Module 3

Syntax Analysis

What is Syntax analysis?

- Syntactic analysis or parsing or syntax analysis is the second phase of NLP.
- Syntax analysis checks the text for meaningfulness comparing to the rules of formal grammar.
- Syntactic and Semantic Analysis differ in the way text is analyzed.
- In the case of syntactic analysis, the syntax of a sentence is used to interpret a text. In the case of semantic analysis, the overall context of the text is considered during the analysis

- A lexical analyzer can identify tokens with the help of regular expressions and patterns.
- But it cannot check the syntax of a given sentence due to the limitations of the regular expression.
- Regular expression cannot check balancing tokens such as parenthesis.
- Therefore this phase uses Context Free Grammar (CFG).
- CFG is a superset of regular grammar.

Syntax analysis

Part-Of-Speech tagging(POS)

Tag set for English (Penn Treebank)

Types of POS

- Rule based POS tagging
- Stochastic POS tagging

Issues in POS

- Multiple tags & words
- Unknown words

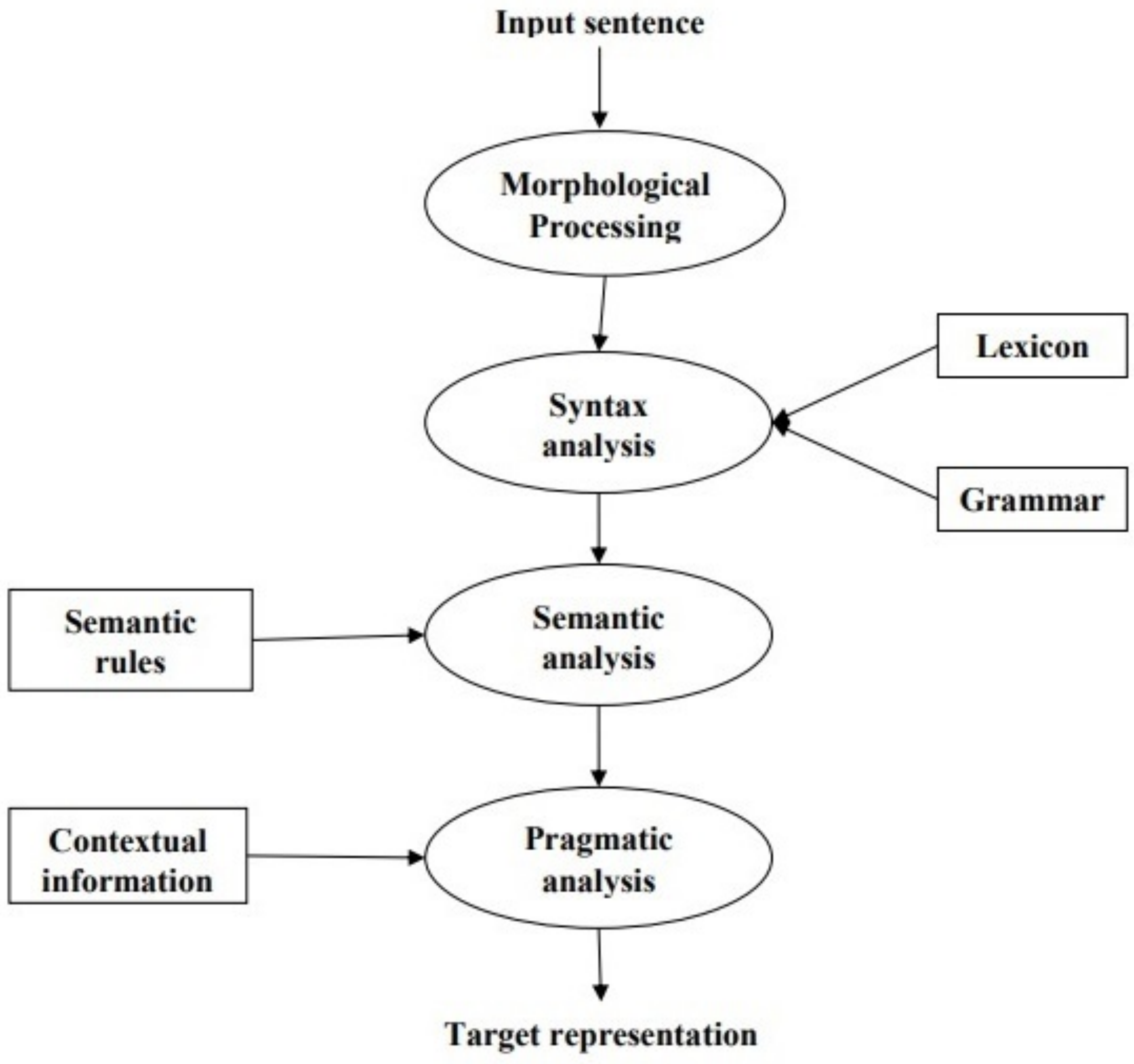
Introduction to CFG

Sequence labeling

- Hidden Markov Model (HMM)
- Maximum Entropy
- Conditional Random Field (CRF)

Syntax analysis

- Syntax analysis is the second phase of NLP. The purpose of this phase is to draw exact meaning, or you can say dictionary meaning from the text. Syntax analysis checks the text for meaningfulness comparing to the rules of formal grammar. For example, the sentence like “hot ice-cream” would be rejected by semantic analyzer.

