Hidden Markov Model(HMM)
Il modello Hidden Markov è un modello probabilistico utilizzato per esplicitare o derivare le caratteristiche probabilistiche di qualsiasi processo casuale. Fondamentalmente afferma che un evento osservato non corrisponderà al suo stato graduale ma correlato a un insieme di distribuzioni di probabilità. Supponiamo che un sistema che viene modellato sia considerato una catena di Markov e nel processo ci siano alcuni stati nascosti. In tal caso, possiamo dire che gli stati nascosti sono un processo che dipende dal processo/catena principale di Markov. L'obiettivo principale delle HMM è effettuare il learning di una catena di Markov osservandone gli stati nascosti. Considerando un processo di Markov X con stati nascosti Y qui l'HMM consolida che per ogni timestamp la distribuzione di probabilità di Y non deve dipendere dalla storia di X secondo quel tempo.

HMM con un esempio
Si considerano due amici A e B. Ora A completa i suoi lavori di vita quotidiana in base alle condizioni meteorologiche. Le tre attività principali completate da A sono: fare jogging, andare in ufficio e pulire la sua residenza. Quello che A sta facendo oggi dipende dal fatto che A dica a B e B non ha informazioni adeguate sul tempo, ma B può presumere le condizioni meteorologiche secondo il lavoro di A. B crede che il tempo operi come una catena di Markov discreta, in cui nella catena ci sono solo due stati se il tempo è piovoso o c'è il sole. Le condizioni meteorologiche non possono essere osservate da B, qui le condizioni meteorologiche sono nascoste a B. Ogni giorno, c'è una certa possibilità che A esegua un'attività dall'insieme delle seguenti attività {“jog”, “work”,” clean”}, che dipendono dal tempo. Dal momento che A dice a B quello che ha fatto, quelle sono le osservazioni. L'intero sistema è quello di un modello Markov nascosto (HMM).

Qui possiamo dire che il parametro di HMM è noto a B perché ha informazioni generali sul tempo e sa anche cosa ama fare in media A.
Quindi si considera, ad esempio, un giorno in cui A ha chiamato B e gli ha detto che ha pulito la sua residenza. In quello scenario, B crederà che ci siano più possibilità di una giornata piovosa e possiamo dire che la convinzione che B ha, è la probabilità di inizio della HMM diciamo che è come di seguito.

Gli stati e le osservazioni sono:

```
states = ( ‘rainy , ‘sunny’)
osservazioni = (‘walk’, ‘shop’, ‘clean’)
start_proba = {‘rainy’ : 0.6 , ‘sunny’ : 0.4}
```

Ora la distribuzione della probabilità ha il peso maggiore nel giorno di pioggia negli USA, quindi possiamo dire che ci saranno più possibilità che un giorno sia di nuovo piovoso e le probabilità per gli stati meteorologici del giorno successivo sono le seguenti:

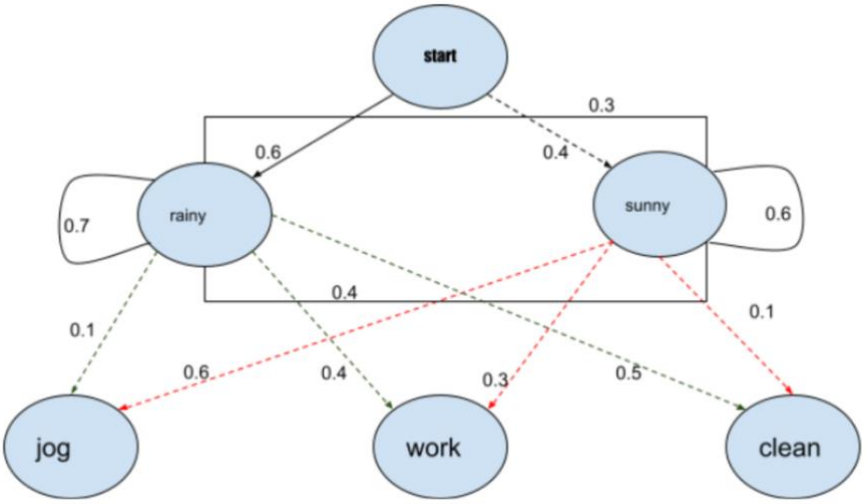
```
transition_proba = {

        ‘rainy’ : {‘rainy’ : 0.7, ‘sunny’ : 0.3},
        ‘sunny’ : {‘rainy’ : 0.4, ‘sunny’ : 0.6}

}
```

Da quanto sopra si può dire che i cambiamenti nella probabilità per un giorno sono probabilità di transizione e in accordo alla probabilità di transizione i risultati emessi per la probabilità di lavoro che A eseguirà sono:

emission_proba = {

‘rainy’ : { ‘jog’ : 0.1 , ‘work’ : 0.4, ‘clean’ : 0.5**}**
‘sunny’ : { ‘jog’ : 0.6, ‘work’ : 0.3, ‘clean’ : 0.1}
}

Queste probabilità possono essere considerate come le probabilità di emissione. Usando queste probabilità B può prevedere gli stati nel tempo o usando le probabilità di transizione B può predire il lavoro che A si appresta ad eseguire il giorno successivo.
La seguente mostra il processo HMM per la definizione delle probabilità:



Quindi qui dall'intuizione di cui sopra e dall'esempio possiamo capire come possiamo usare questo modello probabilistico per fare una previsione.

