***Computation of emission probability***

Hidden

States HS1 → HS2 → HS3 → HS4 →HS5 → HS6 → HS7 → HS8 → ……………………..HSN

Observed

Sates OS1 → OS2 → OS3 → Os4 →OS5 → OS6 → OS7 → OS8 → ……………………..OSN

where HS(i) is pos tags and OS(i) is given word

emission probability is calculated as P(word/tag) = count(word and tag) / count(tag)

we will do these for all words present in training data and letter used it for testing that is for observed words

so we will have **emission\_matrix** as

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | tag1 | tag2 | tag3 | tag4 | . | . | . | tagN |
| word1 | count(word1 and tag1) / count(tag1) | count(word1 and tag2) / count(tag2) | count(word1 and tag3) / count(tag3) | count(word1 and tag4) / count(tag4) | . | . | . | count(word1 and tagN) / count(tagN) |
| word2 | count(word2 and tag1) / count(tag1) | count(word2 and tag2) / count(tag2) | count(word2 and tag3) / count(tag3) | count(word2 and tag4) / count(tag4) | . | . | . | count(word2 and tagN) / count(tagN) |
| word3 | . | . | . | . | . | . | . | . |
| word4 | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| wordN | count(wordN and tag1) / count(tag1) | count(wordN and tag2) / count(tag2) | count(wordN and tag3) / count(tag3) | count(wordN and tag4) / count(tag4) | . | . | . | count(wordN and tagN) / count(tagN) |

wordi and tagi belongs to training data

***Computation of transition probability***

Here we will generate the transition probability matrix it is tag x tag matrix and each value represents the p(tag\_i/tag\_j)=count(tag\_i and tag\_j) / count(tag\_j)

**transition\_matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tag1 | tag2 | . | . | tagN |
| tag1 | count(tag1 and tag2) / count(tag1) | count(tag1 and tag2) / count(tag2) | . | . | count(tag1 and tagN) / count(tagN) |
| tag2 |  |  |  |  |  |
| . |  |  |  |  |  |
| . |  |  |  |  |  |
| tagN | count(tagN and tag1) / count(tag1) | count(tagN and tag2) / count(tag2) |  |  | count(tagN and tagN) / count(tagN) |

here tagi belongs to training data

it follows markov property that is t th state depends on t-1 th state

***computation of initial probabilities of hidden state(Pi)***

it is calculated from transition\_matrix

**Repeated Matrix multiplication**

pi = lim transition\_matrix ^ n

n→ inf

but here we had take n as 10^2

but we can use **Monto carlo** or **left eigen vector** to compute the **pi**

***Accuracy:***

Algorithm Accuracy: 90.04975124378109