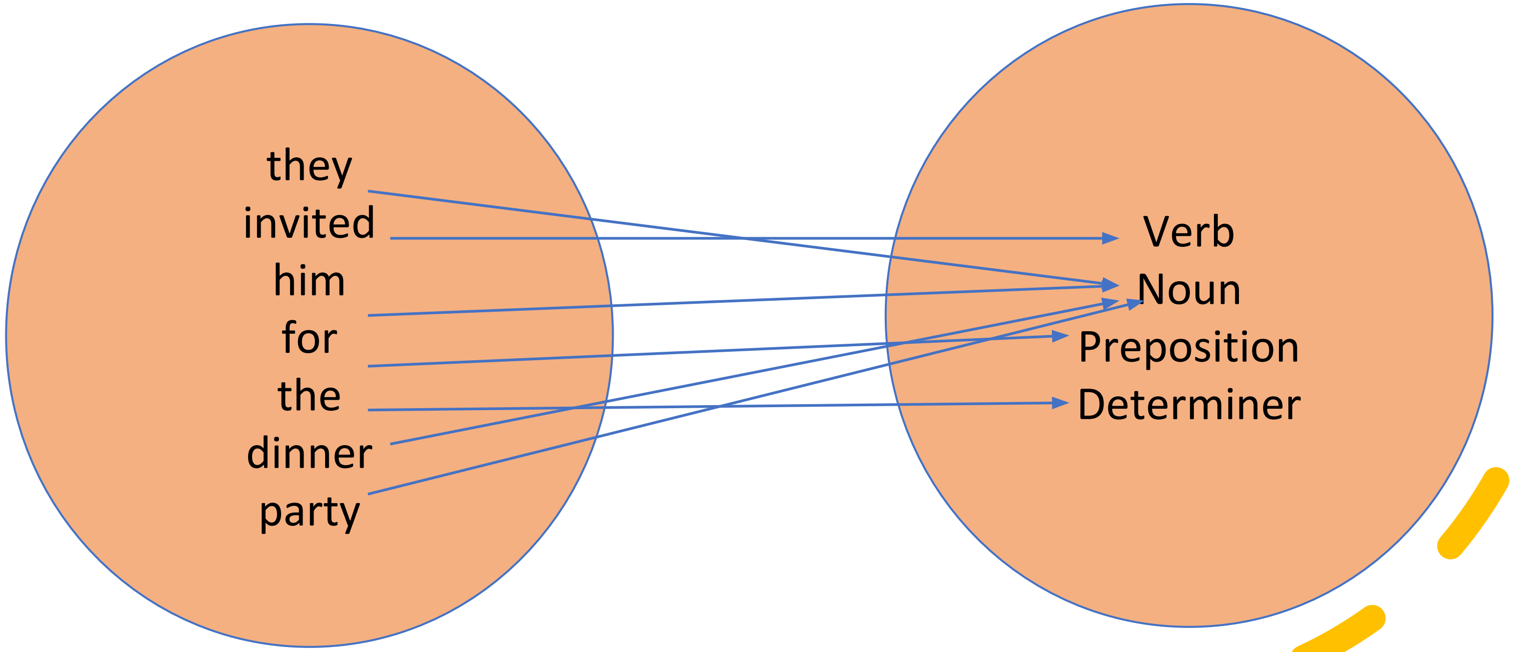


Tagging

Tagging is a kind of **classification** that may be defined as the **automatic assignment of description to the tokens**. Here the descriptor is called tag, which may represent one of the **part-of-speech, semantic information and so on.**

Part of Speech Tagging

Part-of-Speech (POS) tagging, then it may be defined as the **<u>process of assigning one of the parts of speech to the given word.</u>** It is generally called POS tagging. In simple words, we can say that POS tagging is a **task of labelling each word in a sentence with its appropriate part of speech.**



Part of Speech Tag (POS)

Syntactic category or class of any word in a natural language sentence. e.g. Noun, Verb, Adjective, Adverb etc.

E.g. Our dog chased a brown cat away from the home.

Word	POS Category
Our	Pronoun
a, the	Determiners
dog, cat, home	Nouns
brown	Adjective
chased	verb (past tense)
away	Adverb
from	preposition

POS Tagging Problem:

Problem is to identify what is the actual category for each of the word.

Given a text of English, identify the parts of speech of each word

Text: **The boy put the toys in the bag.**

POS Category	Words
Noun (N)	boy, toys, bag
Verb (V)	put
Preposition(P)	in
Determiner (Det)	the

- POS tags tell us more information about the word and its overall role in the sentence
- POS tag of a word also tells some information about its neighbouring words. e.g. Nouns are generally preceded by adjectives and/or determiners.

  e.g. The girl, an ant, handsome boy
- Identifying POS tags is an important initial step for more complex downstream NLP tasks

–Parsing

–Information Extraction

–Sentiment Analysis

–Machine Translation

POS tags can be broadly categorized into two categories:

Closed class:

- Relatively fixed set of words (limited in number)
- Addition of new closed words is very rare.
- Mostly functional: to tie the concepts of a sentence together.

e.g. Prepositions, Determiners, Pronoun, Connectives

Open class:

- Cannot associate a fixed set of words, new words can be frequently encountered with such POS tags
- Mostly content bearing: they refer to objects, actions and features in the world.
- Addition of new open words is very frequent.

e.g. Nouns , Verbs, Adjectives, Adverbs

POS: Level of Detail

- Decision of to take a very coarse grained level or to go to the more fine grained level.
- In coarse grained level, we can identify a word as a noun but we can't tell whether it is a singular noun or a plural noun.
- In fine grained level, we can give grammatical details of the word. For e.g. verb is belong to past tense, future or present etc.
- In very fine grained level, too many part of speech tags which leads to confusion.

POS Tag set Examples

Brown corpus tagset (87 tags):

http://www.scs.leeds.ac.uk/amalgam/tagsets/brown.html

Penn Treebank tagset (45 tags):

http://www.cs.colorado.edu/~martin/SLP/Figures/

C7 tagset (146 tags)

http://www.comp.lancs.ac.uk/ucrel/claws7tags.html

Penn Treebank (PTB)

- Very popular part of speech tagset is by University Pennsylvania.
- It is called Penn Treebank (PTB).
- It contains 45 part of speech tags.
- Fairly standardized for English.
- Popular NLP tools such as Stanford CoreNLP, spaCY used this tag set.
- POS tags are small and compact with 2-4 capital characters.

Upenn Treebank Tagset

Noun Types

POS Type	Explanation	Examples
NN	Singular Common Noun	Woman, Orange, Table
NNS	Plural Common Noun	Women, Oranges, Tables
NNP	Singular Proper Noun	Priya, Zenith, Jack
NNPS	Plural Proper Noun	Indians, Americas

Verb Types

POS Type	Explanation	Examples
VB	Base form of a Verb	walk, play, eat, read
VBD	Past Tense of a Verb	Walked, played, ate, read
VBN	Past Participle of a Verb	Walked, played, eaten, read
VBG	Gerund form of a Verb	fishing, walking, reading
VBZ	3rd person verb on present tense	Walks, plays, eats, read, is
VBP	Non 3rd person verb on present tense	Walk, play, eat, read, am, are
MD	Modal Verb	Can, may, should

Adjective & Adverb Types

POS Type	Explanation	Examples
JJ	Adjective	Intelligent, small, fast
JJR	Comparative Adjective	Better, smaller, faster
JJS	Superlative Adjective	Best, smallest, fastest
RB	Adverb	Back, behind, fast, slow
RBR	Comparative Adverb	Slower, faster
RBS	Superlative Adverb	Slowest, fastest

Pronoun, Determiner, Preposition Types

POS Type	Explanation	Examples
PRP	Pronoun	He, she, they, I, we
PRP$	Possessive Pronoun	His, her, your, our
POS	Possessive Marker	India's, Asian's
DT	Determiner	The, a
CC	Coordinating Conjunction	And, or, also, but
IN	Preposition	In, under, of, from, with
CD	Cardinal Number	20, two

Statement: **I am a girl.**

Word	POS Type	Explanation
I	PRP	Pronoun
am	VBP	Non 3rd person verb on present tense
a	DT	Determiner
girl	NN	Singular Common Noun

