POS Tagging Using HMM

CS626 - Speech, Natural Language Processing, and the Web

Group Id - 74
Shubham Hazra, 210100143, CSE
Om Godage, 21D100006, CSE
Akshat Singh, 210020013, CSE
Cheshta Damor, 210050040, CSE

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.
- **Tag Set:** Universal Tag Set (12 tags in total).
  - ADJ (Adjective)
  - ADP (Adposition)
  - ADV (Adverb)
  - AUX (Auxiliary)
  - CONJ (Conjunction)
  - DET (Determiner)
  - NOUN (Noun)
  - NUM (Numeral)
  - PRT (Particle)
  - PRON (Pronoun)
  - . (Punctuation)
  - X (Unknown)
- **k-fold cross validation:** $k = 5$.

Data Processing (Pre-processing)

- Lower casing: All words in the dataset are converted to lowercase
- Adding Hat ^ : Each sentence in the dataset is prepended with a special sentence start token ^ and its corresponding POS tag -.
- Word vocabularies and POS tag vocabularies are made, which is further used to create mappings from word to tags and vice-versa.

Overall Performance

- **Precision: 0.8746**
- **Recall: 0.9344**
- **F-score:**
  - F1-score: 0.9035
  - F0.5-score: 0.8860
  - F2-score: 0.9218

Per POS Performance

Tag	Precision	Recall	F1 Score
.	0.9839	0.9996	0.9916
ADJ	0.9128	0.9170	0.9149
ADP	0.9544	0.9657	0.9600
ADV	0.8795	0.9025	0.8908
CONJ	0.9632	0.9925	0.9777
DET	0.9772	0.9864	0.9818
NOUN	0.9704	0.9344	0.9520
NUM	0.8428	0.9302	0.8843
PRON	0.9559	0.9829	0.9692
PRT	0.8874	0.8982	0.8928
VERB	0.9788	0.9497	0.9640
X	0.1968	0.8247	0.3177
Average	0.8746	0.9344	0.9035

Table: Precision, Recall, and F1 Score per tag with average values.

Confusion Matrix

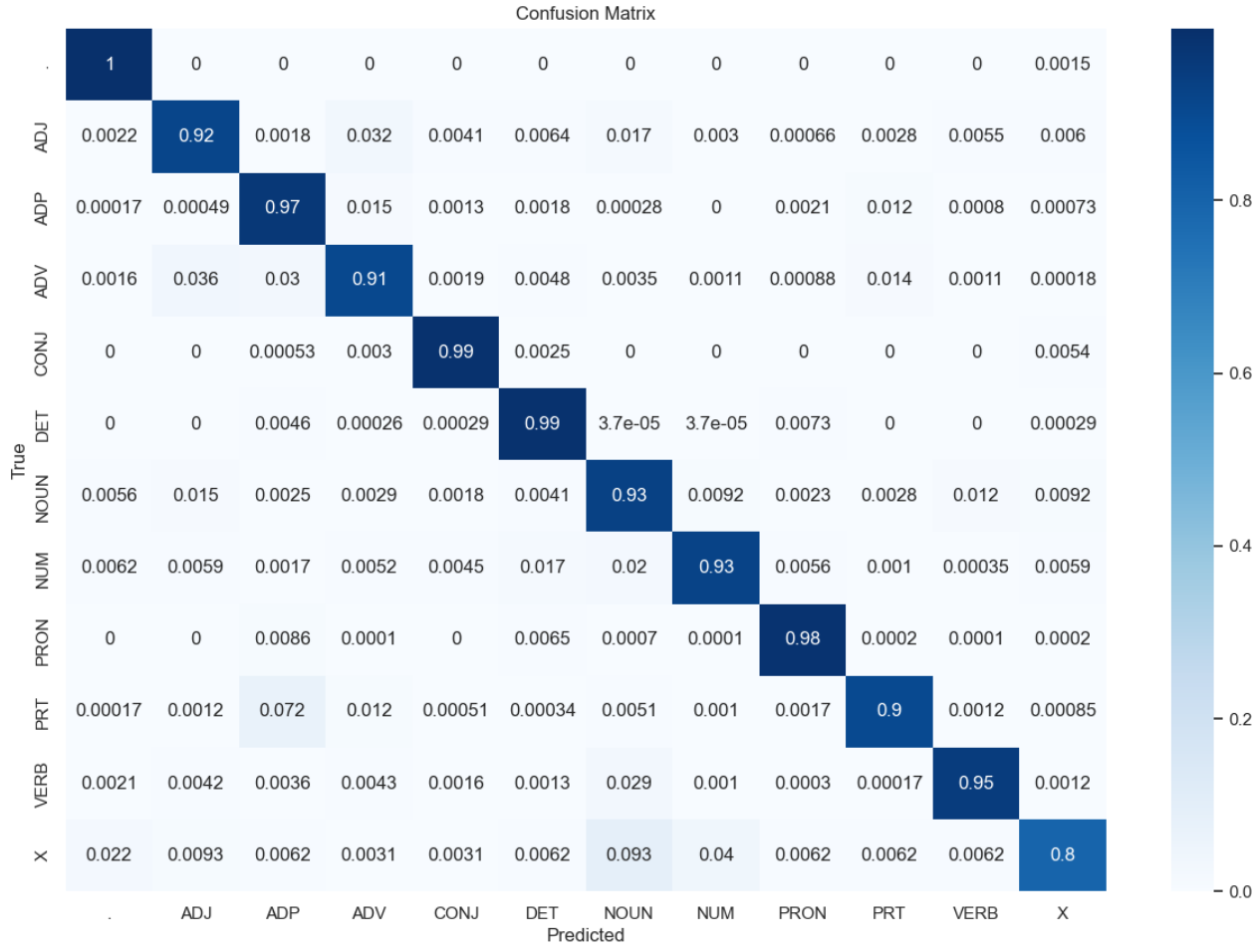


Figure: Confusion Matrix including all POS tags

Error Analysis

- The VERB tag is confused the most with NOUN tag as multiple words are widely used as nouns but in certain cases those words act as verbs in sentences.