HMM in NLP
Le HMM possono essere impiegate in tutti quei campi dove si ha a che fare con dati sequenziali, come, ad esempio, le time series, audio, video o testi.
Un’applicazione particolare in NLP dell HMM è nel Part-of-Speech tagging.
Di seguito si va a vedere come si possono impiegare le HMM per il POS-tagging.

POS-Tagging
Quando si era piccoli si è imparto a scuola che ogni parola in un discorso aveva un proprio significato. Comunemente, in un discorso, ci sono 9 parti, quali:

- Sostantivo
- Pronome
- Verbo
- Avverbio
- Articolo
- Aggettivo
- Preposizione
- Congiunzione
- Interiezione

Queste devono essere inserite in modo corretto in una frase per far in modo che questa abbia un senso compiuto.

La codifica POS è una parte fondamentale del NLP poiché questa è un'attività in cui si realizza una macchina in grado di comunicare con un'altra macchina o con un essere umano. Di conseguenza, per una macchina diventa fondamentale le varie parti, elencate in precedenza, che compongono un discorso.
Il task di classificare le parole nella loro parte del discorso e fornire le loro labels in accordo alla loro parte del discorso viene definito "*Part of Speech Tagging*" (o "*POS Tagging*"). L'insieme delle labels/tags viene definito "*Tagset*".

Esistono diverse tecniche che possono essere utilizzate per il POS tagging, come ad esempio:

- *Tagging POS basato su regole:* I modelli di POS tagging basati su regole applicano una serie di regole scritte a mano e utilizzano le informazioni contestuali per assegnare i tag POS alle parole. Queste regole sono spesso note come "context frame rules". Una di queste regole potrebbe essere: "Se una parola ambigua/sconosciuta termina con il suffisso 'ing' ed è preceduta da un verbo, allora viene etichettata come verbo".

- *Tagging basato sulla trasformazione*: Gli approcci basati sulle trasformazioni utilizzano un insieme predefinito di regole create a mano e regole indotte automaticamente, generate durante l'addestramento.

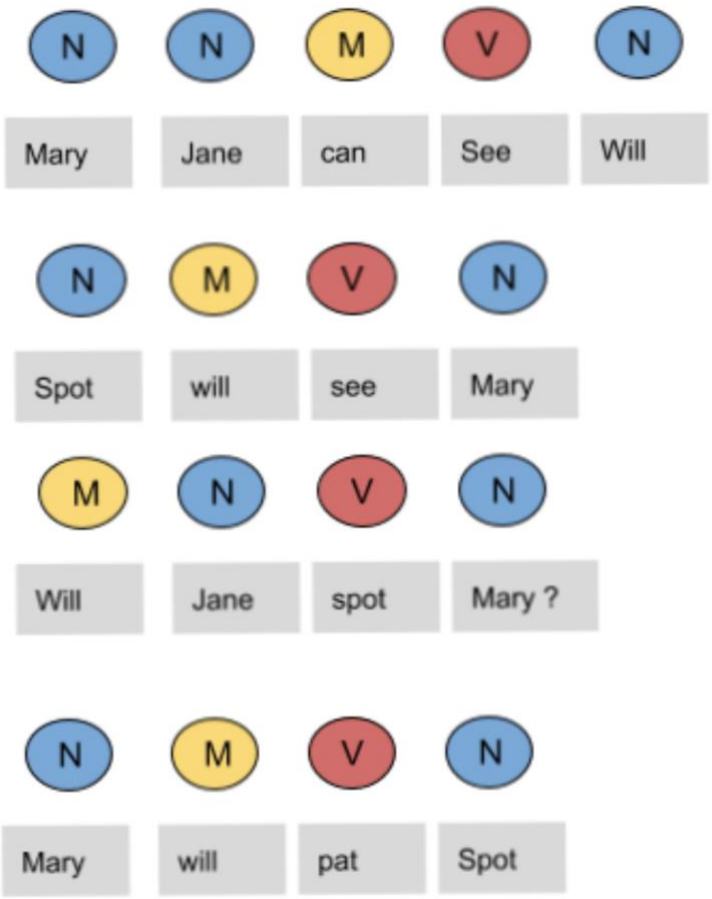
- *Modelli di Deep Learning*: Per l'etichettatura POS sono stati utilizzati diversi modelli di deep learning, come Meta-BiLSTM, che hanno dimostrato un'accuratezza di circa il 97%.