Statement: **Kavya is a intelligent girl.**

Word	POS Type	Explanation
Kavya	NNP	Singular Proper Noun
is	VBZ (Aux. verb)	3rd person verb on present tense
a	DT	Determiner
intelligent	JJ	Adjective
girl	NN	Singular Common Noun

Statement: **And now for something completely different**

Word	POS Type	Explanation
And	CC	Coordinating Conjunction
now	RB	Adverb
for	IN	Preposition
Something	NN	Singular Common Noun
completely	RB	Adverb
different	JJ	Adjective

Statement: **They refuse to permit us.**

They	refuse	to	permit	Us
PRP	VBP	Det	VB	PRP

Statement: **They refuse to permit us to obtain refuse permit.**

They	refuse	to	permit	us	To	obtain	refuse	permit
PRP	VBP	Det	VB	PRP	Det	VB	NN	NN

Automated POS Tagging

- It is the process of assigning a POS tag to each word in an input text.
- **Input:** Sentence with tokenized words or sentence output from morphological parsing, set of POS tags.
- **Output:** POS tag is assigned unambiguously to each word.

Difficulties/Challenges

- A word may have different POS tags. For e.g. to sort the numbers. Sort is verb. Bubble Sort: Sort is Noun.
- Closed class POS tags are not ambiguous but their frequency is also limited in a statement.
- Generally words with open class are more prone to multiple POS tags.
- Open class POS tags occur a lot more frequently and its tagging depends on its neighbouring context that is previous and future words.

Examples of ambiguities in POS tagging

- The **attack/NN** was brutal.
- King was planning to **attack/VB** neighbouring states.
- Tigers usually **attack/VBP** their prey in a group.

- On Sunday, I read two **books/NNS.**
- During winter season, he **books/VBZ** a flight ticket to avail discount.
- They will **book/VBP** a flight on Sunday.

- I read **that/DT** book several times.
- We thought **that/IN** you are in class.
- We were talking about a village group **that/WDT** had won the first prize in cricket.

Some more examples which are ambiguous for humans also!!!

Look and Flies words are Noun as well as verb.

Look word:

Look with Noun tag	Looks as Verb tag
Her looks are keen.	He looks(VBZ) angry.
Take a quick look of a room	She looked(VBD) straight

Fly word:

Fly with Noun tag	Fly as Verb tag
Flies are irritating	Kite flies swiftly in air.
Candidates were dropping like flies during the technical interview.	Insects flying over the water.

- Each word does not have a unique part of speech tags and it might depend on the context in this the word is being used.
- Only by seeing the context, we might be able to identify what is the actual part of speech tag is being used in this particular context.

word="back" has 4 and word "still" has 6 Part of Speech (POS) tags.

Word="Back" with sentence	POS tags
College students prefers **back** benches	Adjective
There is new restaurant at the **back** of our office building.	Noun
Students are hesitant for going **back** to offline college after pandemic over.	Adverb
Dog will bite if you **back** him into a corner.	Verb

Word="still" with sentence	POS tags
Still waters run deep	Adjective
the **still** of late evening	Noun
She **still** leaves there.	Adverb
River waters **stilled** by dams	Verb
It's past midnight but she's **still** doing her homework.	Prepositions
It's a small car, **still** it's surprisingly spacious	conjunctions

- In English language, ambiguity with more no. of POS tags (more than four) is uncommon but 2 tags are common.
- We can't ignore.
- Success of POS tagging is crucial for NLP applications in which machine have to understand language at syntactic level.
- For e.g. Applications such as Machine Translation, recognition

- Ambiguity problem is not rare. Therefore we have to tackle with variety ways.
- To do correct identification of part of speech tagging, only word is not sufficient, but local context is required. That is neighbor hood information of word whose POS is to be determine.

For e.g. **Peter saw her**.

Cannot tell the saw is actually a verb and not a noun.

saw: A hand tool for cutting wood

saw: Past tense of see

- Some word are unambiguous in nature. Carry only one POS.
- Some words are ambiguous.
- Probability technique can be applied.
- Also when we use particular language (English Language), we follow certain rules such as
- Two determiners rarely follow each other. **(e.g. the a girl)**
- Similar 2 base forms of the word they do not follow each other.

  e.g. want eat is wrong , want to eat is correct
- Determiner is always followed by an adjective or noun.

  (e.g. the table, handsome boy, the beautiful girl)
- This can be useful information that if the previous word is identified as a determiner then it more likely that the next word will be adjectives or noun.

- Distinguishing **past tenses (VBD) with past participles (VBN) for irregular verbs** is easy because they have different forms.
- For regular verbs, remember to check the existences of **auxiliary verbs or whether the verbs function as adjectives.**
- VBN: has or have + the past participle
- be + the past participle (be sometimes is omitted)
- Modal verbs as auxiliary verbs: could / may/ should / might/ would / must + have + VBN
- VBNs often used as participial adjectives, whereas VBD often follow a subjective. e.g. The girl **talked (VBD)** to me. I have **forgotten(VBN)** my lines.

Statement: **They refuse to permit us.**

They	refuse	to	permit	Us
PRP	VBP	Det	VB	PRP

Statement: **They refuse to permit us to obtain refuse permit.**

They	refuse	to	permit	us	To	obtain	refuse	permit
PRP	VBP	Det	VB	PRP	Det	VB	NN	NN

Automated POS Tagging

- It is the process of assigning a POS tag to each word in an input text.
- **Input:** Sentence with tokenized words or sentence output from morphological parsing, set of POS tags.
- **Output:** POS tag is assigned unambiguously to each word.

Difficulties/Challenges

- A word may have different POS tags. For e.g. to sort the numbers. Sort is verb. Bubble Sort: Sort is Noun.
- Closed class POS tags are not ambiguous but their frequency is also limited in a statement.
- Generally words with open class are more prone to multiple POS tags.
- Open class POS tags occur a lot more frequently and its tagging depends on its neighbouring context that is previous and future words.

Examples of ambiguities in POS tagging

- The **attack/NN** was brutal.
- King was planning to **attack/VB** neighbouring states.
- Tigers usually **attack/VBP** their prey in a group.

- On Sunday, I read two **books/NNS.**
- During winter season, he **books/VBZ** a flight ticket to avail discount.
- They will **book/VBP** a flight on Sunday.

- I read **that/DT** book several times.
- We thought **that/IN** you are in class.
- We were talking about a village group **that/WDT** had won the first prize in cricket.

Some more examples which are ambiguous for humans also!!!

Look and Flies words are Noun as well as verb.

Look word:

Look with Noun tag	Looks as Verb tag
Her looks are keen.	He looks(VBZ) angry.
Take a quick look of a room	She looked(VBD) straight

Fly word:

Fly with Noun tag	Fly as Verb tag
Flies are irritating	Kite flies swiftly in air.
Candidates were dropping like flies during the technical interview.	Insects flying over the water.

