IITP-03 Assignment 1: POS Tagging

Soyeon Kim
soyeonk2@andrew.cmu.edu

March 9, 2020

1 Task 1: Generating Learning Curves

I increase the ratio of training data from 0.01 to 1.

1.1 Subtask 1: The Curve

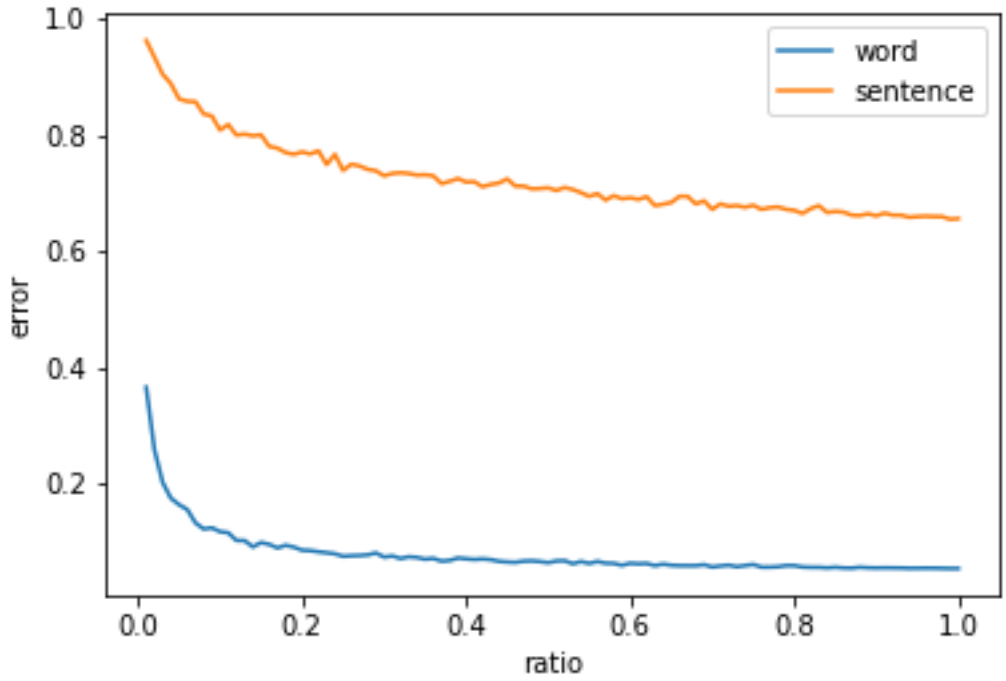

Figure 1: Learning curve

	Word	Sentence
Error rate	0.05409178	0.65588235
Error number	2170 (out of 40117)	1115 (out of 1700)

Table 1: Error with ratio 1

1.2 Subtask 2: Analysis

1. How does the size of the dataset impact the performance of the model/system?

The performance of the larger size of the dataset is much better than the performance of the smaller size of the dataset. However, it is not a linear scale. From ratio 1 to 10, it seems to decrease exponentially while ratio 10 to 100 decreases only a small portion.

2. How do you think the learning curve will change for datasets of different languages? There is no single 'correct' answer to this question, but it has to make sense!

   It is expected that similar learning curves will be generated for different languages. It is general that the larger dataset makes the training accuracy better. If there is a difference, it will be a converged error between the languages. It is because different languages have different tagging system is linguistic model, so I assume that the converged error can be different.

2 Task 2: Building a Better System

2.1 Subtask 1: Your own improved HMM tagger

There are two approaches for improved HMM tagger, one is Trigram and Backoff, the other is Traigram and Smoothing. The difference of these things are detailed as follows:

- Trigram and Backoff

  $$p(t_i|t_{i-1}t_{i-2})$$

  If $p(t_i|t_{i-1}t_{i-2}) = 0$, use

  $$p(t_i|t_{t-1})$$

- Trigram and Smoothing

  $$p(t_i|t_{i-1}t_{i-2}) = \lambda_1\hat{p}(t_i|t_{i-1}t_{i-2}) + \lambda_2\hat{p}(t_i|t_{t-1}) + \lambda_3\hat{p}(t_i)$$

Model	Word Error rate	Word Error number	Sentence Error rate	Sentence Error number
Bigram	0.05409178	2170	0.65588235	1115
Tri + Backoff	**0.04935563**	**1980**	**0.62352941**	**1060**
Tri + Smoothing	0.06967121	2795	0.72058823	1225

Table 2: Model accuracy comparison. Note that the the ratio of training set is all 1. $\lambda_1 = 0.5, \lambda_2 = 0.5, \lambda_3 = 0.0$ for Trigram and smoothing.

2.2 Subtask 2: Analysis

1. What modifications did you make?

   (a) Training HMM: Change the HMM file as follows:
   trans init IN 0.12678248644306087 → tri_trans init init IN 0.12678248644306087

   (b) Tagger:
   I basically follow the algorithm appeared in figure 2.
   If the probability of trigram transition is zero, (i.e., there are no training example) I use two approaches: (1) Backoff and (2) Smoothing.

2. How much improvement did your new model deliver on section 22 (ptb.22.txt)?

   There are no improvement when I use trigram and smoothing, but there are small improvement in trigram and backoff. The number of word error dropped by 190 and the number of sentence error dropped by 55.

   However, there are some trade-off between model: improvement and time efficiency. If I use bigram to viterbi algorithm, run time is only 0.918 secs but for trigram, it takes more than 60 secs.

Input: a sentence $x_1 \dots x_n$, parameters $q(s|u,v)$ and $e(x|s)$.

Initialization: Set $\pi(0, *, *) = 1$

Definition: $\mathcal{S}_{-1} = \mathcal{S}_0 = \{*\}$, $\mathcal{S}_k = \mathcal{S}$ for $k \in \{1 \dots n\}$

Algorithm:

- For $k = 1 \dots n$,
  - For $u \in \mathcal{S}_{k-1}$, $v \in \mathcal{S}_k$,

$$\pi(k,u,v) = \max_{w \in \mathcal{S}_{k-2}} (\pi(k-1,w,u) \times q(v|w,u) \times e(x_k|v))$$

$$bp(k,u,v) = \arg\max_{w \in \mathcal{S}_{k-2}} (\pi(k-1,w,u) \times q(v|w,u) \times e(x_k|v))$$

- Set $(y_{n-1}, y_n) = \arg\max_{(u,v)} (\pi(n,u,v) \times q(\text{STOP}|u,v))$
- For $k = (n-2) \dots 1$, $y_k = bp(k+2, y_{k+1}, y_{k+2})$
- **Return** the tag sequence $y_1 \dots y_n$

Figure 2: algorithm

3. Why do you think that these modifications improved the accuracy of this labelling task?

   Language is not a markov model. It depends all previous systems, so it is natural that more window makes better accuracy of the output.

3 Reference

[1] https://ahgohlearns.wordpress.com/2013/04/29/a-viterbi-trigram-hmm-tagger/