- *Tagging stocastico (probabilistico):* Un approccio stocastico include frequenza, probabilità o statistiche. L'approccio stocastico più semplice individua il tag più frequentemente utilizzato per una parola specifica nei dati di addestramento e utilizza questa informazione per etichettare quella parola nel testo. A volte, però, questo approccio produce sequenze di tag per frasi che non sono accettabili secondo le regole grammaticali di una lingua. Uno di questi approcci consiste nel calcolare le probabilità delle varie sequenze di tag possibili per una frase e assegnare i tag POS della sequenza con la probabilità più alta. I modelli di Markov nascosti (HMM) sono approcci probabilistici per assegnare un tag POS.

POS-Tagging con le HMM
Impiegando le HMM, impieghiamo una tecnica stocastica per il POS tagging, i.e. le HMM ci permettono di selezionare un tag POS accurato per ogni determinata frase.
Sostanzialmente, nel Pos tagging, la probabilità di transizione corrisponde alla probabilità di qualsiasi sequenza, come, ad esempio, quali sono le probabilità che un nome venga dopo un qualsiasi modale o le probabilità che un modale venga dopo un verbo e un verbo dopo un nome. Dovrebbe essere alta in modo da avere una sequenza corretta.

Considerando la frase "A will eat food', allora A è il nome, will è un modale, eat è un verbo e food è un nome, quindi la probabilità per una parola di essere in una particolare classe di una parte del discorso viene definita come *Probabilità di Emissione*.
Il seguente esempio mostra come si possono calcolare due probabilità di emissione per un set di frasi. Si considerano solo 3 Pos Tags che sono nome, verbo e modale.

Nel seguente grafico si evidenzia la caratterizzazione in Pos tag delle frasi.



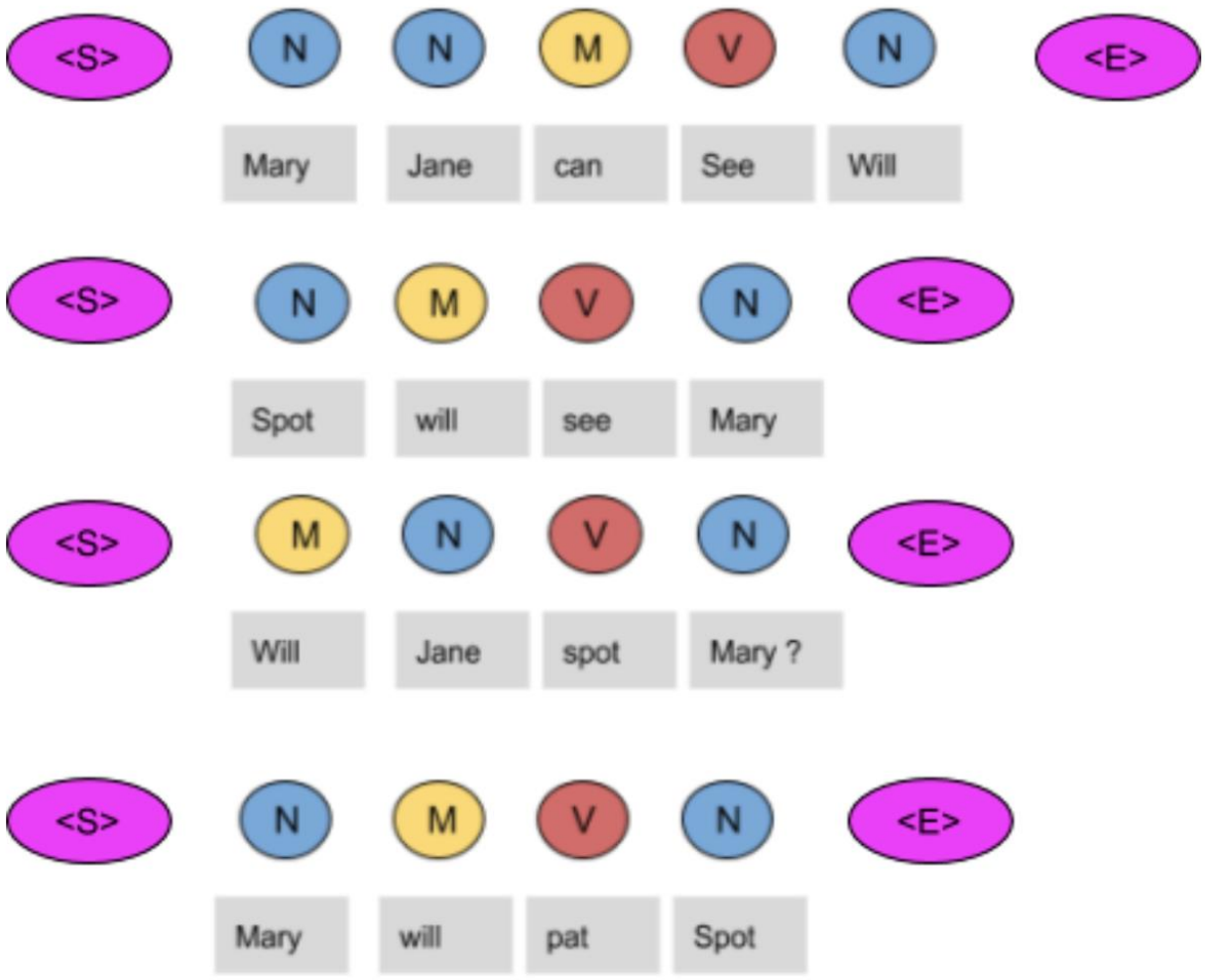
Di seguito si conta quante volte una parola è nome, modale e verbo nelle frasi precedenti.

