Assignment 2

Spring 2019
BBM 497: Introduction to Natural Language Processing Lab.

Burak Emre Özer
burakemreozer@gmail.com

April 12, 2019

1 Introduction

Part-of-speech (POS) tagging is the task of assigning part of speech tags to words in a text. Here we will describe the implementation of a POS-tagger based on Hidden Markov Models and Viterbi algorithm in this assignment. HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states. The Viterbi algorithmis used to compute the most probable path (as well as its probability) of state changing.

2 Task 1: Build a Bigram Hidden Markov Model (HMM)

An HMM is a statistical model of a sequence. It consists of a library of symbols making up the sequence, and a set of states that an element of the sequence might occupy.

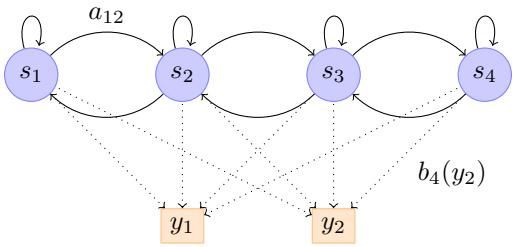


Figure 1: An HMM with 4 states which can emit 2 discrete symbols y_1 or y_2. a_{ij} is the probability to transition from state s_i to state s_j. $b_j(y_k)$ is the probability to emit symbol y_k in state s_j. In this particular HMM, states can only reach themselves or the adjacent state.

Each state has a set of weighted *transition probabilities*: the probability of moving to a different state. A transition probability depends solely upon the previous state; states prior to the previous state have no effect on transition probabilities. An HMM also has a set of *emission probabilities*: the probability of producing a particular element of the sequence

$$P\left(w_n|w_{n-1}\right) = \frac{\#\left(w_{n-1}w_n\right)}{\sum_w \#\left(w_{n-1}w\right)} = \frac{\#\left(w_{n-1}w_n\right)}{\#w_{n-1}} \tag{1}$$

Using the maximum-likelihood estimate can cause problems because unseen transitions are assigned zero-probability. If we also allow zero-probabilities in the symbol emission probabilities we can encounter sentences that have no tag-sequences with non-zero probability. So we will apply different smoothing techniques.

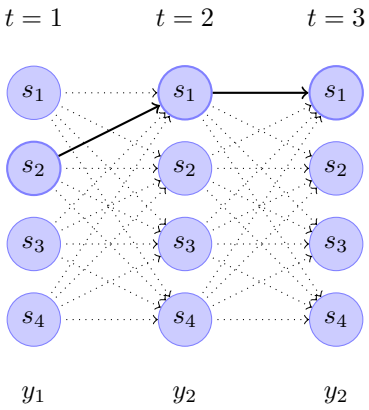


Figure 2: Trellis of the observation sequence y_1,y_2,y_2 for the above HMM. The thick arrows indicate the most probable transitions. As an example, the transition between state s_1 at time t=2 and state s_4 at time t=3 has probability $\alpha_2(1)a_{14}b_4(y_2)$, where $\alpha_t(i)$ is the probability to be in state s_i at time t.

2.1 Unknown Words

If a word is not in the training text then the probability will be zero, zero-probabilities like this will also result in that all possible state-sequences for an observation sequence containing an unknown word will have probability 0 and therefore we cannot choose between them.

2.1.1 Additive/Laplace Smoothing

One way of solving this problem is using Additive/Laplace smoothing.

$$P\left(w_i|w_{i-k}^{i-1}\right) = \frac{\text{count}\left(w_{i-k}^{i}\right) + \alpha}{\text{count}\left(w_{i-k}^{i-1}\right) + \alpha V} \tag{2}$$

Additive smoothing is a type of shrinkage estimator, as the resulting estimate will be between the empirical estimate xi / N, and the uniform probability 1/d. Using Laplace's rule of succession, some authors have argued that should be 1 (in which case the term add-one smoothing is also used), though in practice a smaller value is typically chosen.

2.1.2 Good-Turing Smoothing

Good–Turing frequency estimation is a statistical technique for estimating the probability of encountering an object of a hitherto unseen species, given a set of past observations of objects from different species.

Adjust actual counts c to expected counts c with formula

$$c^* = (c+1)\frac{N_{c+1}}{N_c} \tag{3}$$

-N_c number of n-grams that occur exactly c times in corpus
$-N_0$ total number of unseen n-grams

3 Task 2: Viterbi Algorithm

Since several paths through an HMM may produce the same sequence, paths are ranked by likelihood, by multiplying all of the probabilities together and taking the logarithm of the result. An algorithm known as the Viterbi algorithm (Forney, 1973) provides an optimal state sequence for many purposes.