- Each word does not have a unique part of speech tags and it might depend on the context in this the word is being used.
- Only by seeing the context, we might be able to identify what is the actual part of speech tag is being used in this particular context.

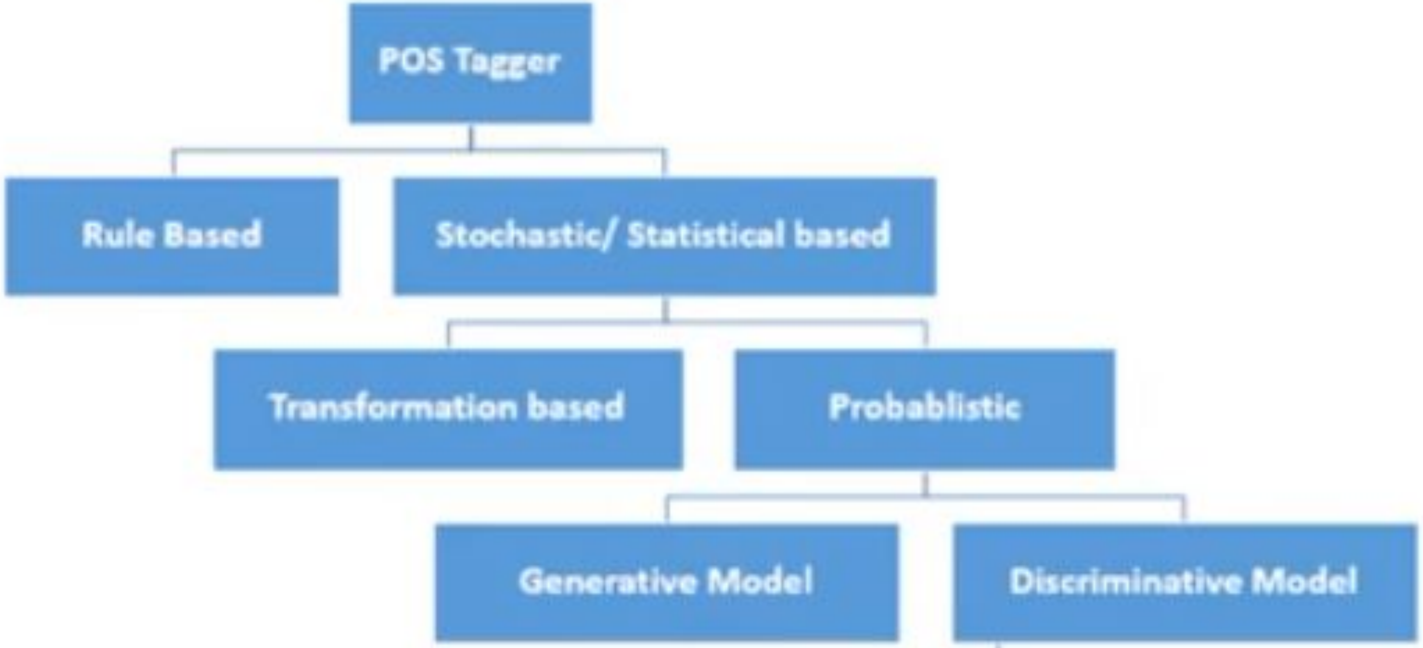
word="back" has 4 and word "still" has 6 Part of Speech (POS) tags.

Word="Back" with sentence	POS tags
College students prefers **back** benches	Adjective
There is new restaurant at the **back** of our office building.	Noun
Students are hesitant for going **back** to offline college after pandemic over.	Adverb
Dog will bite if you **back** him into a corner.	Verb

Word="still" with sentence	POS tags
Still waters run deep	Adjective
the **still** of late evening	Noun
She **still** leaves there.	Adverb
River waters **stilled** by dams	Verb
It's past midnight but she's **still** doing her homework.	Prepositions
It's a small car, **still** it's surprisingly spacious	conjunctions

- In English language, ambiguity with more no. of POS tags (more than four) is uncommon but 2 tags are common.
- We can't ignore.
- Success of POS tagging is crucial for NLP applications in which machine have to understand language at syntactic level.
- For e.g. Applications such as Machine Translation, recognition

Classification of Taggers



1) Rule-based Approach

- Rule-based part-of-speech tagging is the oldest approach that uses hand-written rules for tagging.
- Rule based taggers depends on dictionary or lexicon to get possible tags for each word to be tagged.
- Some language expert will sit together find out what are the symbol patterns.
- Hand-written rules are used to identify the correct tag when a word has more than one possible tag.
- Disambiguation is done by analyzing the linguistic features of the word, its preceding word, its following word and other aspects.

- For example, if the preceding word is article then the word in question must be noun. This information is coded in the form of rules.
- The rules may be context-pattern rules or as regular expressions compiled into finite-state automata that are intersected with lexically ambiguous sentence representations.
- TAGGIT, the first large rule based tagger, used context-pattern rules.

  TAGGIT used a set of 71 tags and 3300 disambiguation rules.

  These rules disambiguated 77% of words in the million-word Brown University corpus.

  Example: I want to read a book.

  Book: Noun or verb

  Rule: before book, determiner(a) is there, therefore it is a noun.

Limitations of Rule-based Approach

- Hard coded rule are required.
- Rules has to be updated based on language change.
- It needs to check lexicon every time which is not efficient approach
- Strong language experts team is required.

2) Stochastic/ Statistical based Approach

It is based on concept of probability and machine learning.

Training and Test corpus are used.

Two approaches are there

a) Transformation based Tagger b) Probabilistic based Tagger

a) Transformation based Tagger

- This tagger is based on concept of Transformation-Based Learning (TBL) approach.
- TBL uses supervised learning.
- There is assumption of pre-tagged training corpus.
- It combines idea of the rule-based and stochastic taggers.
- Like the rule based taggers, TBL is based on rules that specify what tags should be assigned to what words.
- But like the stochastic taggers, TBL is a machine learning technique, in which rules are automatically induced from the data.

Label the training set with most frequent tags

e.g. **The can was rusted.**

The/DT can/MD was/VBD rusted/VBD

Add transformation rules to reduce training mistakes.

The/DT can/NN was/VBD rusted/VBN

Explanation: ‘can’ can be noun as well as modal verb also.

It uses machine learning as well as grammar rules.

1) Model verb is never preceded by determiner.

2) Also in corpus (training data) possibility of ‘determiner model verb’ pair is zero.

Therefore can is replaced as Noun tag

Explanation: ‘rusted’ can be VBD as well as VBN also.

1) In grammar rule, VBN is preceded by VBD

2) Also in corpus, possibility of “VBD-VBD’ pair is negligible.

