CS626 - Speech, Natural Language Processing, and the Web

Assignment-1a POS Tagging Using HMM

Group Id- 61

Samkit Palrecha, 22b0328, chemical

Ayush raj, 22b0410, chemical

Raunak kumar, 21b090016, maths

Vibhu verma, 21b090020, maths

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, ADV, NOUN, NUM, ADP, PRT, CONJ, DET, PRON, VERB, X, .]
- k-fold cross validation (k=5)

Data Processing Info (Pre-processing)

- The code first tokenizes the corpus by splitting each sentence into individual words and their corresponding tags.
- During this process, words are converted to lowercase to ensure uniformity.
- After tokenization, the words and tags are separated into two lists (words and tags).
- Finally, these are recombined into word-tag tuples for each sentence, resulting in combined_sents, which is a structured format ready for training the HMM model.

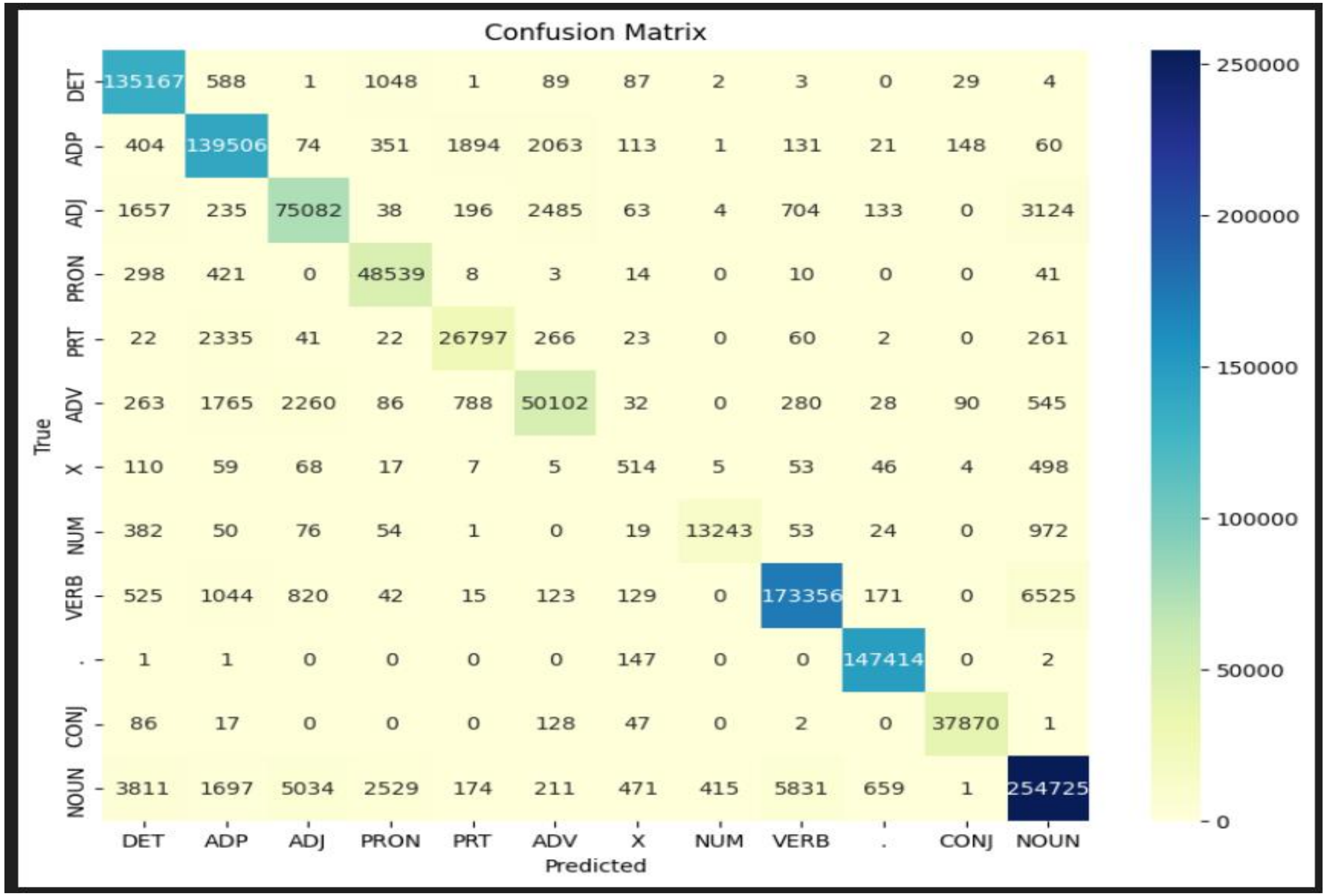
Overall performance

- Precision: 0.91
- Recall: 0.85
- F1-score: 0.86
- F0.5-score: 0.90
- F2-score: 0.86

Per POS performance

- Tag: DET Precision: 1.0000 Recall: 1.0000 F1-score: 1.0000 Support: 4
- Tag: PRON Precision: 1.0000 Recall: 1.0000 F1-score: 1.0000 Support: 1
- Tag: ADJ Precision: 1.0000 Recall: 0.5000 F1-score: 0.6667 Support: 4
- Tag: ADP Precision: 0.8000 Recall: 1.0000 F1-score: 0.8889 Support: 4
- Tag: VERB Precision: 1.0000 Recall: 0.6667 F1-score: 0.8000 Support: 3
- Tag: "." Precision: 1.0000 Recall: 1.0000 F1-score: 1.0000 Support: 3
- Tag: NOUN Precision: 0.5714 Recall: 0.8000 F1-score: 0.6667 Support: 5

Confusion Matrix (12 X 12) (heat map)



