Tip: Add numbers instead of multiplying.

Contents

1 Module 1: HMM Point-wise Disambiguation

Task: Use the provided matrices (tables) to calculate scores and select the correct tag for a specific word.

Question 1.1: The "Book" Ambiguity (Standard)

Using an HMM tagger, determine the POS tag for the word **"book"** in the sentence fragment:

*"... to **book** the ..."*

Context: The previous word "to" has been tagged as **TO**. **Task:** Decide if "book" is a **VB** (Verb) or **NN** (Noun).
Table A: Transition Probabilities $P(Next|Previous)$

Prev \Curr	VB	NN	DT
TO	0.85	0.05	0.10
DT	0.05	0.70	0.00

Table B: Emission Probabilities $P(Word|Tag)$

Tag \Word	"book"	"flight"
VB	0.10	0.20
NN	0.50	0.30

Detailed Step-by-Step Solution

Objective: Compare the score for Tag VB vs Tag NN. Formula: $Score = Transition(Tag|Prev) \times Emission(Word|Tag)$.
Step 1: Calculate Score for Verb (VB)

- *Transition:* In Table A, find Row **TO** and Column **VB**. Value = **0.85**.
- *Emission:* In Table B, find Row **VB** and Column **"book"**. Value = **0.10**.
- *Calculation:* $0.85 \times 0.10 = \mathbf{0.085}$.

Step 2: Calculate Score for Noun (NN)

- *Transition:* In Table A, find Row **TO** and Column **NN**. Value = **0.05**.
- *Emission:* In Table B, find Row **NN** and Column **"book"**. Value = **0.50**.
- *Calculation:* $0.05 \times 0.50 = \mathbf{0.025}$.

Conclusion: Since $0.085 > 0.025$, the HMM tags "book" as **VB**. *Note: The high transition probability (0.85) overrides the lower emission probability.*

Question 1.2: The Zero-Probability Trap (Standard)

Disambiguate the word **"data"** given the previous tag was **JJ** (Adjective). **Candidates: NNS** (Plural Noun), **VBZ** (Verb).
Data Tables:

Trans	NNS	VBZ
JJ	0.6	0.2

Emit	"data"	"files"
NNS	0.4	0.5
VBZ	0.0	0.3

Detailed Step-by-Step Solution

Step 1: Calculate Score for NNS

- Transition ($JJ \to NNS$) = 0.6
- Emission ($NNS \to$ "data") = 0.4
- Score: $0.6 \times 0.4 = \mathbf{0.24}$

Step 2: Calculate Score for VBZ

- Transition ($JJ \to VBZ$) = 0.2
- Emission ($VBZ \to$ "data") = 0.0
- Score: $0.2 \times 0.0 = \mathbf{0.00}$

Conclusion: The tag is **NNS**. *Explanation:* Even though the transition to a verb (VBZ) is possible (0.2), the emission probability of 0.0 acts as a "veto". If the word never appears as that tag in the training data, the total score becomes zero.

Question 1.3: 3-Way Ambiguity (Tough)

Disambiguate the word **"round"** given the previous tag is **DT** (Determiner). **Candidates:** **NN** (Noun), **JJ** (Adjective), **VB** (Verb).
Transition Matrix ($P(Col|Row)$):

	NN	JJ	VB
DT	0.60	0.20	0.05

Emission Matrix ($P(Word|Tag)$):

Tag	"round"
NN	0.01
JJ	0.05
VB	0.02

Detailed Step-by-Step Solution

We must calculate scores for all three candidates to find the winner.
1. Noun (NN):

$$0.60 \text{ (Trans)} \times 0.01 \text{ (Emit)} = \mathbf{0.006}$$

2. Adjective (JJ):

$$0.20 \text{ (Trans)} \times 0.05 \text{ (Emit)} = \mathbf{0.010}$$

3. Verb (VB):

$$0.05 \text{ (Trans)} \times 0.02 \text{ (Emit)} = \mathbf{0.001}$$

Conclusion: Comparing $\{0.006, 0.010, 0.001\}$, the highest score is 0.010. The correct tag is **JJ** (Adjective).

Question 1.4: Algebraic Logic / Reverse Engineering (Tough)

An HMM is deciding between Tag A and Tag B.

- The final calculated Score for Tag B is **0.12**.
- We know the transition to Tag A is $P(A|Prev) = 0.4$.

What is the **minimum** Emission Probability $P(Word|A)$ required for the system to select Tag A instead of Tag B?