Parole	**Nome**	**Modale**	**Verbo**
Mary	4	0	0
Jane	2	0	0
Will	1	3	0
Spot	2	0	1
Can	0	1	0
See	0	0	2
Pat	0	0	1

Si divide la comparsa di ogni parola per il numero totale di ogni parte del discorso nel set di frasi. Ad esempio, il nome appare nove volte nelle frasi, quindi ogni termine verrà diviso per nove.

Parole	Nome	Modale	Verbo
Mary	4/9	0	0
Jane	2/9	0	0
Will	1/9	¾	0
Spot	2/9	0	¼
Can	0	1/4	0
See	0	0	2/4
Pat	0	0	1/4

La tabella mostra, per ogni parola, la probabilità di emissione.
Si definiscono ora le probabilità di transizione e, per farlo, si configurano altri due tags <S> e <E>.
<S>, cioè Start, viene inserito all'inizio di ogni frase e <E>, cioè End, alla fine, nel seguente modo:



Ora, poiché la probabilità di transizione risulta essere la probabilità delle sequenze, si può definire una tabella per l'insieme di frasi di cui sopra in base alla sequenza delle parti del discorso.

	Nome	**Modale**	**Verbo**	**Fine**
Inizio	3	1	0	0
Nome	1	3	1	4
Modale	1	0	3	0
Verbo	4	0	0	0

Nella tabella, bisogna verificare la combinazione delle parti del discorso per calcolare la probabilità di transizione. Per esempio, si può vedere che nell'insieme di frasi il modale è apparso 3 volte prima di un verbo e 1 volta prima di un nome. Questo significa che è apparso nell'insieme per 4 volte e che la probabilità che il modale venga prima di un verbo sarà di ¾ e prima di un nome sarà di ¼.
Quindi, eseguendo quest'ultima operazione per ogni entità, si ottiene la seguente tabella:

	Nome	**Modale**	**Verbo**	**Fine**
Inizio	¾	¼	0	0
Nome	1/9	3/9	1/9	4/9
Modale	¼	0	¾	0
Verbo	4/4	0	0	0

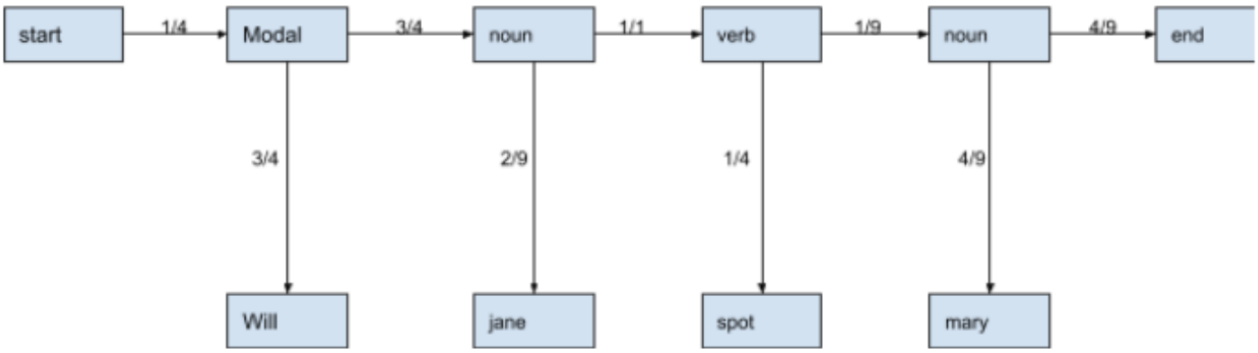
I valori riportati nella tabella sono i rispettivi valori di transizione per un dato insieme di frasi.

A questo punto ci si chiede come fa l'HMM a determinare la sequenza appropriata di tag per una particolare frase a partire dalle tabelle ricavate in precedenza.