	DET	ADP	ADJ	PRON	PRT	ADV	X	NUM	VERB	.	CONJ	NOUN
DET	135167	588	1	1048	1	89	87	2	3	0	29	4
ADP	404	139506	74	351	1894	2063	113	1	131	21	148	60
ADJ	1657	235	75082	38	196	2485	63	4	704	133	0	3124
PRON	298	421	0	48539	8	3	14	0	10	0	0	41
PRT	22	2335	41	22	26797	266	23	0	60	2	0	261
ADV	263	1765	2260	86	788	50102	32	0	280	28	90	545
X	110	59	68	17	7	5	514	5	53	46	4	498
NUM	382	50	76	54	1	0	19	13243	53	24	0	972
VERB	525	1044	820	42	15	123	129	0	173356	171	0	6525
.	1	1	0	0	0	0	147	0	0	147414	0	2
CONJ	86	17	0	0	0	128	47	0	2	0	37870	1
NOUN	3811	1697	5034	2529	174	211	471	415	5831	659	1	254725

Interpretation of confusion (error analysis)

Maximal Confusions:

- **ADP (Adpositions) confused with NOUN**:
    - ADP (139,506 correctly tagged) but about **2,063 instances were confused with NOUN**.
    - **Reason**: Certain prepositions can be easily confused with nouns depending on their context. For example, words like "before" and "after" can function both as prepositions and nouns.
- **PRT (Particles) confused with ADV (Adverbs)**:
    - PRT (26,797 correctly tagged) but **2,335 instances were confused with ADP**.
    - **Reason**: Some particles can be mistaken for adverbs due to similar syntactic functions. Words like "up" and "off" can act as both particles and adverbs based on usage.
- **ADV (Adverbs) confused with ADJ (Adjectives)**:
    - ADV (50,102 correctly tagged) but **2,260 instances confused with ADJ**.
    - **Reason**: Many adverbs and adjectives have the same or similar forms, especially in comparative or superlative forms. For example, “fast” can be both an adjective and an adverb.

