Assignment 1 - Demo
HMM for Sentence Parsing

Khyati Patel, 200050102
Koustubh Rao, 200100176
Gowri Sriya Mannepalli, 200050043

02-09-2022

Problem Statement

- Given a sequence of words, produced the POS tag sequence
- Technique to used: HMM-Viterbi
- Used Universal Tag Set (12 in number): tagset = ['NOUN', 'VERB', '.', 'ADP', 'DET', 'ADJ', 'ADV', 'PRON', 'CONJ', 'PRT', 'NUM', 'X']
- 5-fold cross validation

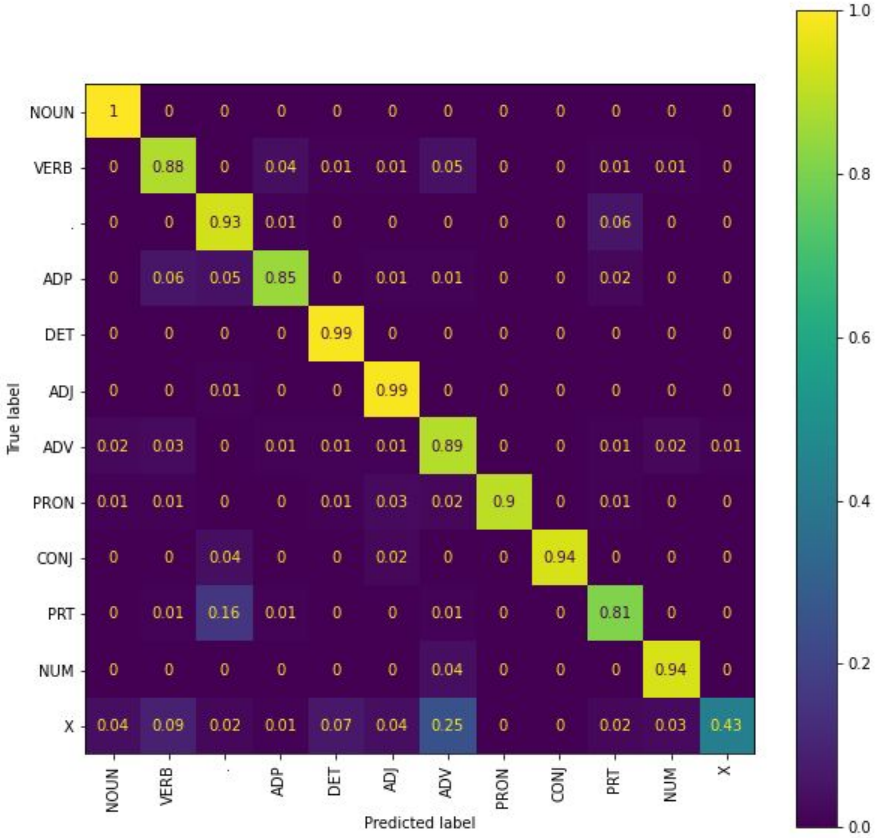
Overall performance

- Precision - 0.9305
- Recall - 0.9261
- F-score (3 values)
  - F1-score - 0.9273
  - F0.5-score - 0.9290
  - F2-score - 0.9290

Per POS performance (P,R,F1,F0.5,F2)

- NOUN: 0.95,0.89,0.92,0.94,0.90
- VERB: 0.97.0.94,0.95,0.96,0.94
- . : 0.97,1.00,0.98,0.97,0.99
- ADP : 0.92,0.93,0.93,0.92,0.93
- DET : 0.95,0.99,0.97,0.96,0.98
- ADJ : 0.84,0.88,0.86,0.85,0.87
- ADV : 0.88,0.85,0.86,0.87,0.86
- PRON: 1.00,0.94,0.97,0.98,0.95
- CONJ : 0.88,0.99,0.93,0.90,0.97
- PRT : 0.65,0.81,0.72,0.68,0.77
- NUM : 0.96,0.90,0.93,0.95,0.91
- X : 0.17,0.43,0.24,0.19,0.32

Confusion Matrix (12 X 12)



Interpretation of confusion matrix (error analysis)

- VERB->ADV, .->PRT, ADJ->., ADV->VERB, PRON->ADJ, NUM->ADV, X->ADV
- ADV/ADP->VERB

  Example: words like ‘right’ and ‘like’ can be both adverb and verb

  I like sweets.(verb)

  Like as a father pitieth his own children.(adverb)

  That is a fault that will right itself.(verb)

  He stood right in my way.(adverb)
- PRON->ADJ

  Example: words like ‘some’ can be both pronoun and adjective

  Some men are wise and some are foolish.

  Reasoning:

  In both these cases as emission and transition probabilities can be considerably equivalent for some words there is a possibility of confusion in POS tagging for these words

Data Processing Info (Pre-processing)

- Pre-processing step “Unzipping brown corpus”: Done using nltk function for brown corpus, “nltk.corpus.brown.tagged_words”. This would break our dataset to form a list of tuples, each tuple being (token, tag)
- Pre-processing step “Lower casing”: Converted all tokens (both for training data and testing data) into lowercase letters before passing into Viterbi function
- Transition Probabilities: Iterated through every consequent pair(say, [tag_i,tag_j]) of tags in training data, and found frequencies of tag_j occurring after tag_i. Transition probability would be these frequencies divided by the total number of tokens having tag_i.
- Emission Probabilities: Iterated through all the tokens, and stored frequencies of token having POS tag tag_i in tagToken[token_i][tag_i]. Emission probability would be these frequencies divided by the total number of tokens having tag_i.
- Handling unseen words in Viterbi: Considered the probability of unseen word belonging to each of the 12 POS tags is equal, i.e, emission probability = 1/(1+tag_count[tag_i])

Inferencing/Decoding Info

A loop with iterations equal to the number of words in test sentence was run. A dictionary with the universal tagset as keys and tag frequencies following “.” was initialized. For each word the values of the dictionary were updated with the emission probabilities for that word. In case an unseen word was encountered, we updated the dictionary with 1/(1+tag_count) instead. The motivation is that if this word were a part of the training corpus then 1/(1+tag_count) would have been the emission probability. Then a nested for loop was run to update the values with transition probabilities and to choose the maximum for each tag.

To prevent overflow or underflow, tagging was done in batches of 5 or 7. At the end of each batch, the values for each tag was set equal to their tag frequencies.

Any thoughts on generative vs. discriminative POS tagging

- Discriminative POS tagging models (an example being the Maximum Entropy Markov Model sir talked about in class) use probability estimates and maximum likelihood to find conditional probability P(y|x), whereas generative POS tagging models (an example being the above implemented HMM) use Bayes theorem to find joint probability P(y,x).
- Generative POS tagging models like HMM based POS tagging cannot handle “free word order” and “agglutination” well

~ If adjective after noun is equally likely as adjective before noun, the transition probability is no better than uniform probability which has high entropy and is uninformative.

~ When the words are long strings of many morphemes, POS tagging w/o morph features is highly inaccurate.