Part-of-speech(POS) tagging on the Hidden Markov Model

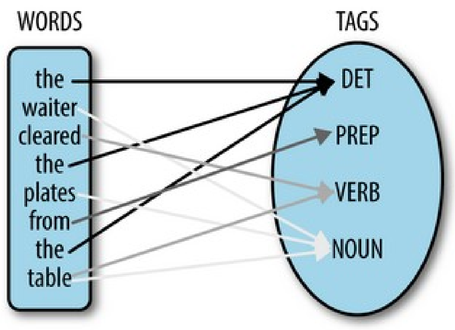
Seungwoo Lee

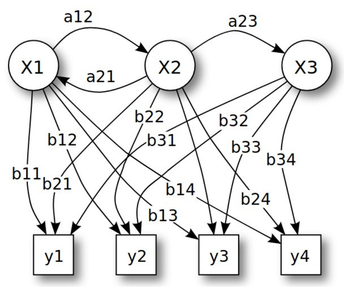
School of Computer Science & Engineering, Seoul National University

lsw8075@gmail.com

Backgrounds

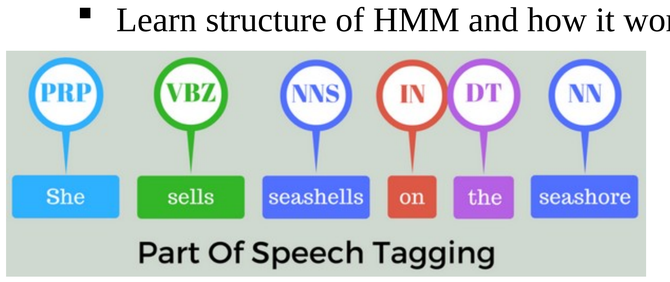
- Part of Speech(POS) is a category of words which have similar grammatical properties.
- POS tagging in sentences is very important for Natural Language Processing(NLP).
- Hidden Markov Model(HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobservable (hidden) states.
- HMM used extensively in tasks processing sequential data, like NLP

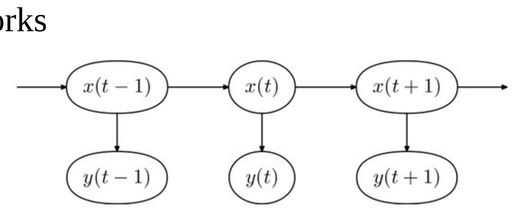




Project Goal

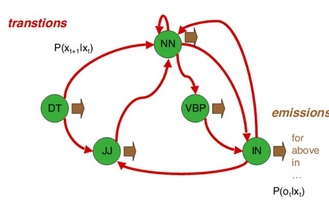
- **To learn how HMM works in real world problem solving**
  - Use HMM to solve POS-tagging problem in NLP
  - Learn structure of HMM and how it works

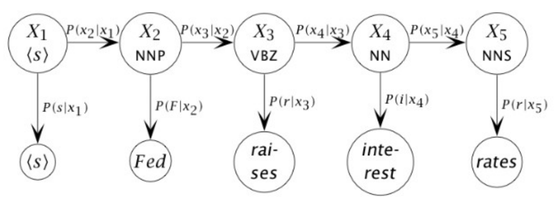


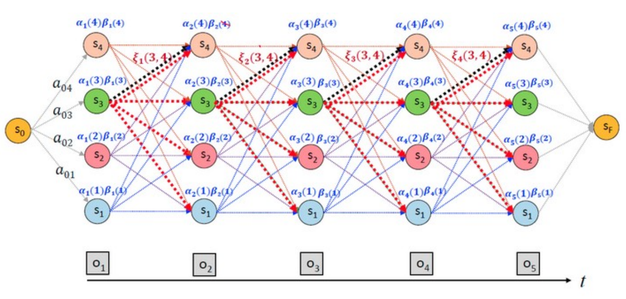


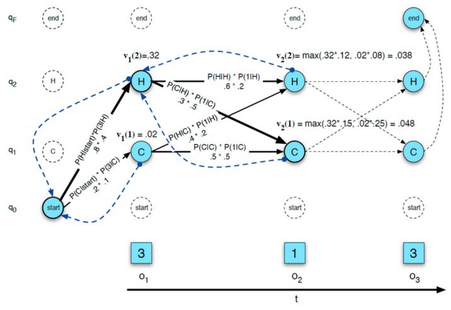
Methodology

- **Use Brown Corpus for training and testing**
  - English corpus from Brown University used extensively in NLP
  - 57,340 sentences with pre-tagged POS in 500 files
- **Extract statistical data to determine prior probability**
  - Calculate start/transition/emission probability by counting POSes and words
- **Training HMM using EM algorithm**
  - Combine forward probability and backward probability
  - Iterate until there is no meaningful difference on log-likelihood before and after training
- **POS tagging using Viterbi algorithm**
  - Calculate Viterbi probability by dynamic programming method
  - Backtrace hidden states from HMM for decoding most likely sequence of states