Example 1: Sentence: "Buffalo buffaloes Buffalo buffaloes buffalo buffalo Buffalo buffaloes"
Result: ['NOUN', 'NOUN', 'NOUN', 'NOUN', 'NOUN', 'NOUN', 'NOUN', 'NOUN', 'NOUN']
Expected Result: ['NOUN', 'NOUN', 'NOUN', 'VERB', 'NOUN', 'VERB', 'NOUN', 'VERB']
Example 2:
Sentence: "The old man the boats"
Result: ['DET', 'ADJ', 'NOUN', 'DET', 'NOUN']
Expected Result: ['DET', 'ADJ', 'VERB', 'DET', 'NOUN']

Inferencing/Decoding Information

- **Preprocessing**: Each sentence undergoes preprocessing to prepare it for tagging.
- **Base Case**: Initialize the start of the sentence by adding tags and their associated probabilities.
- **Inductive Step**:
  - For each word, iterate over possible previous and current tags.
  - Compute the log probabilities for all possible tag transitions.
  - Select the transition with the highest probability for each tag.
- **Lexical Probabilities**:
  - Consider lexical probabilities for each word.
  - If a word is not in the vocabulary, use the minimum lexical probability available.
- **Tracking**:
  - Store log probabilities and corresponding best tags in an array.
  - Backtrack through the array to determine the optimal tags for each word in the sentence.

Benchmarking against ChatGPT

- ChatGPT tends to confuse tags that are context-dependent or tags involving ambiguous word usage or complex sentence structures. For instance
  Example: foxes foxes fox fox foxes
  GPT: 'NOUN', 'VERB', 'VERB', 'NOUN', 'VERB'
  HMM: 'ADP', 'DET', 'NOUN', 'NOUN', '.'
- HMM might perform better in specific domains where the transition probabilities align well with the tag sequences seen during training.
- ChatGPT's pre-trained models can generalize well across different text styles or languages, potentially offering better performance in varied contexts not seen during HMM training.
- The performance of HMM heavily depends on the quality and quantity of training data.

Benchmarking against ChatGPT (continued)

- Performance of Gemini for each POS (50 sentences)

Figure: Confusion Matrix including all POS tags - Gemini

Benchmarking against ChatGPT (continued)

- Average accuracy = 0.8818

Tag	Precision	Recall	F1 Score
.	1.0000	0.6163	0.7626
ADJ	0.8205	0.9143	0.8649
ADP	0.9848	0.7927	0.8784
ADV	0.7692	0.8333	0.8000
CONJ	0.5200	1.0000	0.6842
DET	0.9765	0.8925	0.9326
NOUN	0.9700	0.9848	0.9773
NUM	1.0000	1.0000	1.0000
PRON	0.6154	0.9412	0.7442
PRT	0.6875	0.7857	0.7333
VERB	0.9910	0.9322	0.9607
Average	**0.8486**	**0.8812**	**0.8489**

Table: Precision, Recall, and F1 Score per Tag - Gemini

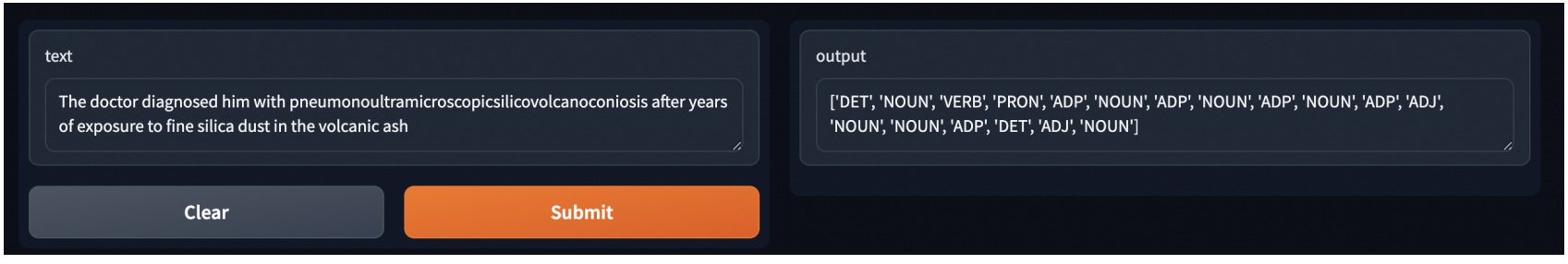
Benchmarking against ChatGPT (continued)

- Cases where Gemini predicts Particle as adverb. Eg: up, out, down

Challenges Faced

- Handling Out-of-Vocabulary Words: When a word is not present in the vocabulary, its tagging is influenced by:
  - The log of the minimum lexical probability of any word, which is used as a fallback.
  - The most likely transition probability given the previous tag.
- Challenges:
  - This approach can lead to inconsistent tagging for the same word depending on its context (i.e., the previous tag).
  - Ensuring that the fallback probability is suitably chosen to balance between unseen words and observed transitions.

Inconsistent Tagging



Learning

- We gained a strong understanding of HMMs and the Viterbi algorithm for POS tagging, focusing on handling unknown words, smoothing, and optimizing transitions and emissions
- This learning can be useful for other layers that are build on top of POS tagging, like semantics, and also be applied to other tasks such as Machine Translation and Speech Recognition

References

1. For Brown corpus: `http://www.nltk.org/nltk_data`
2. For GUI:
   - `https://www.gradio.app/`