Statement:

Race is **incorrectly** tagged in the following statement:

is**/VBZ** expected**/VBN** to**/TO** race**/NN** tomorrow**/NN**

In the second case this race is **correctly** tagged as an NN:

the**/DT** race**/NN** for**/IN** outer**/JJ** space**/NN**

For example, in the Brown corpus, race is most likely to be a noun:

P(NNIrace) = .98 **P(VBIrace) = .02**

Therefore race is labelled as NN because its occurrence (probability) is more.

Brill's tagger learned a rule that applies exactly to this mistagging of race:

Change NN to VB when the previous tag is TO.

This rule would change race/NN to race/VB in exactly the following situation, since it is preceded by to/TO:

/VBN to/TO race/NN expected/VBN to/TO **race/VB**

b) Probabilistic:

Approach is to "pick the most-likely tag for this word".

Two approaches are there **whether you generate the data from the class or the class from the data.**

Problem statement: Data available of the form [d, c]

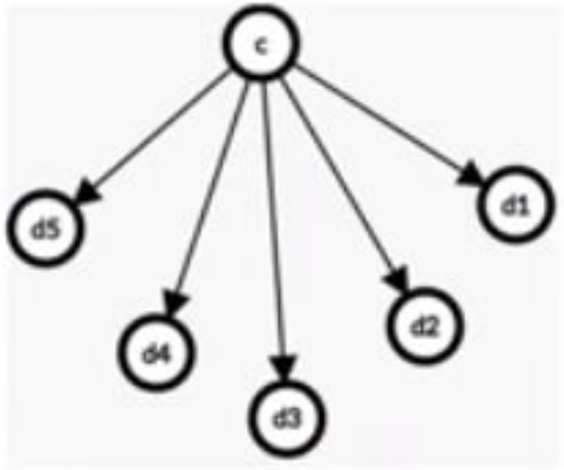
where d is observations and c is hidden classes.

Two types of Model are there

i) Generative Model

ii) Discriminative Model

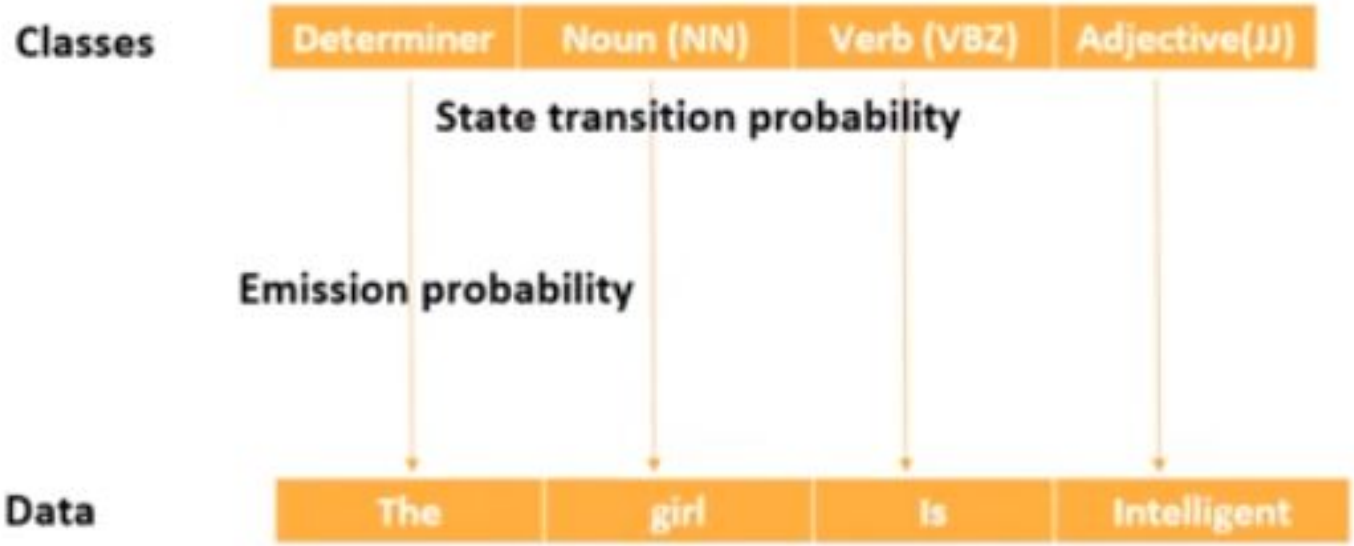
1) Generative Model



Example are **Hidden Markov Model, Naive Bayes Classifier**

- In generative model, we assume that class is there and it generates data.
- For e.g. class is given (subjects), and all words are generated from subject.
- For e.g. Part- of- Speech Tagging: POS tags are there and words are generated from that.
- Here flow is downward (Classes generates data)

Generative Model



For e.g. POS tag noun(class) generates data like girl, boy, man, woman, water etc.

Statement is The girl is intelligent. In this case, determiner class generates the, NN class generates girl, VBZ generates is and JJ generates adjective.

2)Discriminative Model:

In the discriminative model, data is there and you assume that hidden state is a generated from the data. Here flow is upward. (that data identifies class)

For e.g. keywords of the subjects are given(data) and from the keywords we determine subject. Keywords are data and subject is class. Class is hidden. By seeing nature of data, we can assign probability of the particular class.

Example are **Maximum entropy model, Conditional Random field.**

Discriminative Model

For e.g. For every word, various features are given with respect to previous words and future words. From that features, we can decide POS tag for that word. That is flow is upward. For a given word, using features of neighboring word, we determine class or tag.

Statement: **They refuse to permit us.**

They	refuse	to	permit	Us
PRP	VBP	Det	VB	PRP

Statement: **They refuse to permit us to obtain refuse permit.**

They	refuse	to	permit	us	To	obtain	refuse	permit
PRP	VBP	Det	VB	PRP	Det	VB	NN	NN

Examples of ambiguities in POS tagging

- The **attack/NN** was brutal.
- King was planning to **attack/VB** neighbouring states.
- Tigers usually **attack/VBP** their prey in a group.

- On Sunday, I read two **books/NNS.**
- During winter season, he **books/VBZ** a flight ticket to avail discount.
- They will **book/VBP** a flight on Sunday.

- I read **that/DT** book several times.
- We thought **that/IN** you are in class.
- We were talking about a village group **that/WDT** had won the first prize in cricket.

Some more examples which are ambiguous for humans also!!!

Look and Flies words are Noun as well as verb.

Look word:

Look with Noun tag	Looks as Verb tag
Her looks are keen.	He looks(VBZ) angry.
Take a quick look of a room	She looked(VBD) straight

Fly word:

Fly with Noun tag	Fly as Verb tag
Flies are irritating	Kite flies swiftly in air.
Candidates were dropping like flies during the technical interview.	Insects flying over the water.

