Assignment-Discussion POS Tagging

Singamsetty Sandeep, 213050064
Anmol Namdev, 213050044
Sakharam Sahadeo Gawade, 213050027

21 August 2021

Problem Statement

- Given a sequence of words, produce the POS tag sequence
- Technique to be used: HMM-Viterbi
- Use Universal Tag Set (12 in number)
- 5-fold cross validation
- Tags: ADJ, ADP, ADV, CONJ, DET, NOUN, NUM, PRT, PRON, VERB, X and **.**

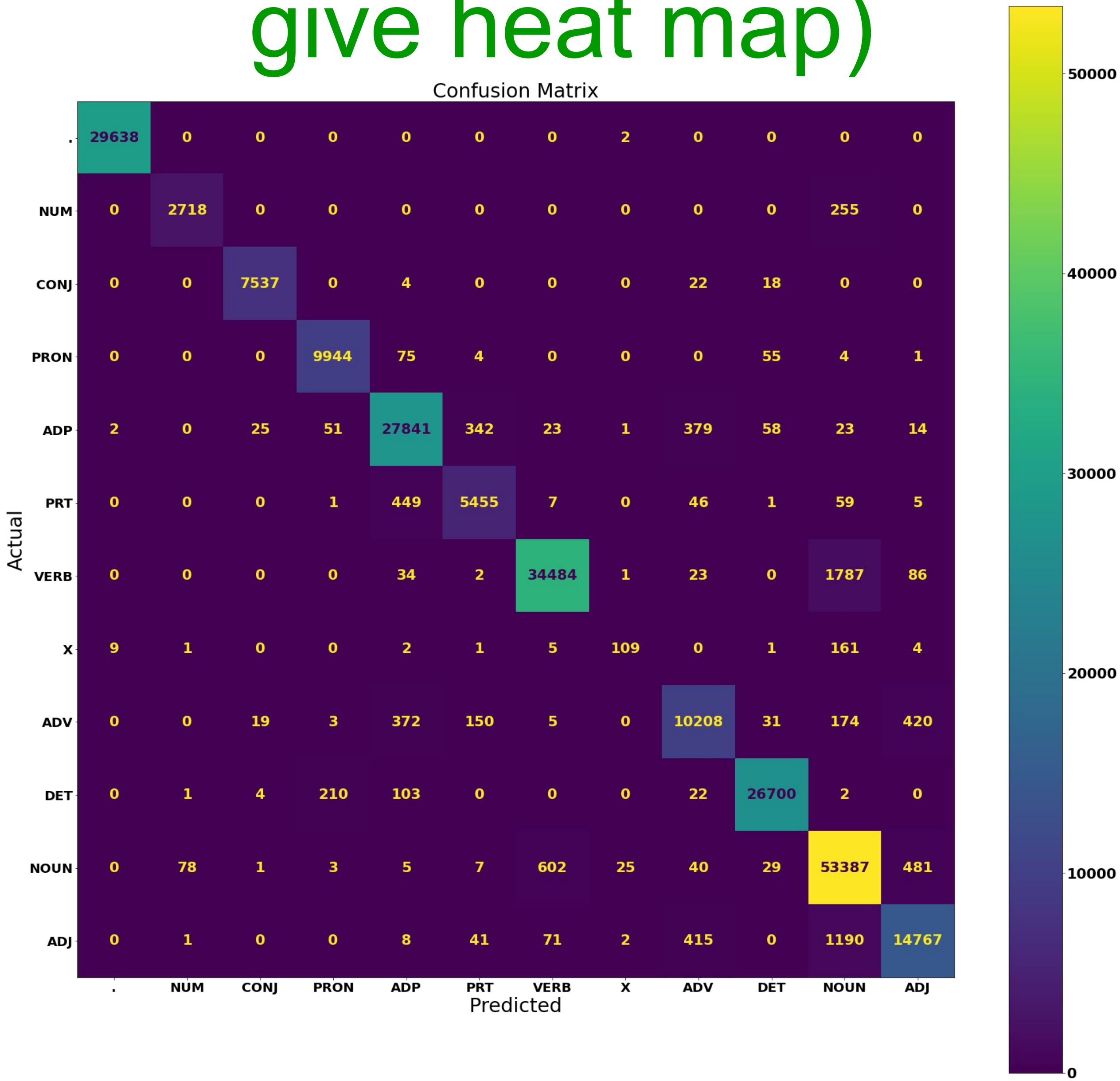
Overall performance

- Precision: 0.962716 ± 0.000113
- Recall: 0.962716 ± 0.000113
- F-score (3 values)
  - F1-score: 0.962716 ± 0.000113
  - F0.5-score: 0.962716 ± 0.000113
  - F2-score: 0.962716 ± 0.000113