Si consideri la frase "Will Jane spot Mary" che viene taggata nel seguente modo:

- Will è modale
- Spot è verbo
- Jane è nome
- Mary è nome

Quindi si calcola la probabilità che questa sequenza sia corretta nel seguente modo:



Nel grafico si hanno le probabilità di emissione delle parole nella frase indicate nelle linee verticali, mentre le linee orizzontali rappresentano tutte le probabilità di transizione.

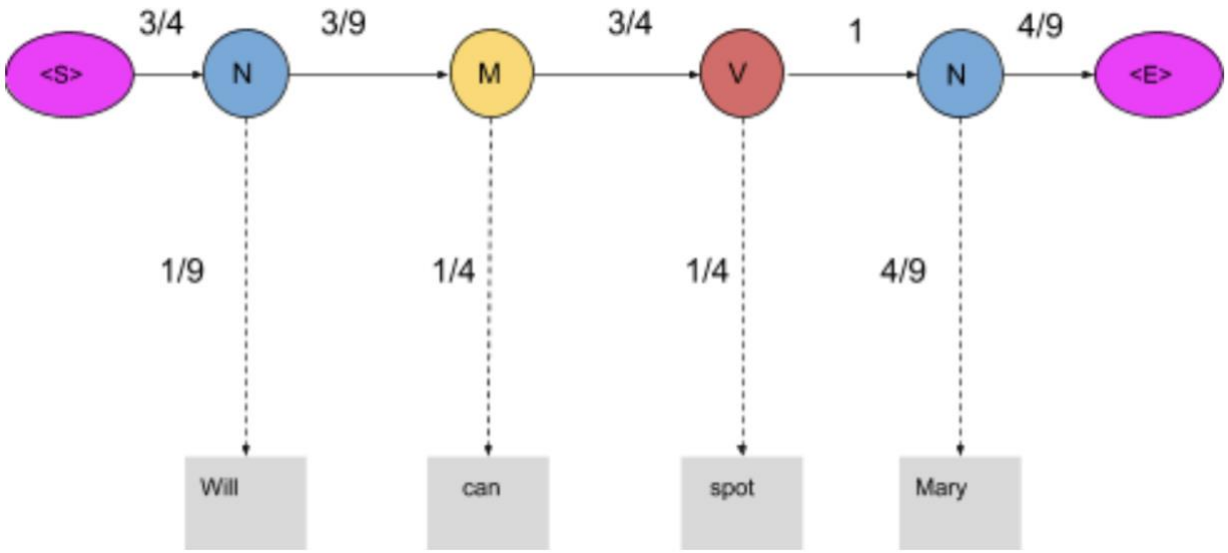
La correttezza del POS tagging viene misurata dal prodotto di tutte queste probabilità. Il prodotto delle probabilità rappresenta la probabilità che la sequenza sia corretta.
Quindi, nell'esempio precedente, si ha la seguente correttezza per il POS tagging.

¼ * ¾ * ¾ * 2/9 * 1/1 * ¼ * 1/9 * 4/9 * 4/9 = 0.0001714678.

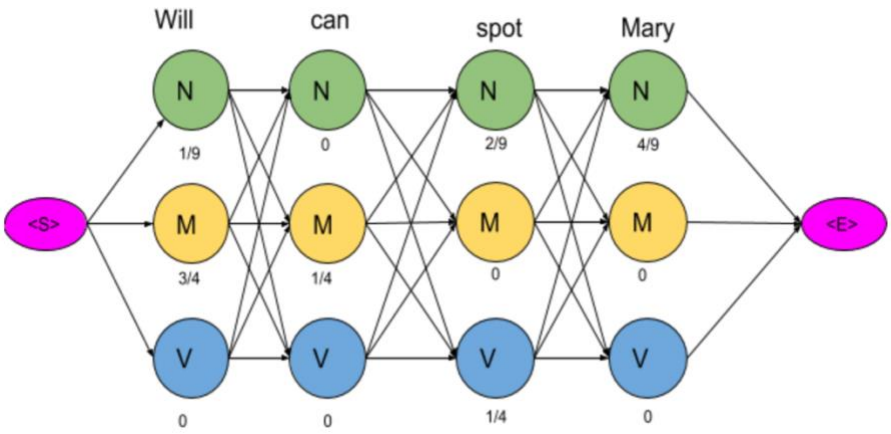
Il risultato ottenuto mostra che il POS tagging eseguito è corretto in quanto il risultato è maggiore di zero. Quando il risultato è zero, il tagging eseguito non sarà correttto.

L'esempio fornito è piccolo: solo 3 tipi di tag POS, con 81 possibili combinazioni diverse di tag. In questo caso, calcolare le probabilità di tutte le 81 combinazioni sembra fattibile. Tuttavia, quando si ha a che fare con un grande insieme di dati il numero di combinazioni aumenta esponenzialmente. Chiaramente, maggiore è il numero di tag POS e maggior accuratezza si avrà, tuttavia si avrà anche una maggior complessità computazionale.