- Each word does not have a unique part of speech tags and it might depend on the context in this the word is being used.
- Only by seeing the context, we might be able to identify what is the actual part of speech tag is being used in this particular context.

word="back" has 4 and word "still" has 6 Part of Speech (POS) tags.

Word="Back" with sentence	POS tags
College students prefers **back** benches	Adjective
There is new restaurant at the **back** of our office building.	Noun
Students are hesitant for going **back** to offline college after pandemic over.	Adverb
Dog will bite if you **back** him into a corner.	Verb

Word="still" with sentence	POS tags
Still waters run deep	Adjective
the **still** of late evening	Noun
She **still** leaves there.	Adverb
River waters **stilled** by dams	Verb
It's past midnight but she's **still** doing her homework.	Prepositions
It's a small car, **still** it's surprisingly spacious	conjunctions

- In English language, ambiguity with more no. of POS tags (more than four) is uncommon but 2 tags are common.
- We can't ignore.
- Success of POS tagging is crucial for NLP applications in which machine have to understand language at syntactic level.
- For e.g. Applications such as Machine Translation, recognition

Challenges in POST

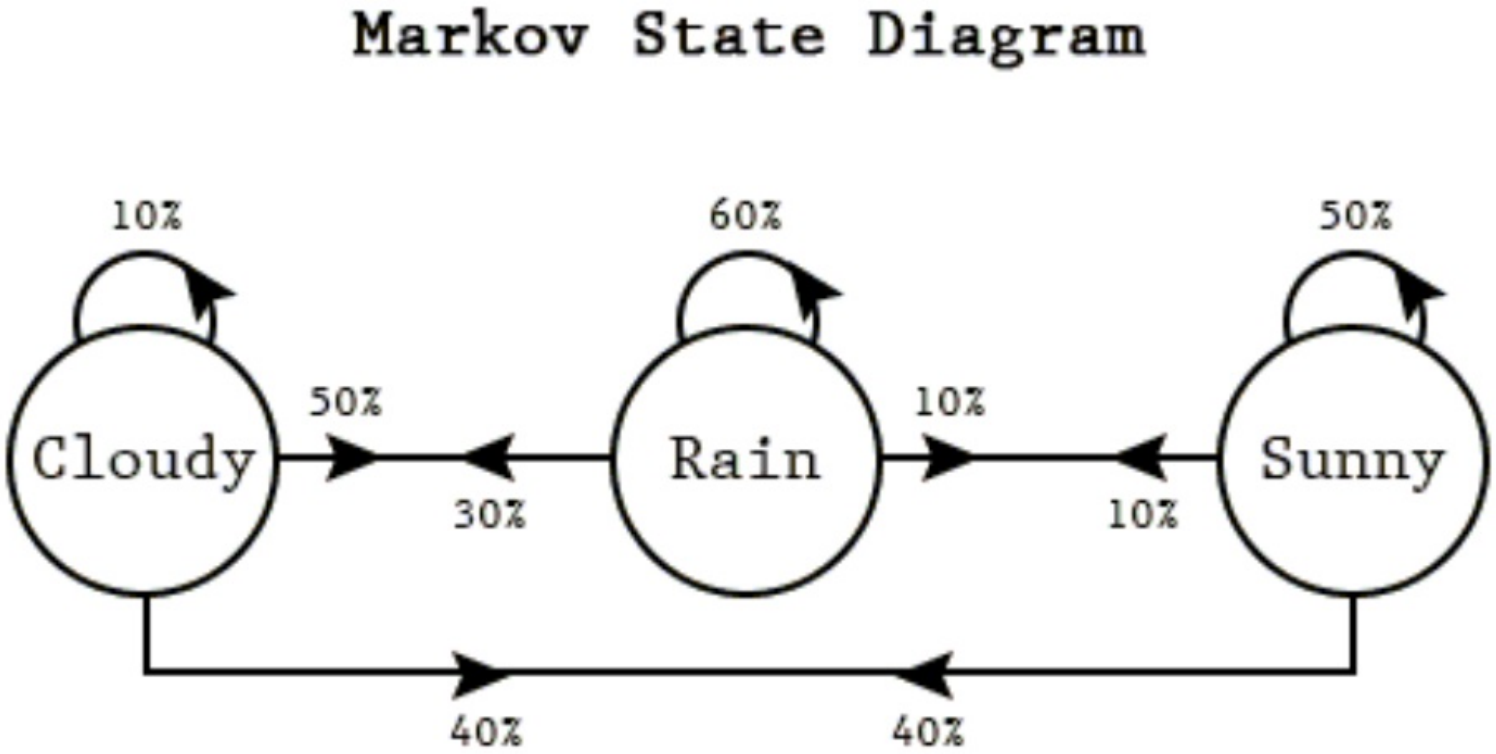
- There exists Ambiguity in the English language.
- In English many common words have **multiple meanings and**
- **Hence multiple POS**
- The job of a POS tagger is to **resolve this ambiguity accurately based on the context of use. For example, the word 'shot' can be a noun or verb.**
- If a POS tagger gives **poor accuracy then this has an adverse effect the the tasks that follow. This is called down stream error propagation.**
- To improve accuracy, POS tagging is combined with other processing.
- For e.g. joint POS tagging and dependency parsing is an approach to improve accuracy compared to independent modelling.

Markov Models

- Markov models are used to predict the future state based on the current hidden or observed state. Markov model is a finite-state machine where each state has an associated probability of in any other state after one step.
- They can be used to model real-world problems where hidden and observable states are involved.
- Consider an example:
- Suppose a bag contains four marbles, two red and two blue. After repeating this pattern several times, we begin to notice a pattern: The probability of selecting a red marble is always two out of four, or 50%.
- The reason is: The probability of selecting a particular color of marble is determined by the number of that color in the bag. In other words a past history (i.e. the content of the bag) determines the future state(i.e. the probability of selecting color of marble)

Markov chain

- A Markov Chain is a **stochastic process** that models a finite set of states, with fixed conditional probabilities of jumping from a given state to another.
- A Markov chain is a model that tells us something about the probabilities of sequences of random states/variables. A Markov chain makes a very strong assumption that if we want to predict the future in the sequence, all that matters is the current state. All the states before the current state have no impact on the future except via the current state.
- To show what a Markov Chain looks like, we can use a digraph, where each node is a state (with a label or associated data), and the weight of the edge that goes from node a to node b is the probability of jumping from state a to state b.
- Here’s an example, modelling the weather as a Markov Chain.



Transition Matrix

	C	R	S
C	0.1	0.5	0.4
R	0.3	0.6	0.1
S	0.4	0.1	0.5

Fig: weather as a Markov Chain.

We can express the probability of going from state *a* to state *b* as a **matrix component**, where the whole **matrix characterizes our Markov chain** process, corresponding to the **digraph's adjacency matrix**.

