CS626 - Speech, Natural Language Processing, and the Web

POS Tagging UsingHMM

Group Id-
Shruti Patel, 24M0825 , CSE
Kaustubh Shivshankar Shejole, 24M2109, CSE
Tushar Katakiya, 24M2110, CSE
Kush Mangukiya, 24M0769, CSE

Date: 07/09/2024

Problem Statement

- **Objective:** Given a sequence of words, produce the POS tag sequence using HMM-Viterbi
- **Input:** The quick brown fox jumps over the lazy dog
- **Output:** The$_{DET}$ quick$_{ADJ}$ brown$_{ADJ}$ fox$_{NOUN}$ jumps$_{VERB}$ over$_{ADP}$ the$_{DET}$ lazy$_{ADJ}$ dog$_{NOUN}$
- **Dataset:** Brown corpus
- Use Universal Tag Set (12 in number)
  –[ADJ,ADP,ADV,CONJ,DET,NOUN,NUM,PRON,PRT,VERB, . ,X]
- k-fold cross validation (k=5)

Data Processing Info (Pre-processing)

- **Lower Casing:** Converting all text to lowercase standardizes the input, ensuring that the model treats *"Laplace"* and *"laplace"* as the same word, which helps in reducing the variability in text data and improving consistency in token representation.

- **Laplace Smoothing:** Laplace smoothing is a technique used to handle missing data in probabilistic models by adding a small constant (here 1) to the frequency counts. This ensures that even unseen words or events receive a non-zero probability.

- **Tokenization:** After the user inputs data through the UI, we tokenize the input and then provide the tokenized data to the model. e.g. *India,* as *'india'* and *',*'.

Overall performance

- Precision :94.42 %
- Recall :94.33 %
- F-score (3 values)
  - F_1-score : 94.31 %
  - $F_{0.5}$-score:94.34 %
  - F_2-score:94.31 %

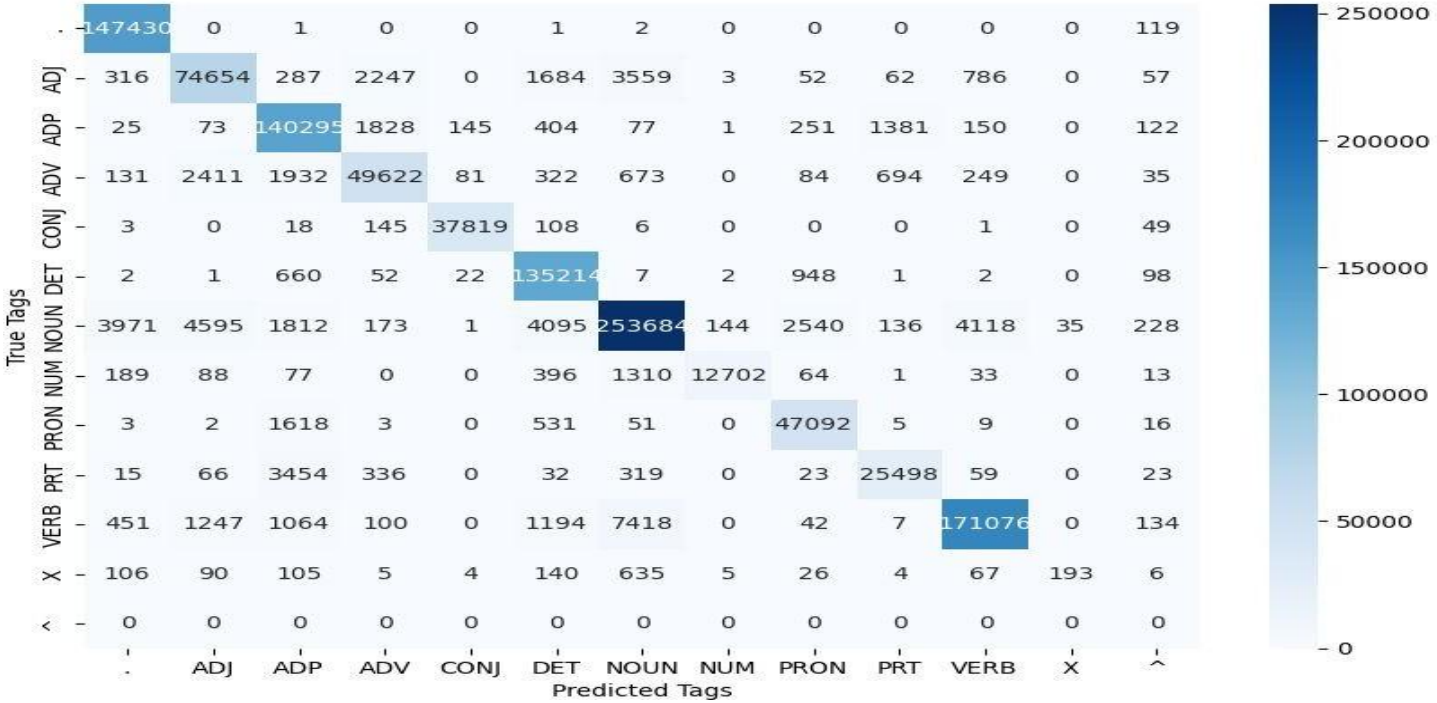
Per POS performance (1/2)