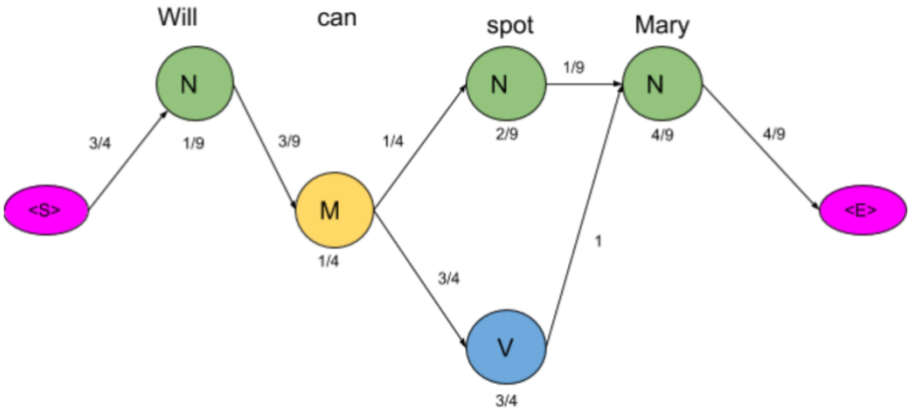
Di seguito si visualizzano le 81 combinazioni come percorsi e, usando le probabilità di transizione ed emissione, si contrassegna ogni vertice e bordo.
Si considera la frase: "Will can spot Mary", con le seguenti probabilità di transizione:



E, verificata anche la correttezza come: 3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164, si avrà:



Il passo successivo è di eliminare tutti i vertici e archi con probabilità = 0 e tutti i vertici che non conducono all'endpoint <E>.



Ora ci sono solo due percorsi che portano alla fine e si va a calcolare la probabilità associata a ciascun percorso.

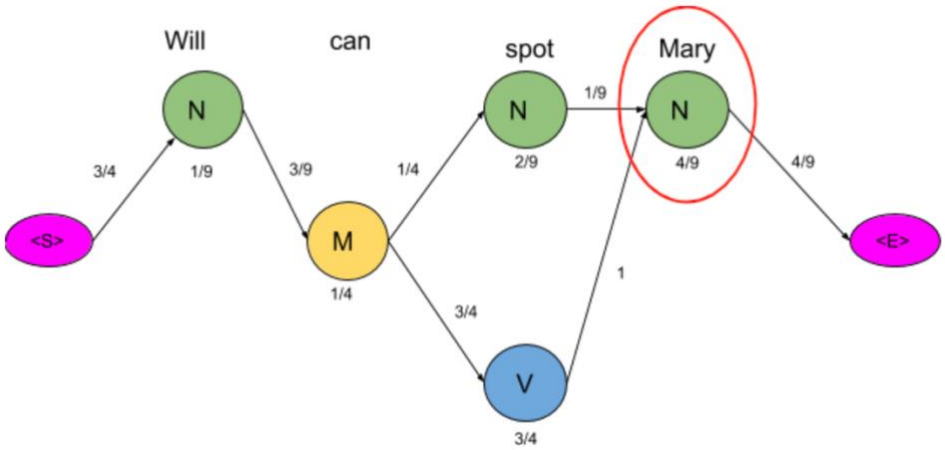
<S>→N→M→N→N→<E> = 3/4*1/9*3/9*1/4*1/4*2/9*1/9*4/9*4/9=0.00000846754
<S>→N→M→N→V→<E> = 3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164

Chiaramente, la probabilità della seconda sequenza è molto più alta e, quindi, l'HMM etichetterà ogni parola della frase in base a questa sequenza.