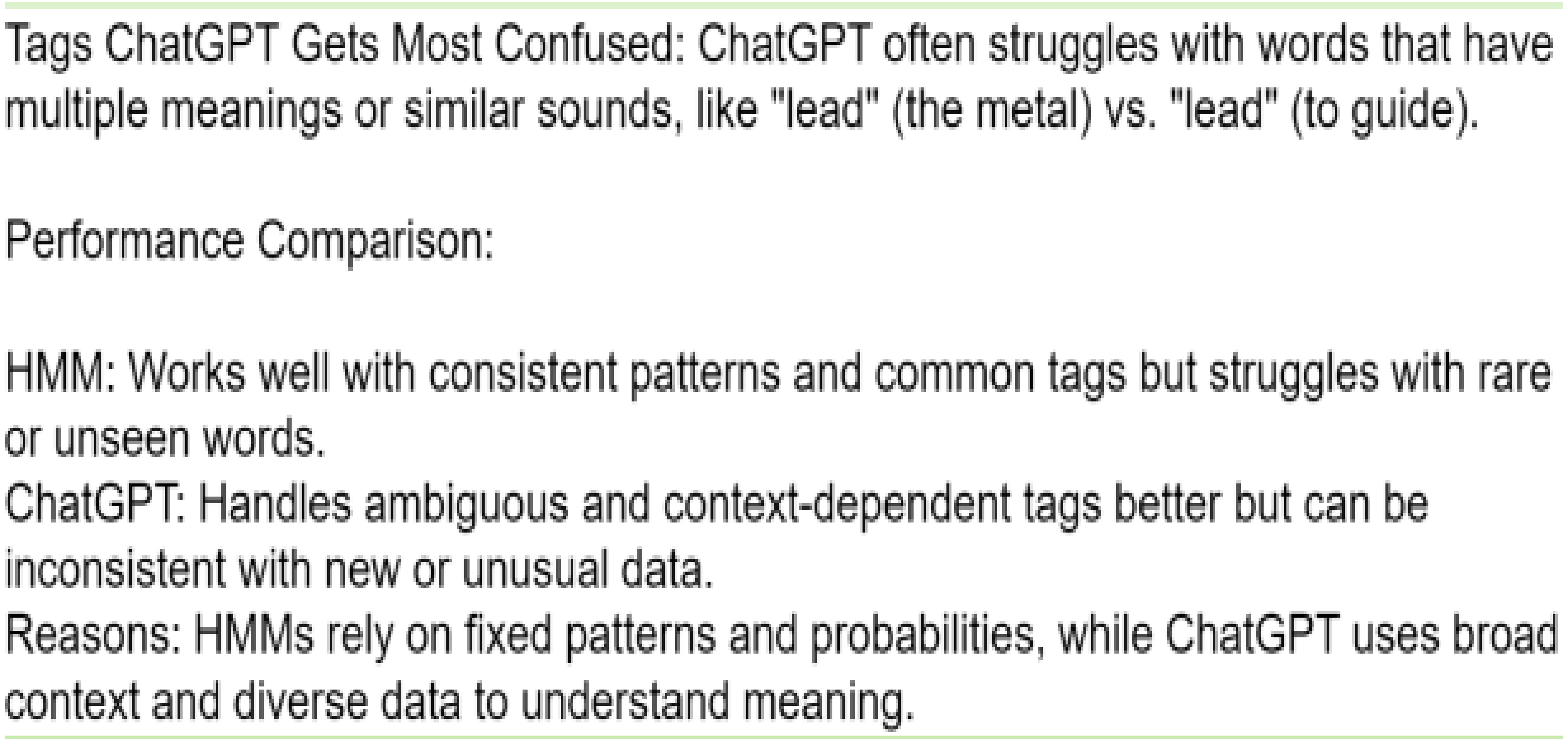
Interpretation of confusion (error analysis)