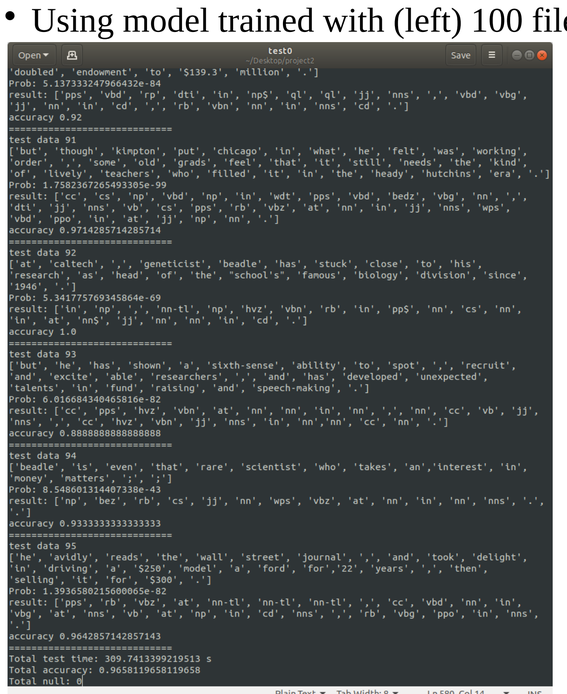


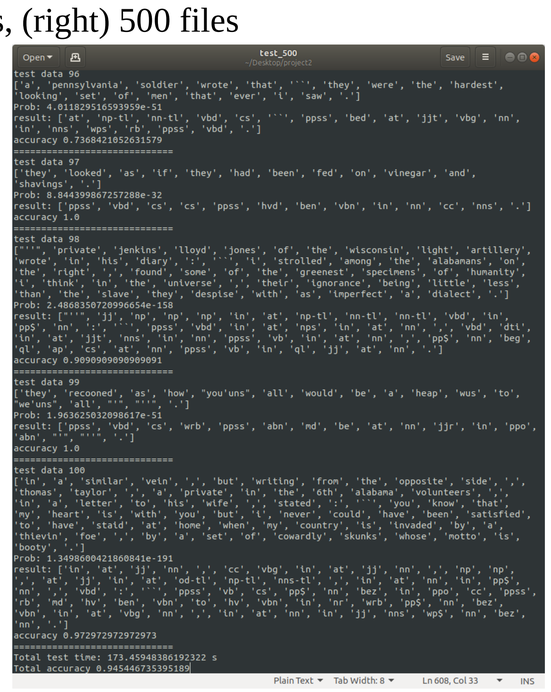




Results & Analysis

- **Do training & testing**
- **Training time is much longer than expected (above a day)**
  - So, I used prior probability on model instead of running EM algorithm
  - Only used 100 among 500 files for training, about 11,000 sentences total
  - The training taked about 2.5 hours, the model consists of total 289 states, 20272 symbols(words).
- **Do POS tagging on test dataset**
  - Test dataset consisted of 703 sentences on 5 files which was not included in training dataset
  - It taked about 5 minutes to test one file
  - Result on about a half of sentences were nearly 90%~100% accuracy, on average.
  - But decoding on the other half of sentences did not work, with probability 0.0.
  - Why?
- **Disclosing why it does not work on some sentences**
  - Tried a test on a file in training dataset (ca40) to find a clue
  - There was no sentences with probability 0.0
  - It was not because floating point precision problem
  - It was because not-working sentences used words that did not appeared in training dataset
- **Analysis for accuracy drop**
  - Each decoding accuracy of tests on 4 files is about 95% on average
  - But one file shows about 81% accuracy on average. Why?
  - It turned out that multi-word proper nouns drops the accuracy
  - Careful concern of upper/lowercase of letter may resolve the problem
- **Below are results of POS tagging**
  - Using model trained with (left) 100 files, (right) 500 files





Lessons learned

- **Learned about how HMM works, in detail**
  - By solving real world problem (POS tagging)
- **Training Takes a lot of time**
  - Maybe it would have been much better to start early
- **It feels great to see POS tagging works!**