Name: Nilakshi Nagrale

Assignment: CSCI-544 Assignment 3 - Part-of-Speech Tagging Using HMM

USC ID: 2403347301

This assignment focuses on implementing Hidden Markov Models (HMMs) for Part-of-Speech (POS) tagging using the Wall Street Journal section of the Penn Treebank. The implementation steps were – constructing a vocabulary, training an HMM model, & performing POS tagging using both greedy decoding & Viterbi decoding methods.

Steps:

Step 1: Understanding the Dataset:

- The dataset had three files: train, dev, & test.
- The train & dev files contain sentences with words & their corresponding POS tags.
- The test file contains sentences without POS tags, requiring model-based prediction.
- Each line in the dataset follows the format (tab-spaced): word_index \t word \t POS_tag.

Step 2: Vocabulary Creation:

- Extracted all words from the train dataset.
- Counted the occurrences of each word.
- Replaced words appearing fewer than a chosen threshold (2) with the special token <unk>.
- Sorted words in descending order based on frequency.
- Saved the vocabulary in vocab.txt with each line formatted as: word \t index \t occurrences.
- Calculated the total size of the vocabulary & occurrences of <unk>.

Q&A:

1) What is the selected threshold for unknown word replacement?

   Selected threshold for unknown words: 2

2) What is the total size of your vocabulary and what are the total occurrences of the special token '<unk>' after replacement?

   Vocabulary size: 23183

Total count of special token <unk>: 20011

Step 3: HMM Model Learning:

- Computed the transition probabilities:
  - Formula: $t(s' \mid s) = count(s \rightarrow s') / count(s)$
  - Counted occurrences of POS tag transitions from train data.
- Computed the emission probabilities:
  - Formula: $e(x \mid s) = count(s \rightarrow x) / count(s)$
  - Counted occurrences of word-POS tag pairs.
- Stored computed probabilities in a JSON file hmm.json containing:
  - transition: A dictionary mapping (s, s') to t(s'|s).
  - emission: A dictionary mapping (s, x) to e(x|s).
- Determined the total number of transition & emission parameters.

<u>Q&A:</u>

1) How many transition and emission parameters in your HMM?

   Number of Transition Parameters = 2025

   Number of Emission Parameters = 1043235

Step 4: Greedy Decoding with HMM:

- Implemented the greedy decoding algorithm – For each word in a sentence, selected the POS tag with highest emission probability given the previous tag.
- Evaluated the model on dev data & recorded accuracy.
- Predicted POS tags for test data & saved results in greedy.out.
- Used eval.py script to evaluate performance. (for dev file)

<u>Q&A:</u>

1) What is the accuracy of the dev data?

   Greedy Decoding Algo Accuracy: 0.935 (93.5%)

Step 5: Viterbi Decoding

- Implemented the Viterbi algorithm for more accurate POS tagging – Used DP to determine the most probable sequence of tags for each sentence.
- Evaluated the model on dev data & recorded accuracy.
- Predicted POS tags for test data & saved results in viterbi.out.
- Used eval.py script for performance evaluation. (for dev file)

Q&A:

2) What is the accuracy of the dev data?

   Viterbi Decoding Algo Accuracy: 0.9481 (94.81%)

Conclusion

This assignment involved implementing an HMM-based POS tagging system, covering vocabulary creation, model learning, & decoding methods. The results can be evaluated using the given eval.py script.

USC ID: 2403347301