CS626 - Speech, Natural Language Processing, and the Web

Assignment-1a
POS Tagging Using HMM

Group Id- 37

Amish Sethi, 22B3029, BS. Economics
Ashutosh Agarwal, 22B2187, B.Tech. Mechanical
Jayesh Ahire, 22B2101, B.Tech. Mechanical
Pratham Agarwal, 22B2111, BS. Economics

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} $\text{quick}_{\text{ADJ}}$ $\text{brown}_{\text{ADJ}}$ fox_{NOUN} $\text{jumps}_{\text{VERB}}$ 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)

- Tokenization to define the sentence boundaries
  - start_token = '^'
  - end_token = '$'

- Lowercase all the tokens in the sentence, reducing the size of the vocabulary.

```
result=[]
for i in range(len(tagged_sentences)):
    temp=[]
    sentence = tagged_sentences[i]
    temp.append((start_token,start_token))
    for word,tag in sentence:
        temp.append((word.lower(),tag))
    temp.append((end_token,end_token))
    result.append(temp)

tagged_sentences = result
```

Overall performance

- Precision : 0.9484
- Recall : 0.9348
- F-score (3 values)
  - F_1-score: 0.9334
  - $F_{0.5}$-score: 0.9364
  - F_2-score: 0.9339

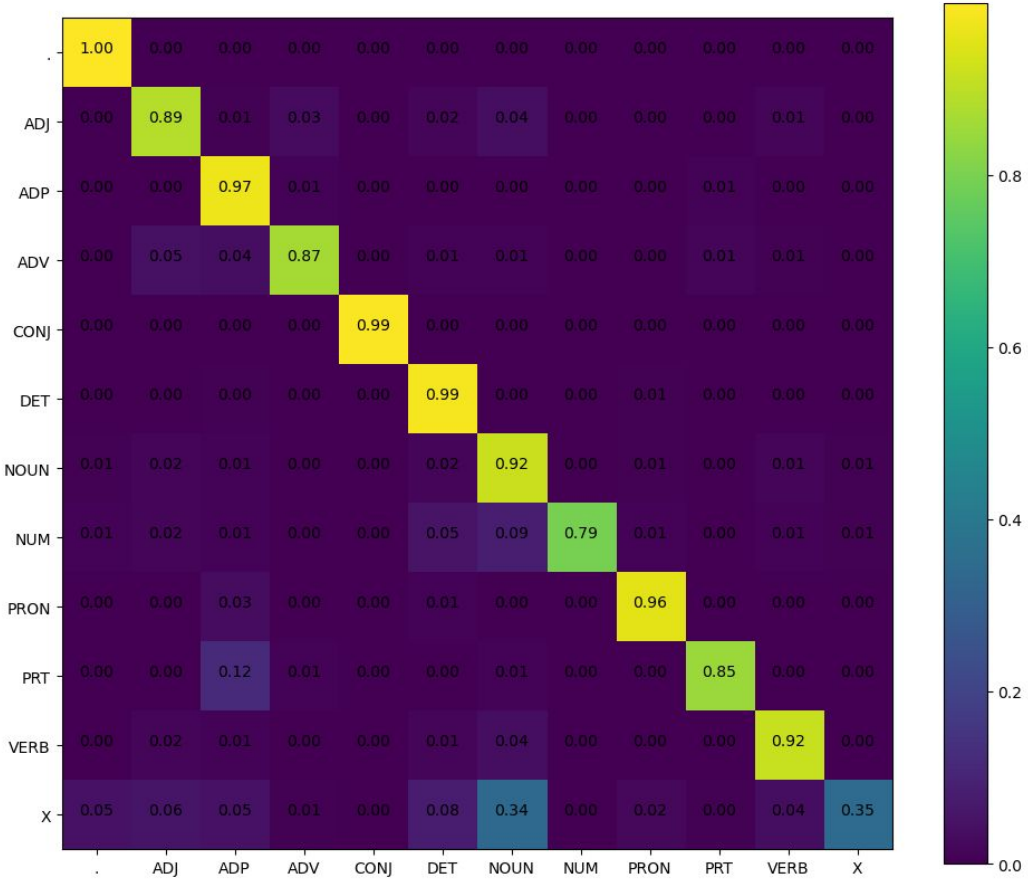
The general formula for non-negative real β is:

$$F_\beta = \frac{(1 + \beta^2) \cdot (\text{precision} \cdot \text{recall})}{(\beta^2 \cdot \text{precision} + \text{recall})}$$

