CS563: Natural Language Processing

Indian Institute of Technology Patna

Assignment 1

POS Tagging using HMM

Group Members:

- Muhammed Sinan C K (1901CS38)
- Vaishakh Sreekanth Menon (1901CS68)
- Varsha Tumburu (1901CS69)

POS tagging with Hidden Markov Model

HMM (Hidden Markov Model) is a stochastic technique for POS tagging. Hidden Markov models are known for their applications to reinforcement learning and temporal pattern recognition such as speech, handwriting, gesture recognition, musical score following, partial discharges, and bioinformatics.

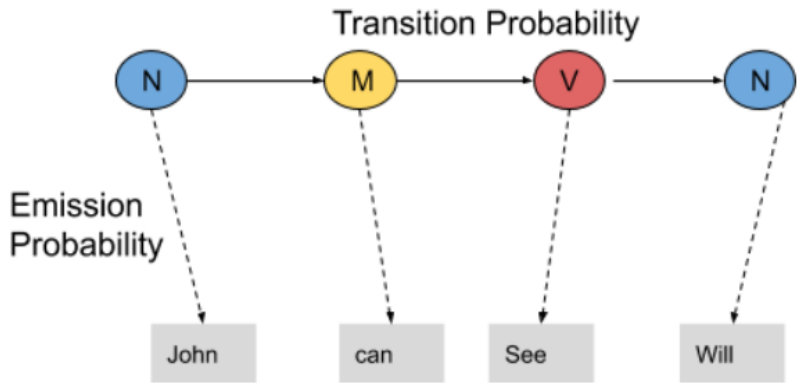

Figure 1

In figure 1 we can see there is emission probability and transition probability. In transition probability we will get the probability of transition from one part of speech to another. For example we can find the transition probability from noun to verb. Aim of this assignment is to find the best POS tagging which has higher probability to occur . Where probability is calculated by using emission probability and transition probability.

Example:

Let's take an example "will can spot mary "

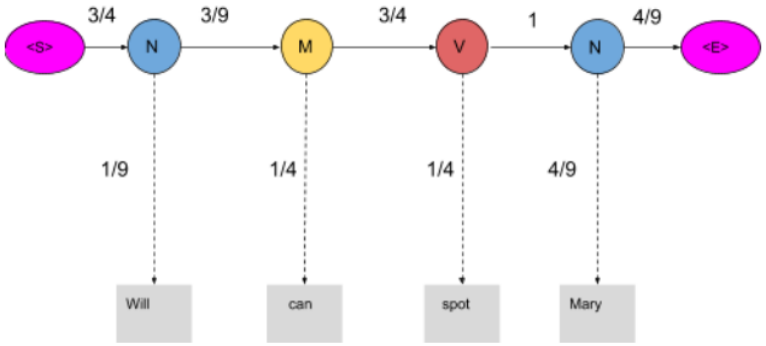

Figure 2

Here in figure 2 probability for this POS tagging is

3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164

We multiplied transition and emission probability directly because we assumed bigram dependency only.

Viterbi algorithm

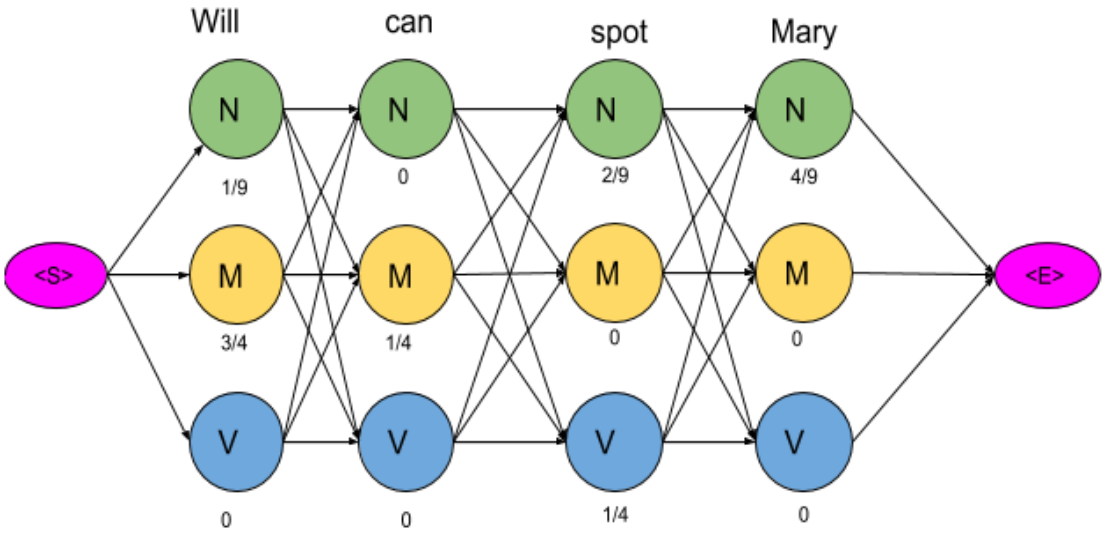

Figure 3

Figure 3 representing the entire graph structure representing all transition probabilities between part of speech and final probability we have to multiply with emission probability which will give probability of given word with respect to part of speech.But by bruteforce if we calculate probabilities for all path and choosing best one will lead to higher time complexity. Because if there is k part of speeches and n is the sentence length then time complexity will be o(m^k) which is exponential.