- **VERB (Verbs) confused with NOUN**:
  - VERB (173,356 correctly tagged) but **6,525 instances confused with NOUN**.
  - **Reason**: Gerunds (verb forms ending in -ing) and infinitives can often be confused with nouns. Additionally, some verbs used as nouns (e.g., "run" in "the run") lead to ambiguity.
- **NOUN confused with ADJ (Adjectives)**:
  - NOUN (254,725 correctly tagged) but **3,124 instances confused with ADJ**.
  - **Reason**: Noun-adjective confusion often occurs when nouns are used attributively (e.g., "stone wall" where "stone" functions like an adjective).
- **NUM (Numbers) confused with NOUN and ADJ**:
  - NUM (13,243 correctly tagged) but **1,972 instances confused with NOUN and ADJ**.
  - **Reason**: Some numbers, particularly when spelled out (e.g., “first”, “second”), can function both as adjectives or nouns, depending on the sentence structure.

Inferencing/Decoding Info

The Viterbi implementation starts by initializing base cases for the first word in the sentence using the start probabilities and emission probabilities for each tag. It then iterates through the sentence, updating probabilities for each word using the highest probability of the previous tag, the transition probabilities, and the emission probabilities. At each step, the best tag path is tracked. Finally, it selects the most likely path at the end of the sentence based on the highest probability.

Benchmarking against ChatGPT



Tags ChatGPT Gets Most Confused: ChatGPT often struggles with words that have multiple meanings or similar sounds, like "lead" (the metal) vs. "lead" (to guide).

Performance Comparison:

HMM: Works well with consistent patterns and common tags but struggles with rare or unseen words.
ChatGPT: Handles ambiguous and context-dependent tags better but can be inconsistent with new or unusual data.
Reasons: HMMs rely on fixed patterns and probabilities, while ChatGPT uses broad context and diverse data to understand meaning.

Challenges faced

- **Matrix Initialization**: Setting up the correct probability matrices for the start, transition, and emission probabilities can be tricky. If there are missing or zero values, the algorithm might fail or misclassify.
- **Handling Unknown Words**: If the word in the sentence doesn’t appear in the training data, handling such cases in the emission probabilities might have been challenging. This often results in assigning an "X" or default tag.
- **Efficient Computation**: The algorithm involves iterating over all possible tags for every word in the sentence, which can become computationally expensive for large tag sets or sentences. Optimizing this process to avoid memory issues can be challenging.
- **Numerical Underflow**: Due to the multiplication of small probabilities, numerical underflow could occur. Handling this with log probabilities may have been something to consider.
- **Decoding the Best Path**: Conceptualizing how the algorithm traces back the optimal sequence of tags (based on the maximum probability path) across the entire sentence may have been confusing, especially in dynamic programming contexts.
- **Zero Probabilities**: Dealing with zero probabilities in emission or transition matrices (when a word-tag or tag-tag pair never appears in the training data) can introduce complexity.

Learning

This assignment provided a deeper understanding of the Viterbi algorithm and Hidden Markov Models (HMMs) for POS tagging, highlighting the importance of managing probabilities and efficiently decoding sequences. Key takeaways include:

1. **Probabilistic Models**: Learning how to understand, manage and interpret probabilities for sequences and states is crucial for various NLP tasks.
2. **Dynamic Programming**: The Viterbi algorithm’s use of dynamic programming to optimize sequence prediction is applicable to other sequence-based tasks, such as Named Entity Recognition (NER) and machine translation.
3. **Error Handling**: Addressing issues like unknown words and zero probabilities can improve model robustness and is relevant for scaling models to handle diverse text corpora.
4. **Scalability**: Understanding how to efficiently implement and scale algorithms helps in applying similar techniques to larger datasets and more complex models, such as deep learning for NLP tasks.
5. **Handling Sparse Data**: Techniques for smoothing and handling sparse data (e.g., unseen words or rare tags) are essential for building robust models across different NLP tasks, ensuring better generalization and performance.

References

- For Brown corpus: http://www.nltk.org/nltk_data

- For GUI: Gradio

- Other references:
- https://www.geeksforgeeks.org/viterbi-algorithm-for-hidden-markov-models-hmms/
- https://www.mygreatlearning.com/blog/pos-tagging/#optimizing-hmm-with-viterbi-algorithm