- Tag DET : P:93.82 % , R:98.69 %, F1:96.19 %
- Tag NOUN : P:94.75 %, R:92.07 % F1:93.39 %
- Tag ADJ : P:89.7 %, R:89.18 % F1:89.44 %
- Tag VERB : P:96.9 %, R:93.62 %, F1:95.23 %
- Tag ADP : P:92.71 %, R:96.92 %, F1:94.77%
- Tag . : P:96.59 %, R:99.92 %, F1:98.22 %
- Tag ADV F1:89.61 %
- Tag CONJ : P:91.03 %, R:88.24 %, F1:99.24 %
- Tag PRT : P:99.34 %, R:99.13 %, F1:88.51 %
  : P:91.76 %, R:85.49 %,

Per POS performance (2/2)

- Tag PRON : P:92.12 %, R:95.46 %, F1:93.76 %
- Tag NUM : P:98.79 %, R:85.40 %, F1:91.61 %
- Tag X : P:84.65 %, R:13.92 %, F1:23.92 %

Confusion Matrix (13 X 13)



Interpretation of confusion (error analysis)

- **ADJ vs NOUN :** This happens because some adjectives can be mistaken for nouns depending on their placement in a sentence (e.g., *"orange","light"*).

- **ADV vs. ADJ**: There is noticeable confusion between ADV (adverbs) and ADJ (adjectives). Some words can serve as both adverbs and adjectives depending on context (e.g., *"hard"* in *"The test was hard "*(ADJ) vs. *"They worked hard."*(ADV)).

- **NOUN vs. VERB:** Many words can serve as both nouns and verbs (e.g., *"run," "play"*), making them difficult to distinguish without context. Most of the noun in english also appear as verbs.

Inferencing/Decoding Info

- **Matrix Initialization:** Started by initializing a matrix to store the highest probability paths for each state at each time step.

- **Iterative Calculation:** Iteratively filled this matrix by calculating the probability of transitioning to each state, combining it with the observation likelihood and the previous state's maximum probability.

- **Backtracking:** Finally, traced back from the last time step in the matrix to determine the most likely sequence of states.

Benchmarking against ChatGPT (1/3)

- PRT predicted as ADP: 44 times (highest confusion)
- Compare the performance of HMM for each POS against ChatGPT
  - Used first 100 sentences of treebank dataset as ground truth.
  - Overall accuracy:
    - HMM: 83.11%
    - ChatGPT: 81.78%

Benchmarking against ChatGPT (2/3)

PoS Tag	ChatGPT	HMM
.	80.3	98.8
ADJ	86.6	80.9
ADP	81.5	97
ADV	77.8	77.8
CONJ	82.7	100
DET	84.3	94.9
NOUN	86.2	87.9
NUM	85.9	69.6
PRON	91.1	71.1
PRT	32	32
VERB	80.4	91.6
X	76.3	0

Benchmarking against ChatGPT (3/3)

- Possible reasons:
  - Our HMM was well trained statistically so we think that it performs better in the cases where transition and lexical probabilities play a role in determining the pos of the next tag.
  - '.' tag was well defined in HMM than ChatGPT.
  - ChatGPT had a more better sense of the pos that requires proper knowledge like it classifies pronoun and adjective in a better way than HMM.

Challenges faced

- **Handling Unseen Words:** Figuring out how to manage unseen words in the HMM.
- **Coding the Viterbi Algorithm:** Implementing the Viterbi algorithm was difficult, especially in preparing and maintaining the dynamic programming table.
- **Backtracking:** Performing backtracking to correctly reconstruct the most likely sequence of states required careful attention and was challenging to implement effectively.

Learning

- **Missing Data Handling:** Learned to apply techniques like Laplace smoothing to handle missing data in probabilistic models.
- **Evaluation Metrics:** Gained understanding of key performance metrics such as recall, precision, and F1 score.
- **Dynamic Programming:** Discovered how the Viterbi algorithm uses dynamic programming to simplify the process of finding the most likely sequence of states reduces time complexity efficiently.
- This understanding of data handling, evaluation, and dynamic programming can be applied to other NLP tasks like NER, preparing for advanced techniques in sequence modeling.

References

1. For Brown corpus
   http://www.nltk.org/nltk_data
2. For GUI
   a. https://www.gradio.app/
   b. https://streamlit.io/
   c. Any JS or python framework
3. Other references

Pushpak Bhattacharyya and Aditya Madhav Joshi, Natural Language Processing, Print ISBN: 978-93-5746-283-9 eISBN: 978-93-5746-239-6,
Wiley India, 2023.
CS626: Speech, Natural Language Processing and the Web Course

Marking Scheme (50)

1. Demo working- 10/10 (if not working - 0)
2. Implemented Viterbi and Clarity on Viterbi- 5/5
3. Transition and Lexical tables clearly described- 5/5
4. Confusion matrix drawn and error analysed- 5/5
5. **Overall F_1-score**
   - **a.** **> 90** - 10/10
   - **b.** **>80 & <=90** - 8/10
   - **c.** **>70 & <=80** - 7/10
   - **d.** **so on**.
6. Unknown word handling- done (5/5; else 0)
7. Benchmarking against ChatGPT (10 else 0)

Note: Must have GUI, otherwise no mark will be given for demo.