So to solve this there is viterbi algorithm which will use dynamic programming approach and it's time complexity O($k^2 n$).
In this approach let's create a 2d matrix of size (n,k) called dp where n is the length of the sentence and k is the number of part of speech.
Here dp[index][tag] means what is the best probability for getting word in the index has part of speech has a tag .

For find the best probability considering previous words in the sentence and it's pos tagging

Equation

```
Iterate through all prev_tag in index-1:
     dp[index][tag]= max(dp[index-1][prev_tag] x
transition_probability[prev_tag to tag] x
emission_probability(current word | tag), dp[index][tag])
```

After calculating all probabilities of 2d matrix dp , we can backtrack from end of the sentence to beginning of the sentence to find the best part of the speech tag sequence of a given sentence which has higher probability.

Overall HMM Model Statistics

Approach	Average Accuracy	Improvement
36 POS Tags Scheme	79%	8.86%
4 POS Tags Scheme	86%	

Results using 36 POS Tags

Sample Testcase:

Most predicted tags match with the ground truth except for a few cases, which back the 79% accuracy that we receive on the model. Notice that we use <RARE> tags wherever words repeat less than 4 times so that the model is not underfitted based on rarely occurring words. This allows the model to predict better.

```
"sentence": "But he has not said before that the country wants half the debt <RARE>",
"ground_truth": [
    "CC",
    "PRP",
    "VBZ",
    "RB",
    "VBD",
    "IN",
    "IN",
    "DT",
    "NN",
    "VBZ",
    "PDT",
    "DT",
    "NN",
    "VBN"
],
"prediction": [
    "CC",
    "PRP",
    "VBZ",
    "RB",
    "VBD",
    "IN",
    "IN",
    "DT",
    "NN",
    "VBZ",
    "NN",
    "DT",
    "NN",
    "NN"
```


Classification report for 36 POS tags

	precision	recall	f1-score	support
#	0.00	0.00	0.00	1
''	0.00	0.00	0.00	1
-LRB-	1.00	0.82	0.90	22
-RRB-	1.00	0.82	0.90	22
:	1.00	0.90	0.95	52
<START>	0.00	0.00	0.00	0
CC	1.00	0.96	0.98	461
CD	0.78	0.64	0.71	717
DT	0.99	0.97	0.98	1624
EX	0.89	0.80	0.84	10
IN	0.98	0.95	0.96	1957
JJ	0.65	0.69	0.67	1140
JJR	0.76	0.56	0.64	90
JJS	0.92	0.67	0.77	36
LS	0.00	0.00	0.00	2
MD	0.99	0.94	0.97	173
NN	0.70	0.76	0.73	2550
NNP	0.62	0.85	0.72	1931
NNPS	0.62	0.20	0.30	41
NNS	0.81	0.51	0.63	1183
PDT	0.67	0.50	0.57	4
PRP	0.99	0.94	0.96	288
PRP$	0.99	0.90	0.95	145
RB	0.80	0.52	0.63	482
RBR	0.44	0.22	0.29	32
RBS	0.83	0.62	0.71	8
RP	0.58	0.76	0.66	42
TO	0.99	0.98	0.99	426
UH	0.00	0.00	0.00	1
VB	0.89	0.80	0.84	502
VBD	0.85	0.77	0.81	614
VBG	0.57	0.41	0.48	283
VBN	0.61	0.60	0.61	414
VBP	0.91	0.76	0.83	234
VBZ	0.84	0.81	0.82	383
WDT	0.92	0.90	0.91	93
WP	0.97	0.81	0.89	43
WP$	1.00	1.00	1.00	2
WRB	1.00	0.88	0.94	33
accuracy			0.79	16042
macro avg	0.73	0.65	0.68	16042
weighted avg	0.81	0.79	0.79	16042

Results using 4 POS Tags

Sample Testcase:

We can see that there is only one misprediction in this sample testcase. And in this way the 4 POS tagger performs better while predicting than our previous tagger.


```
{
    "sentence": "<RARE> which went down 2 1/2 Tuesday, lost another 1/2 to 50 <RARE>",
    "ground_truth": [
        "N",
        "O",
        "V",
        "O",
        "O",
        "O",
        "N",
        "V",
        "O",
        "O",
        "O",
        "O",
        "O"
    ],
    "prediction": [
        "N",
        "O",
        "V",
        "O",
        "O",
        "O",
        "N",
        "V",
        "O",
        "O",
        "O",
        "O",
        "N"
    ]
```


Classification report for 4 POS Tags

	precision	recall	f1-score	support
A	0.95	0.56	0.71	1851
N	0.75	0.98	0.85	6242
O	0.99	0.92	0.95	5636
V	0.93	0.63	0.75	2526
accuracy			0.86	16255
macro avg	0.90	0.77	0.81	16255
weighted avg	0.88	0.86	0.85	16255

Observation:

The overall POS tagging accuracy for 4 POS tag setting is better than the 36 POS tag setting. There is an improvement of 8.86% just by changing this setting.

The reason for this improvement is that, in case of the 4 tags POS tagging, ambiguities are lesser as compared to 36 tags. Higher the options, higher are the chances of choosing the wrong tag. For the 36 tags setting, more transition probabilities and emission probabilities are calculated. This increases the scope of error as at each step there are more choices to take than in the 4-step setting.

Due to this, the rate of committing error is higher in the 36-tag setting. Also, since number of tags are higher for the 36 tags setting, a larger dataset will be required to correctly capture the right transition and emission probability distributions for it to perform as good as the 4-tag setting.