Algorithm 1 Viterbi Algorithm

procedure VITERBI(Q, A, X, B, q_0)
  for $q \in Q$ **do**
    $P(q, 1) = a(q_0, q)\ b(q, x_1)$
    $\text{Back}(q, 1) = q_0$
  end for
  for t from 2 to T **do**
    for $q \in Q$ **do**
      $P(q, t) = \max_{q' \in Q} P(q', t-1)\ a(q', q)\ b(q, x_t)$
      $\text{Back}(q, t) = \arg\max_{q' \in Q} P(q', t-1) a(q', q)$
    end for
  end for
  $\hat{Z}(T) = \arg\max_{q' \in Q} P(q', T)$
  return Backtrace path following backpointers from the most probable state
end procedure

This implementation of Viterbi was sourced from Ron Artstein, CSCI 544 – Applied Natural Language Processing, Spring 2018, Written Homework 2.

Viterbi algorithm that will assign the most probable PoS tag sequence $t_1 \cdots t_m$ for a given a word sequence $w_1 \cdots w_m$ as follows:

$$\underset{t_1 \cdots t_m}{\text{argmax}} p\left(t_1 \cdots t_m | w_1 \cdots w_m\right) \tag{4}$$

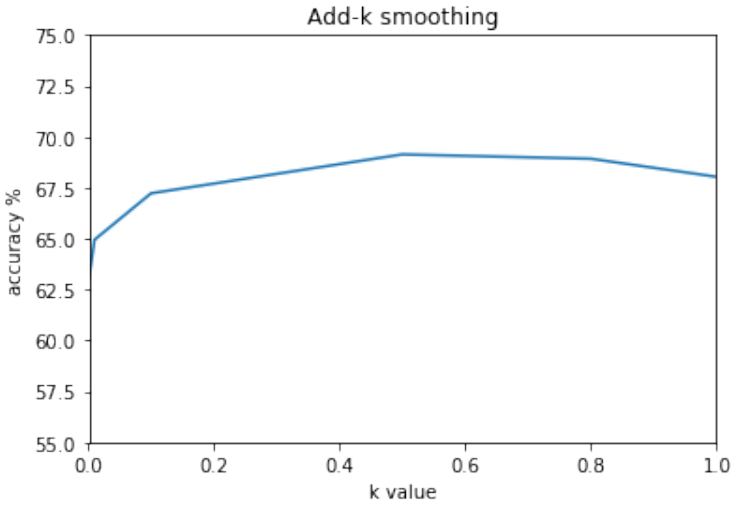
4 Task 3: Evaluation

We will compute the accuracy of the PoS tagger. The accuracy is the ratio of the correctly assigned tags to the total number of words in the test data:

$$A(W) = \frac{\#\text{ of } -\text{correct found tags}}{\#\text{ of total-words}} \tag{5}$$

Additive Smoothing Results

First we applied Additive Smoothing to the emission possibilities. The success rate varied from 60% to 70% according to the value between 0 and 1. The highest success percentage was determined for $\mathbf{k = 0.5}$ with **69.13%**.



Good-Turing Smoothing Results

Good-Turing Smoothing was applied during the test for words that were not included in the training set. The number of unknown words required for this approach is calculated as follows. The unknown word in each incoming test sentence was thrown into a set and its number was constantly updated. Thus, we have converged to the original formula without having to know the size of the dictionary beforehand.

This was the approach with the highest percentage of success in the assignment.

# of Correct Found Tags: 15580

# of Total Words: 18154

Accuracy: 85.82 %

Other approaches and Future works

We have implemented a HMM POS-tagger and looked as different strategies for dealing with unknown words. The smoothing approaches to the transition possibilities did not change the percentage of success but future work for the development of the model is to apply **Kneser-Ney Smoothing** to emission possibilities.

5 Links

https://cl.lingfil.uu.se/~nivre/statmet/haulrich.pdf

https://stats.stackexchange.com/questions/114863/in-kneser-ney-smoothing-how-are-unseen-words-handled

https://jarrodkahn.com/posts/49/viterbi-algorithm/

https://github.com/go2chayan/Kneser_Ney_Ngram

http://homepages.inf.ed.ac.uk/sgwater/teaching/lsa2015/lectures/day3-nup.pdf