Per POS performance

	precision	recall	f1-score	support
.	0.98	1.00	0.99	29490
ADJ	0.87	0.89	0.88	16572
ADP	0.92	0.97	0.94	29106
ADV	0.90	0.86	0.88	11214
CONJ	0.99	0.99	0.99	7631
DET	0.92	0.99	0.95	27536
NOUN	0.95	0.92	0.93	55064
NUM	0.99	0.80	0.88	2958
PRON	0.93	0.96	0.94	9933
PRT	0.91	0.85	0.88	5916
VERB	0.97	0.91	0.94	36732
X	0.15	0.31	0.21	279

Confusion Matrix (12 X 12)



Interpretation of confusion (error analysis)

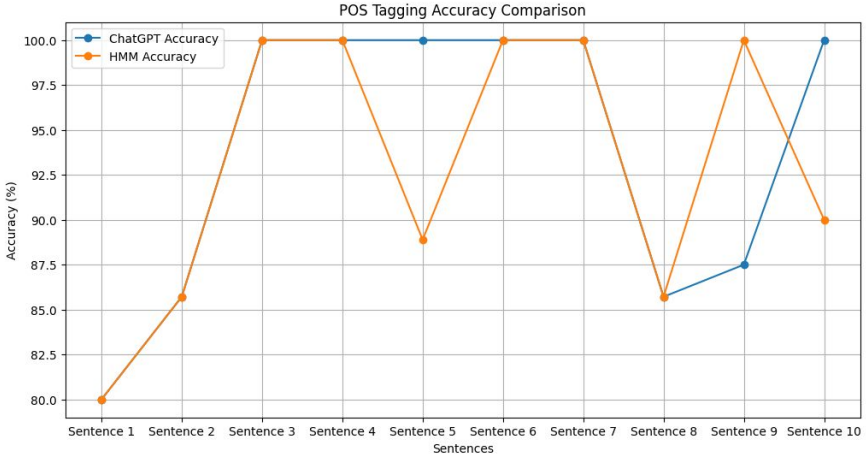
- By trying out many examples, we see that the tag **Noun** and **Verb** are getting confused the most by our model.
- There are some words like Running, fish which are getting confused
- Reasons:
  1. The model is not getting the contextual understanding
  2. There is training data imbalance.

Inferencing/Decoding Info

- Viterbi is implemented as follows:
  - **prev dictionary** stores P(word|tag) of the last seen word, i.e., previous level.
  - **curr dictionary** will store P(word|tag) of curr word and for each iteration update its value accordingly. After completion of one level, the curr is stored in prev.
  - **parent dictionary** helps with backtracking to find the best possible tag for each word in given sentence.
  - **Backtacking formula**:
    - final_tags[i] = parent[i+1][final_tags[i+1]].
  - It tries to estimate the level of tag (i) due to which the tag at level (i+1) has high probability.
    - final_tags: List of tags corresponding to the word sequence.

Benchmarking against ChatGPT

- ChatGPT often gets confused between:
  - **VERB / NOUN** and **ADJ / ADV**
    - Running (VBG) is (VBZ) fun (NN)
    - He (PRP) sings (VB) beautifully (ADJ)



- **HMM**:
  - **Strengths**: It performs well on well-defined patterns and sequences.
  - **Weakness**: Limited in handling complex or ambiguous contexts
- **ChatGPT:**
  - **Strengths**: more sensitive to context and fine shades of meaning.
  - **Weakness**: It is likely to overgeneralize and choose more common tags.

Challenges faced

- The HMM model depends on the context captured through the transition probabilities, which may falter when words are equally likely to be from different tags.
- The Viterbi algorithm is not easy to debug. When things are not as expected, it's hard to tell whether the problem comes from transition/emission probability computation or path tracking.
- The transitional probability of some unknown words was 0, requiring the inclusion of residual probabilities to handle unseen word transitions effectively.

Learning

- **Sequential Modeling**: Gained hands-on experience in handling sequence labeling problems, key for tasks like Named Entity Recognition (NER) and Speech Recognition.

- **Probabilistic Modeling**: Built a Hidden Markov Model (HMM) from scratch, learning to compute transition and emission probabilities for sequence-based prediction.

- **Error Analysis & Evaluation**: Performed detailed error analysis using confusion matrices and cross-validation, ensuring model robustness and generalization.

Scalability: Insights from this assignment can scale to other NLP tasks like Chunking, Machine Translation, and Sentiment Analysis.

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 e.g. Lectures notes, videos, blogs etc