ELL884: POS Tagging

Aditya Arya (mt6210958@iitd.ac.in)
10 February 2024

1 Overview

As part of Assignment 1, developed a Parts Of Speech Tagger using Hidden Markov Model (HMM) Maximum Entropy Markov Model (MEMM) from scratch.

2 Hidden Markov Model

2.1 Initialisation

I initialised Transition Matrix (A), Emission Probability Matrix (B), and Initial State Probabilities (PI) using Maximum Likelihood Estimate. Employed Laplace Smoothing for catering observation occurences not in training corpus. [1]

$$a_{tag_i,\ tag_j} = \frac{count(tag_i, tag_j) + \alpha}{count(tag_i) + \alpha \cdot N}$$

$$b_{tag_i,\ word_j} = \frac{count(tag_i, word_j) + \alpha}{count(tag_i) + \alpha \cdot V}$$

$$\pi_{tag_i} = \frac{count(start, tag_i) + \alpha}{count(tag_i) + \alpha \cdot N}$$

Here, α : smoothing parameter, N : Total Tags, V : Number Of Words in Vocabulary

2.2 Training

I used Baum Welch Algorithm [2] to train A, B, PI. Since the initialisation approach was to Maximize Likelihood on training data, I fine-tuned A, B, and PI on sentences encountered in test data. This allowed for better fitting, and improvement in accuracy vis-a-vis learning over train data which leads to overfitting. The score achieved on kaggle for this approach was best noted at **0.803**

As test data had words which weren't part of training corpus, their values for emission probabilities didn't exist in B. On investigation, 1350 sentences out of 4000 total had an unseen word, however they formed only 2.5% of the total words in training set. Hence they were sparsely distributed across sentences, and each sentence had only 1.5 unseen words on an average.

To overcome this in light of the dataset given, I created another entry for unseen words, "OOV-observations (Out Of Vocabulary) and assigned emission probabilities using k-rare hapex words approach. The approach hypotheses that words unseen in training set are similar in nature to the class of words which appear very rare in the training set. k is the threshold frequency for consideration, a hyperparameter. I set it to 5 in this case.

$$b_{tag_i,OOV} = \frac{count(tag_i) + \alpha}{count_{f(words)<k}(words) + \alpha \cdot N}$$

One of the issues faced in regard to training was of time constraints, as each sentence took approximately

50-55 seconds on an average for one epoch, which was sufficient for most cases given the initialisation. Due to training time constraints, I randomly sampled 40% for the sentences with OOV observations to be trained. For sentences without OOV, evaluating without EM learning gave high accuracy, and improvement in score after learning over them was marginal to the time spent in learning. Hence, I chose to directly evaluate such cases.

2.3 Approaches

- Basic Implementation, assigning NN tag (majority vote rule) to unseen words : Score - **0.709**
- Alternate handling of unseen words, emission probabilities of unseen word being a Maximum Likelihood estimate over tags seen : Score - **0.762**

$$b_{tag_i,\ OOV} = \frac{count(tag_i) + \alpha}{count(total\ words) + \alpha \cdot N}$$

- Final Approach, using Hapex words and Baum Welch Algorithm : Score - **0.803**

3 Maximum Entropy Markov Model

It is a sequence classifier that combines the principles of Markov models with maximum entropy modeling accounting feature considerations pertaining to each observation, which was missing in our HMM approach [3]

The decode algorithm is similar to Viterbi as in HMM, except that of replacing terms with transition and emission probabilities with $P(q_j|q_i, o_t)$, which we compute in this model.

3.1 Feature Selection

I have taken into account two set of features, one depending on the previous state and the other on the present observation to predict the tag of the present observation. The former is the frequency of occurence of each tag preceding the word, and for the latter I have taken into account whether the word was the start/end of the given sequence.

4 Notebook

`https://www.kaggle.com/code/arya92/hmm-memm/`

Referenties

[1] J. Diesner en C. Carley. "Part of Speech Tagging for English Text Data". In: *School of Computer Science, Carnegie Mellon University* (2005). URL: `https://www.cs.cmu.edu/~epxing/Class/10701-06f/project-reports/diesner.pdf`.

[2] Dan Jurafsky en James H Martin. *Speech and Language Processing*. Pearson, 2019.

[3] Adwait Ratnaparkhi. "Maximum Entropy Markov Models for Information Extraction and Segmentation". In: *Proceedings of the International Conference on Machine Learning* ().