Detailed Step-by-Step Solution

Logic: For Tag A to be selected, its score must be strictly greater than Tag B's score.
Step 1: Set up the Inequality

$$Score(A) > Score(B)$$

$$P(A|Prev) \times P(Word|A) > 0.12$$

Step 2: Substitute Known Values Let $x = P(Word|A)$.

$$0.4 \times x > 0.12$$

Step 3: Solve for x

$$x > \frac{0.12}{0.4}$$

$$x > 0.3$$

Result: The emission probability $P(Word|A)$ must be strictly greater than **0.3**.

2 Module 2: Combinatorics (Counting Sequences)

Task: Calculate the number of theoretically possible tag paths based on a lexicon. These questions require logical counting, not probabilities.

Question 2.1: Basic Counting (Standard)

How many distinct POS tagging sequences are possible for the sentence:

"Time flies like an arrow"

Lexicon (Dictionary):

Word	Possible Tags	Count
Time	NN, VB	2
flies	NNS, VBZ	2
like	VB, IN, JJ, NN	4
an	DT	1
arrow	NN	1

Detailed Step-by-Step Solution

Logic: The choice of a tag for one word does not restrict the choice for another word (in a basic combinatorics context). Therefore, we multiply the number of options for each word position.

$$\text{Total} = \text{Count}(\text{Time}) \times \text{Count}(\text{flies}) \times \text{Count}(\text{like}) \times \text{Count}(\text{an}) \times \text{Count}(\text{arrow})$$

$$\text{Total} = 2 \times 2 \times 4 \times 1 \times 1$$

$$\text{Total} = \mathbf{16} \text{ sequences}$$

Question 2.2: Conditional Counting (Standard)

Calculate the valid tag sequences for the sentence: **"I saw her"**. **Lexicon:** I (1 tag), saw (2 tags: VBD, NN), her (2 tags: PRP, PRP$).
Constraint Logic:

- If "saw" is tagged as **NN** (Noun), "her" **cannot** be tagged as **PRP** (it must be PRP$).
- If "saw" is tagged as **VBD** (Verb), there are no restrictions on "her".

Detailed Step-by-Step Solution

Since the options for "her" depend on "saw", we split the problem into cases based on the ambiguous word "saw".
Case A: "saw" is VBD

- Word 1 (I): 1 option
- Word 2 (saw=VBD): 1 option

- Word 3 (her): 2 options (PRP or PRP$)
- Count: $1 \times 1 \times 2 = \mathbf{2}$ paths.

Case B: "saw" is NN

- Word 1 (I): 1 option
- Word 2 (saw=NN): 1 option
- Word 3 (her): 1 option (Must be PRP$; PRP is forbidden)
- Count: $1 \times 1 \times 1 = \mathbf{1}$ path.

Total Valid Sequences: $2 + 1 = \mathbf{3}$.

Question 2.3: Grammar Constraints (Tough)

Sentence: **"The man walks"**.

- **The**: DT (1 tag)
- **man**: NN, VB (2 tags)
- **walks**: NNS, VBZ (2 tags)

Grammar Rule: A Determiner (DT) **cannot** be immediately followed by a Verb (VB). Any sequence containing $DT \to VB$ is invalid. How many valid sequences remain?

Detailed Step-by-Step Solution

Step 1: Calculate Total Theoretical Sequences

$$1(\text{The}) \times 2(\text{man}) \times 2(\text{walks}) = 4 \text{ total paths}$$

Step 2: List and Check Paths

1. $DT \to NN \to NNS$ (Valid: Det followed by Noun)
2. $DT \to NN \to VBZ$ (Valid: Det followed by Noun)
3. $DT \to VB \to NNS$ (**Invalid**: Det followed by Verb)
4. $DT \to VB \to VBZ$ (**Invalid**: Det followed by Verb)

Result: There are $4 - 2 = \mathbf{2}$ valid sequences.

Question 2.4: Ambiguity Buckets (Tough)

A sentence has 3 words: W_1, W_2, W_3.

- W_1 has 2 possible tags: $\{A, B\}$.
- W_2 has 2 possible tags: $\{C, D\}$.

- W_3 has 1 possible tag: $\{E\}$.

Logic Rule: 1. If W_1 is tagged A, then W_2 **must** be tagged C. 2. If W_1 is tagged B, W_2 can be either C or D. How many valid sequences exist?

Detailed Step-by-Step Solution

Case 1: Start with Tag A

- $W_1 = A$ (1 option)
- $W_2 = C$ (Forced to 1 option)
- $W_3 = E$ (1 option)
- Path count: $1 \times 1 \times 1 = 1$

Case 2: Start with Tag B

- $W_1 = B$ (1 option)
- $W_2 = C$ or D (2 options)
- $W_3 = E$ (1 option)
- Path count: $1 \times 2 \times 1 = 2$

Total: $1 + 2 = \mathbf{3}$ valid sequences.