- There are three different states: Cloudy, rainy, and sunny.
- We mention the transition probabilities based on the above diagram:

If sunny today, then tomorrow

a) 50% probability for sunny b) 10% for rainy and C) 40% for cloudy

If rainy today then tomorrow

a) 10% probability for sunny b) 60% for rainy and c) 30% for cloudy

If cloudy today then tomorrow

40% probability for sunny b) 50% for rainy and c) 10% for cloudy

- Using this Markov chain we want to conclude: What is the probability that Wednesday will be cloudy if Monday is Sunny?

1) Sunny(Monday) – Sunny(Tuesday) – Cloudy(Wednesday)

The probability to cloudy Wednesday is (0.5x0.4 = 0.2)

2) Sunny(Monday)– rainy(Tuesday) – Cloudy(Wednesday)

The probability to a cloudy Wednesday is (0.1x0.3 = 0.03)

3) Sunny(Monday) – Cloudy(Tuesday) – Cloudy(Wednesday)

The probability to a cloudy Wednesday is (0.4x0.1 = 0.04)

The total probability of a cloudy Wednesday = 0.2+0.03+0.04 = 0.27

POS Tagging With Hidden Markov Model

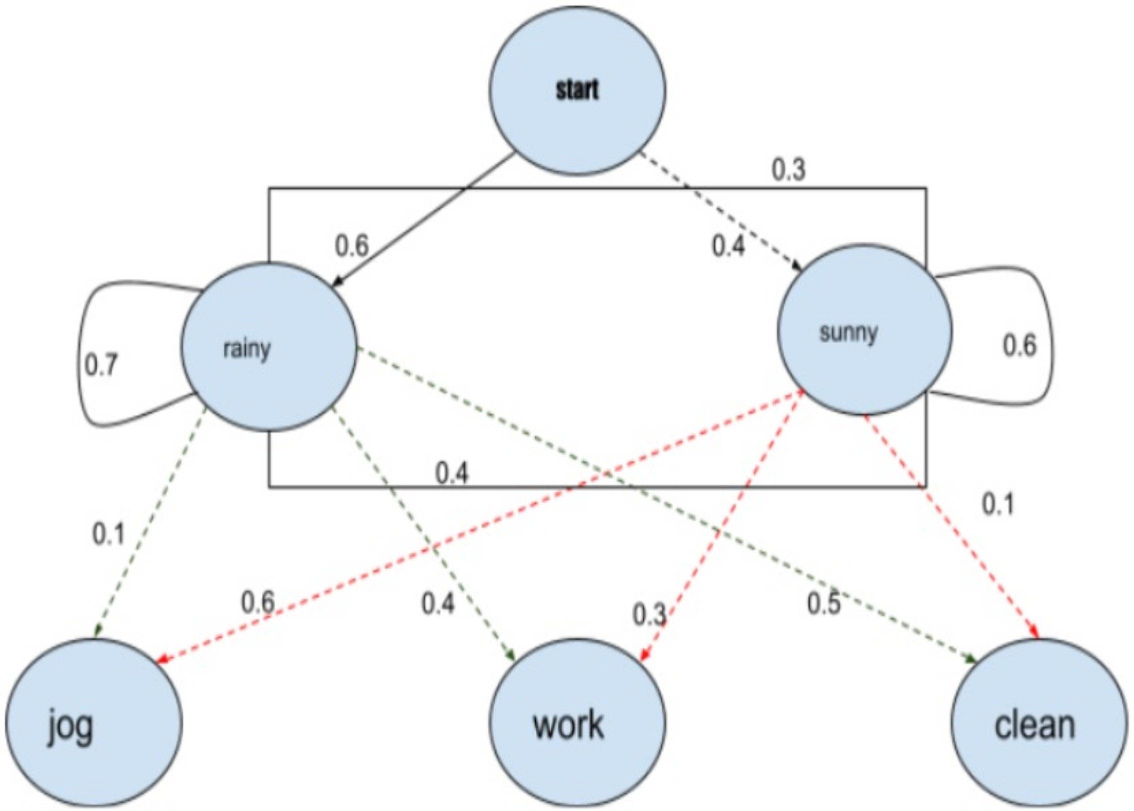
- HMM is a stochastic technique for POS tagging.
- For developing HMM model we need two types of probabilities:

  Emission probabilities and Transition probabilities and
- The transition probability is the likelihood of any sequence.
- For example "what are the chances for a noun word to come after any modal and a modal after a verb and a verb after a noun".

HMM – Hidden Markov Models

- The hidden Markov model is another type of Markov model where there are a few states hidden.
- It is a hidden variable model which can give an observation of another hidden state using Markov assumptions.
- The hidden state is the variable that cannot be directly observed but can be inferred by observing one or more states using Markov assumptions.
- Markov assumption is the assumption that a hidden variable is dependent only on the previous hidden state.
- A Markov model has five components:
    A. An initial probability distribution
    B. One or more hidden states
    C. Transition probability distribution
    D. A sequence of observations
    E. Emission probability

- A sequence of observations likelihood, also called emission probabilities. Each observation expresses the probability of an observation, generated from a state. The emission prob. Is used to define the hidden variable in terms of its next hidden state.
- We represent the Morkov model representation pictorially.



- The above hidden Markov model predicts whether someone will be jogging, working, or cleaning on a particular day.
- This action will depend upon whether the day is rainy or sunny.
- There are two hidden states: rainy and sunny.
- These are hidden states because the process output is whether the person is jogging, working, or cleaning depends upon these states.
- The sequence of observation is jog, work and clean.
- An initial probability distribution is start probability.
- Transition probability is the transition of one state to another state given the current state.
- Emission probability is the probability of observing the output.