Per POS performance

TAG	Precision,	Recall,	F1 Score
ADJ:	0.935756,	0.895844,	0.915364
ADP:	0.963543,	0.966834,	0.965185
ADV:	0.906342,	0.896521,	0.901400
CONJ:	0.993168,	0.994211,	0.993689
DET:	0.992910,	0.986765,	0.989828
NOUN:	0.937492,	0.976785,	0.956735
NUM:	0.970744,	0.916908,	0.943045
PRT:	0.905442,	0.901960,	0.903685
PRON:	0.971924,	0.984805,	0.978321
VERB:	0.979300,	0.946676,	0.962712
X:	0.790199,	0.382647,	0.51430
.:	0.999702,	0.999925,	0.9998143

Confusion Matrix (12 X 12) (can give heat map)



Interpretation of confusion (error analysis)

- Adjective predicted as noun
  - Example: The minister calmed the angry.
- Noun predicted as verb
  - Example: Give it a try.
  - Example: Walking is good for health.
- Verb predicted as Noun
  - Example: I like to read newspaper.
- Noun predicted as Adjective
  - Example: Be good and do good.
  - Example: Race Car, Cricket Bat etc.
- Unseen words in test corpus are tagged as Nouns, that might increase some bias towards being predicted as Nouns

Data Processing Info (Pre-processing)

- Corpus is a combination of lower and upper case- fixed to lower case
- Added start (^^) and end ($$) delimiters to sentences
- If the data has ^^ or $$ they are set to <DEL> and tagged as X
- For Transition Probabilities the the tags of the bigrams of each sentence is used
- For Emission Probabilities each word and its tag for each sentence is used.

Probability Calculations

Lexical Probability:

Formula : # times a word appeared as a TAG / total # TAG appearances

i) we have calculated TAG counts for all the words in the training corpus

Tag_word_counts = {TAG1:{w1:c1,w2:c2,...},TAG2:{w1:c1,...}}

w1,w2 etc. corresponding to each TAG are words that are tagged as a TAG

c1,c2 etc are counts, i.e. c1 = #w1 is tagged as TAG1 etc.

ii) Using the above Tag_word_counts, we calculated Lexical probability

Lexical_prob = {TAG1:{w1:p1,w2:p2,...},TAG2:{w1:p1,...}}

p1,p2 are corresponding lexical probabilities

p1 = # times a w1 appeared as a TAG1 / total # TAG1 appearances

Eg. #times ‘sun’ appeared as Noun / total number of Nouns in dataset

Probability Calculations cont.

Transition Probability:

Bigram Assumption: A tag T1 depends only on the previous tag T0

Formula : # times TAG2 is preceded by TAG1 / total # TAG1 appearances

i) we have calculated Bigram counts for all the tags in the training corpus

Bigram_tag_counts = {TAG1:{tag1:c1,tag2:c2,...},TAG2:{tag1:c1,...}}

TAG and corresponding counts of tags followed for each TAG

ii) Using the above Bigram_tag_counts, we calculated Transition probability

Transition_prob = {TAG1:{tag1:p1,tag2:p2,...},TAG2:{tag1:p1,...}}

p1,p2 are corresponding transition probabilities

p1 = # times tag1 is preceded by TAG1 / total # TAG1 appearances

Eg. #times ‘ADJ' is preceded by Noun / total number of Nouns (that are followed by some tag)

Inferencing/Decoding Info

- For each word in a sentence
  - The probability of best candidates for each tag at previous level is multiplied with the emission probability and the transition probability of possible tags based on current word
  - The best candidate for each tag at the current level is chosen and the previous tag is kept a track of for backtracking
  - The result of forward propagation at each level looks like the following

    LEVEL_K:{TAG1:{best_candidate_among_Tag1,probability_of the best candidate} , {TAG2:{previous_tag_of_best_candidate_ among_Tag2, probability_of the best candidate}, **...**}
  - When backtracking start from the end to start choosing the predicted tag based on LEVEL information and the predicted tag that follows the word.
  - In case an unseen word arrives the next tag is set to ‘NOUN’ and the emission probability is set to a low probability (0.0001). This is based on the observation from the corpus that ≈ 63% of the unseen words were noun.

Any thoughts on generative vs. discriminative POS tagging

- Generative POS tagging is data driven, so accuracy entirely depends on the corpus used for training. On the other hand, discriminative approach when its rule based, has accuracy constraints for ambiguous languages and where rules are not concrete.
- Generative approach eases out the process of calculations and training using mathematical formulations, where discriminative approach, when its rule based, has a lot of computation constraints and complex mathematical formulations.
- Introduction of Bigram assumption in generative approach makes things comfortable, whereas in discriminative approach, this assumption can not be made because of rules depending on the entire sentence or context in some cases.
- If we need to incorporate certain arbitrary rule based features in generative model, adding them into the model becomes a hard task if we need to add lots of them. A feature like a verb ending with ‘s’ is most likely to be in its singular form and a Generative model like HMM might get confused with it and to resolve it, such features need to be incorporated by encoding into either transition or emission probabilities.