3 Module 3: The Viterbi Algorithm (5 Marks)

Task: Calculate the most likely path through the trellis table. You must initialize ($t = 1$) and recurse ($t = 2$).

Question 3.1: Full Table Calculation (Standard)

Fill the Viterbi table for the sentence **"They run"**.

- **Tags:** N (Noun), V (Verb).
- **Start Probabilities:** $P(N|S) = 0.6$, $P(V|S) = 0.2$.

Table A: Transitions ($P(Col|Row)$)

	N	V
N	0.3	0.7
V	0.5	0.5

Table B: Emissions ($P(Word|Tag)$)

	"They"	"run"
N	0.5	0.1
V	0.0	0.5

Detailed Step-by-Step Solution

Step 1: Initialization (Word 1 = "They") Formula: $V_1(tag) = P(tag|Start) \times P(\text{"They"}|tag)$

- $V_1(N) = 0.6 \times 0.5 = \mathbf{0.30}$
- $V_1(V) = 0.2 \times 0.0 = \mathbf{0.00}$

Step 2: Recursion (Word 2 = "run") Formula: $V_2(curr) = \max[V_1(prev) \times Trans] \times Emit$
Calculate Score for Tag N:

- Path from Prev N: $0.30 \times 0.3(\text{N} \to \text{N}) = 0.09$
- Path from Prev V: $0.00 \times 0.5(\text{V} \to \text{N}) = 0.00$
- **Max Path:** 0.09 (Coming from N)
- **Final Score:** $0.09 \times 0.1(\text{Emit N} \to \text{run}) = \mathbf{0.009}$

Calculate Score for Tag V:

- Path from Prev N: $0.30 \times 0.7(\text{N} \to \text{V}) = 0.21$
- Path from Prev V: $0.00 \times 0.5(\text{V} \to \text{V}) = 0.00$
- **Max Path:** 0.21 (Coming from N)
- **Final Score:** $0.21 \times 0.5(\text{Emit V} \to \text{run}) = \mathbf{0.105}$

Conclusion: Comparing final scores (0.009 vs 0.105), the best tag for "run" is **V**.

Question 3.2: Backtracking Logic (Standard)

You have computed the Viterbi table for a 3-word sentence. The stored backpointers are:

- At $t = 3$ (Tag V): Best Previous = Tag N

- At $t = 2$ (Tag N): Best Previous = Tag D
- At $t = 1$ (Tag D): Best Previous = Start

If Tag V has the highest score at the final step ($t = 3$), what is the full tag sequence?

Detailed Step-by-Step Solution

To find the sequence, we follow the backpointers in reverse order (from end to start): 1. End at $t = 3$: **Tag V**. 2. Backpointer says previous was **Tag N**. 3. Backpointer from N says previous was **Tag D**.
Sequence: D $\rightarrow$ N $\rightarrow$ V.

Question 3.3: Reverse Engineering Viterbi (Tough)

You are given the final Viterbi score $V_2(N) = 0.048$ for the second word. We know:

- The score at the previous step was $V_1(N) = 0.4$.
- The emission probability for the second word is $P(\text{Word}_2|N) = 0.2$.
- The best path to the current state came from N at $t = 1$.

Calculate the transition probability $P(N|N)$.

Detailed Step-by-Step Solution

Step 1: Set up the Equation We know that:

$$V_2(N) = V_1(N) \times P(N|N) \times P(\text{Word}_2|N)$$

Step 2: Plug in Known Values

$$0.048 = 0.4 \times P(N|N) \times 0.2$$

Step 3: Solve for P(N—N)

$$0.048 = 0.08 \times P(N|N)$$

$$P(N|N) = \frac{0.048}{0.08}$$

$$P(N|N) = \mathbf{0.6}$$

Question 3.4: Log-Probability Viterbi (Tough)

In real-world HMMs, we use Log-Probabilities to prevent underflow. Calculate the best path score using **Log-10 Addition**. **Data:**

- $\log V_1(A) = -2.0$
- $\log P(B|A) = -0.5$ (Transition Log-Prob)
- $\log P(\text{word}|B) = -1.5$ (Emission Log-Prob)

Detailed Step-by-Step Solution

Concept: In standard probability, we multiply: $P = A \times B \times C$. In Log space, multiplication becomes addition: $\log(P) = \log(A) + \log(B) + \log(C)$.

Calculation:

$$\text{LogScore} = \log(V_1) + \log(\text{Trans}) + \log(\text{Emit})$$

$$\text{LogScore} = (-2.0) + (-0.5) + (-1.5)$$

$$\text{LogScore} = -\mathbf{4.0}$$