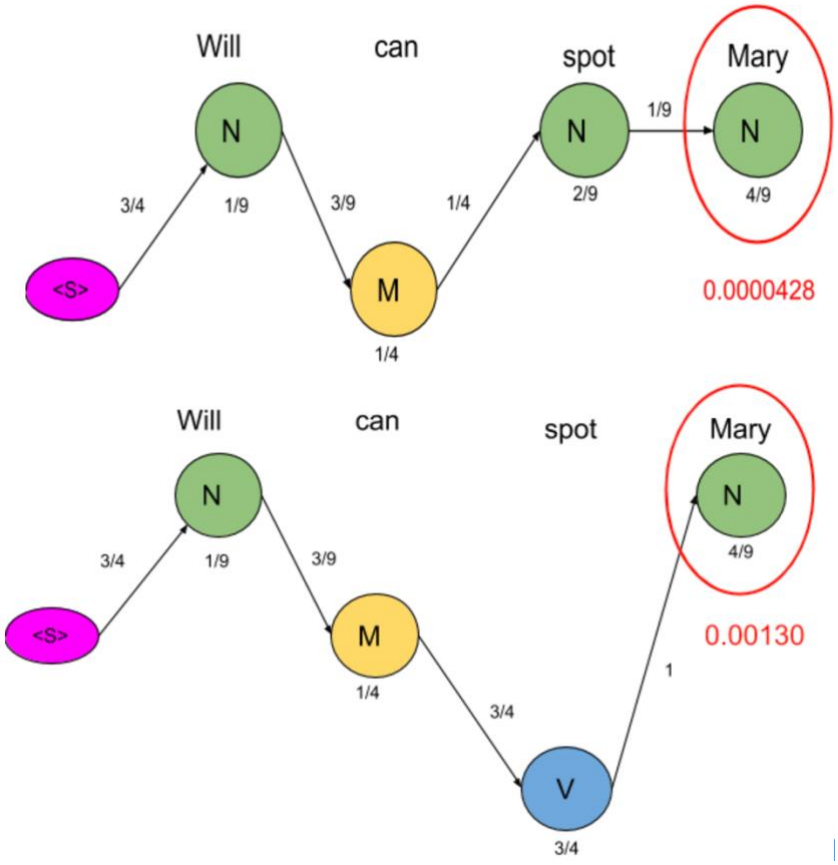
Ottimizzazione delle HMM con l'algortimo di Viterbi

L'algoritmo di Viterbi è un algoritmo di programmazione dinamica per trovare la sequenza più probabile di stati nascosti - il percorso di Viterbi - che risulta da una sequenza di eventi osservati, soprattutto nel contesto dei modelli di Markov nascosti (HMM).

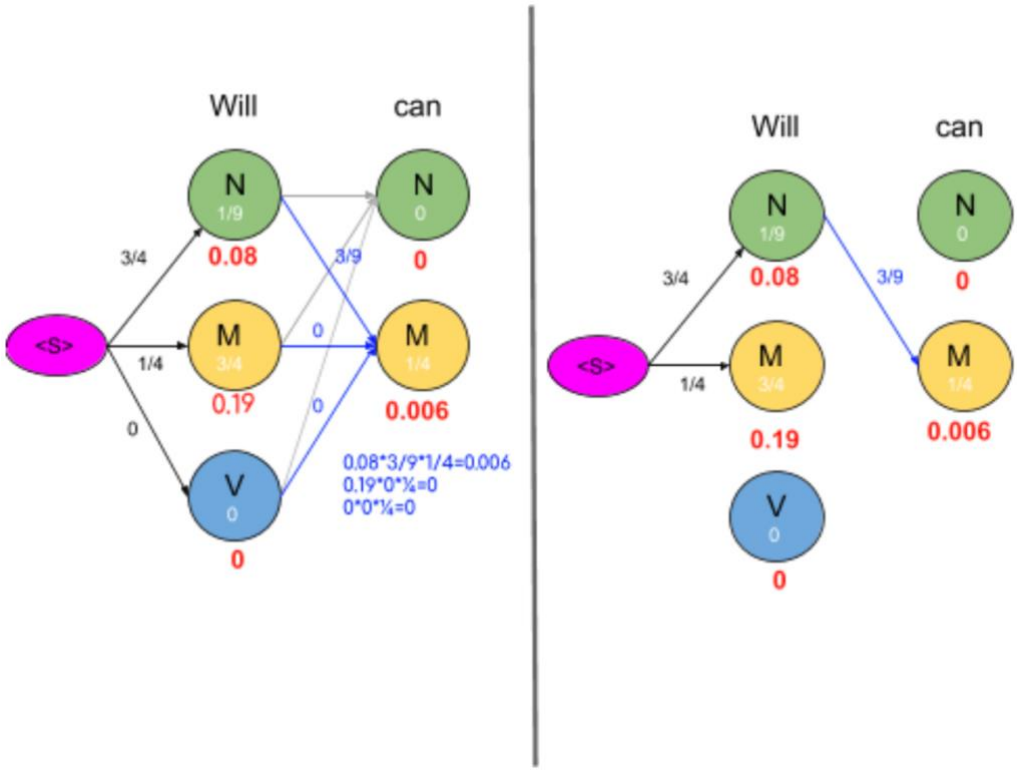
Nella sezione precedente si è ottimizzato l'HMM e ridotto i calcoli da 81 a soli due. Tuttavia, è possibile ottimizzare ulteriormente l'HMM usando l'algoritmo di Viterbi.
A titolo esplicativo si usa lo stesso esempio di prima e si mette in atto l'algoritmo di Viterbi.



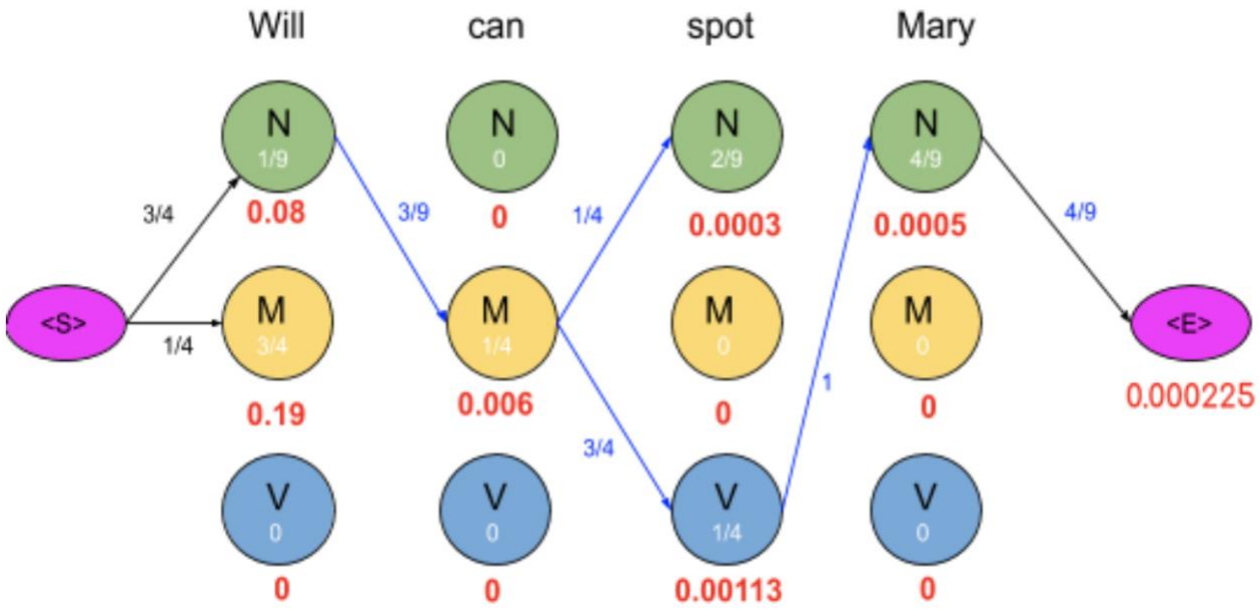
Si consideri il vertice cerchiato nell'esempio precedente.
Esistono due percorsi che portano a questo vertice, come mostrato di seguito, insieme alle probabilità dei due semi-percorsi.



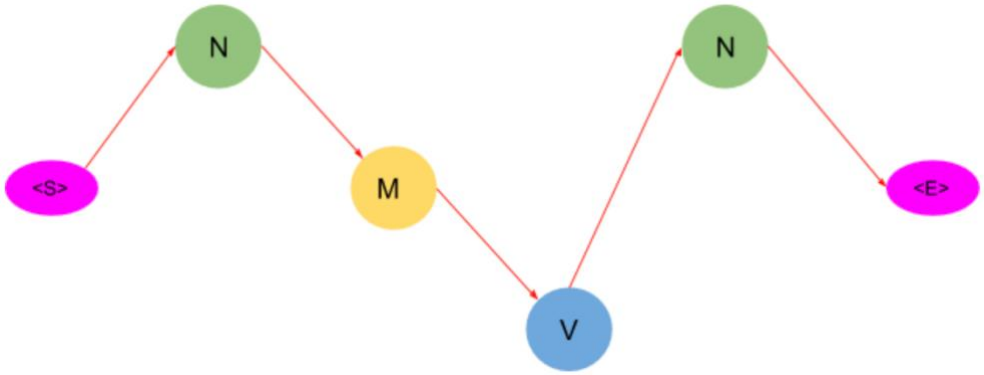
Si è, quindi, interessati al semi-percorso con la probabilità più bassa. La stessa procedura viene eseguita per tutti gli stati del grafico, come mostrato nella figura seguente:



Come si può vedere nella figura precedente, vengono calcolate le probabilità di tutti i percorsi che portano a un nodo e vengono rimossi i bordi o i percorsi che hanno un costo di probabilità inferiore. Inoltre, si possono notare alcuni nodi che hanno una probabilità pari a zero e tali nodi non hanno bordi collegati ad essi, poiché tutti i percorsi hanno probabilità pari a zero. Il grafico ottenuto dopo aver calcolato le probabilità di tutti i percorsi che portano a un nodo è mostrato di seguito:



Per ottenere un percorso ottimale, si parte dalla fine e si procede a ritroso, poiché ogni stato ha un solo bordo in entrata.



L'algoritmo restituisce, rispetto alla soluzione vista precedentemente, un solo percorso e va ad etichettare la frase come di seguito:

- Will come nome
- Can come modello
- Spot come verbo
- Mary come nome

Essendo i tags proposti tutti corretti, si può concludere che il modello, in accoppiata all'algoritmo di Viterbi, riesce ad etichettare con successo le parole con i loro Pos Tag appropriati.