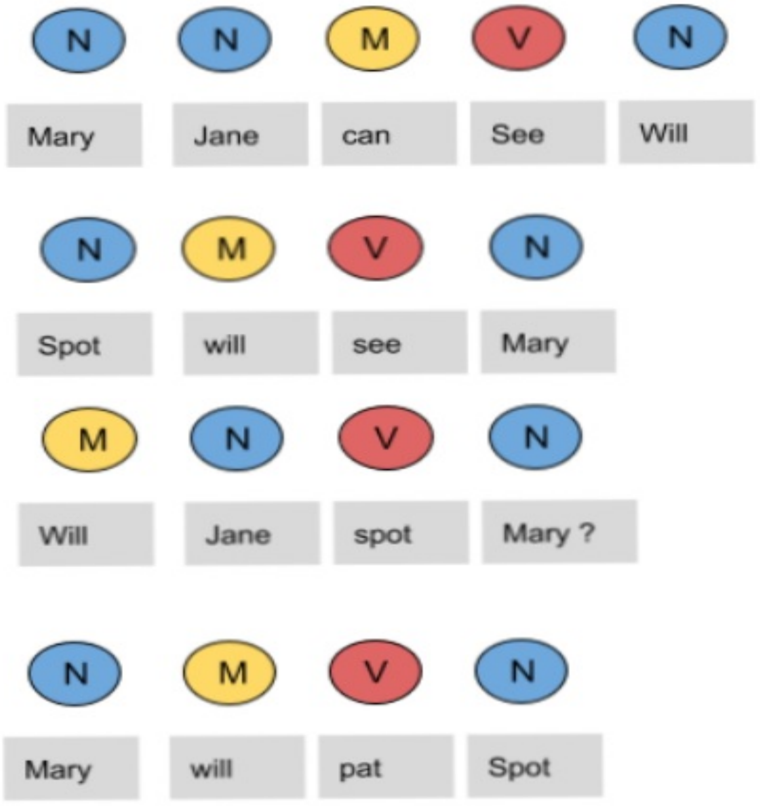
POS Tagging With Hidden Markov Model

- HMM is a stochastic technique for POS tagging.
- For developing HMM model we need two types of probabilities:

  Emission probabilities and Transition probabilities and

- The transition probability is the likelihood of any sequence.
- For example "what are the chances for a noun word to come after any modal and a modal after a verb and a verb after a noun".

- Let's take a look at how we can calculate these two probabilities for a set of sentences:
- Mary Jane can see will
- The spot will see Mary
- Will Jane spot Mary?
- Mary will pat Spot



Calculate Emission probability

Words	Noun	Modal	Verb
mary	4/9	0	0
jane	2/9	0	0
will	1/9	3/4	0
spot	2/9	0	1/4
can	0	1/4	0
see	0	0	2/4
pat	0	0	1/4

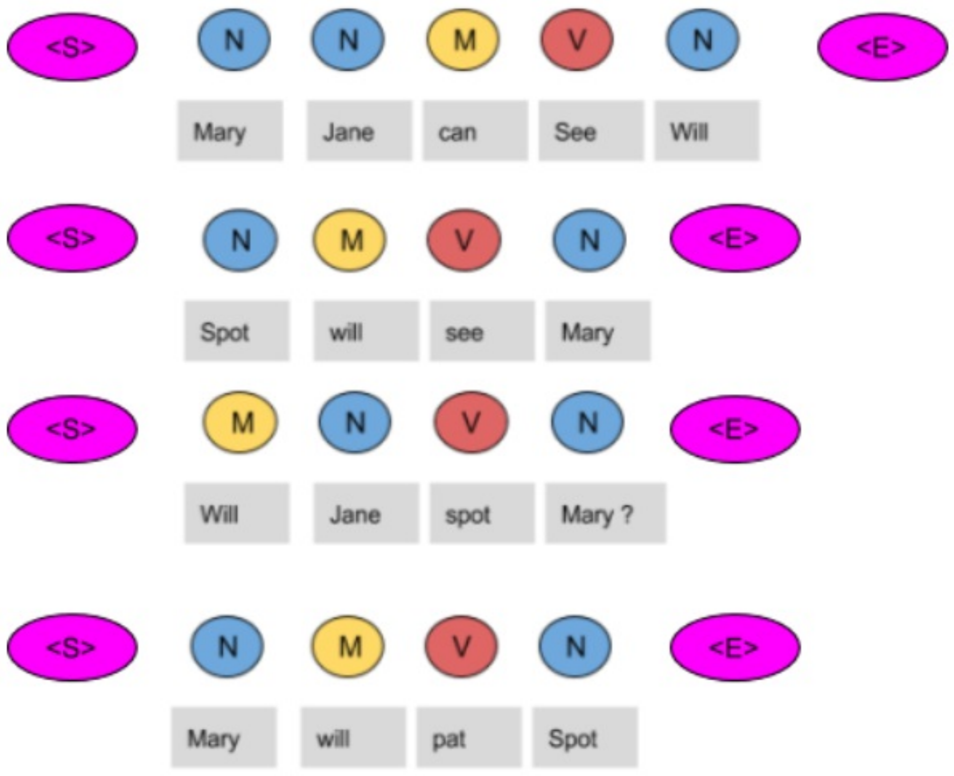
Emission probability Matrix

From the above table, we infer that

The probability that Mary is Noun = 4/9
The probability that Mary is Model = 0
The probability that Will is Noun = 1/9
The probability that Will is Model = 3/4

Calculate Transition probabilities

Add start of the sentence <s> and end of the sentence <e>.

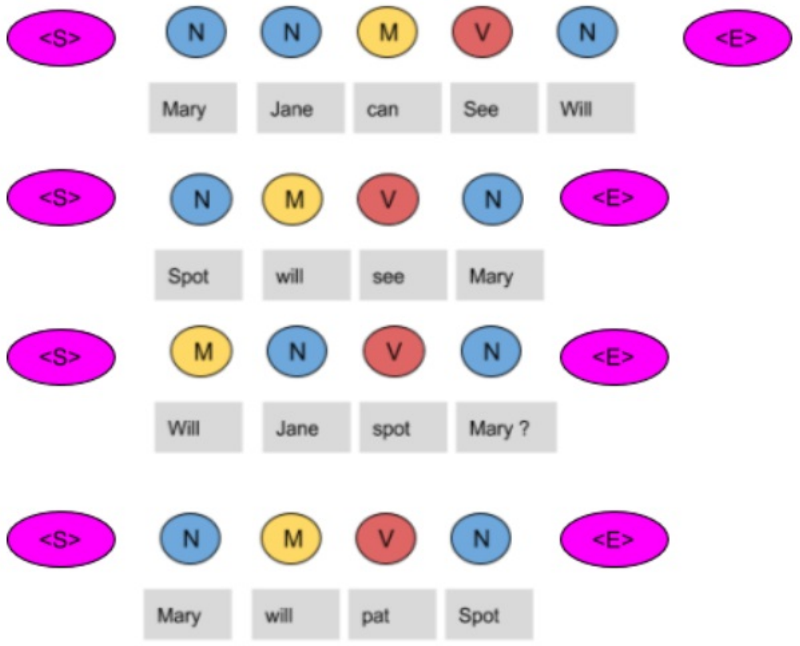


Create a table and fill it with the co-occurrence counts of the tags.

	Noun	Modal	Verb	End
Start	3	1	0	0
Noun	1	3	1	4
Model	1	0	3	0
Verb	4	0	0	0

- To create a table we need to check for the combination of parts of speeches for calculation of the transition probabilities.
- For example, we can see in the set of sentences **modal before a verb** has appeared 3 times and 1 time before a noun.

Transition table



	Noun	Modal	Verb	End
Start	3/4	1/4	0	0
Noun	1/9	3/9	1/9	4/9
Model	1/4	0	3/4	0
Verb	4/4	0	0	0

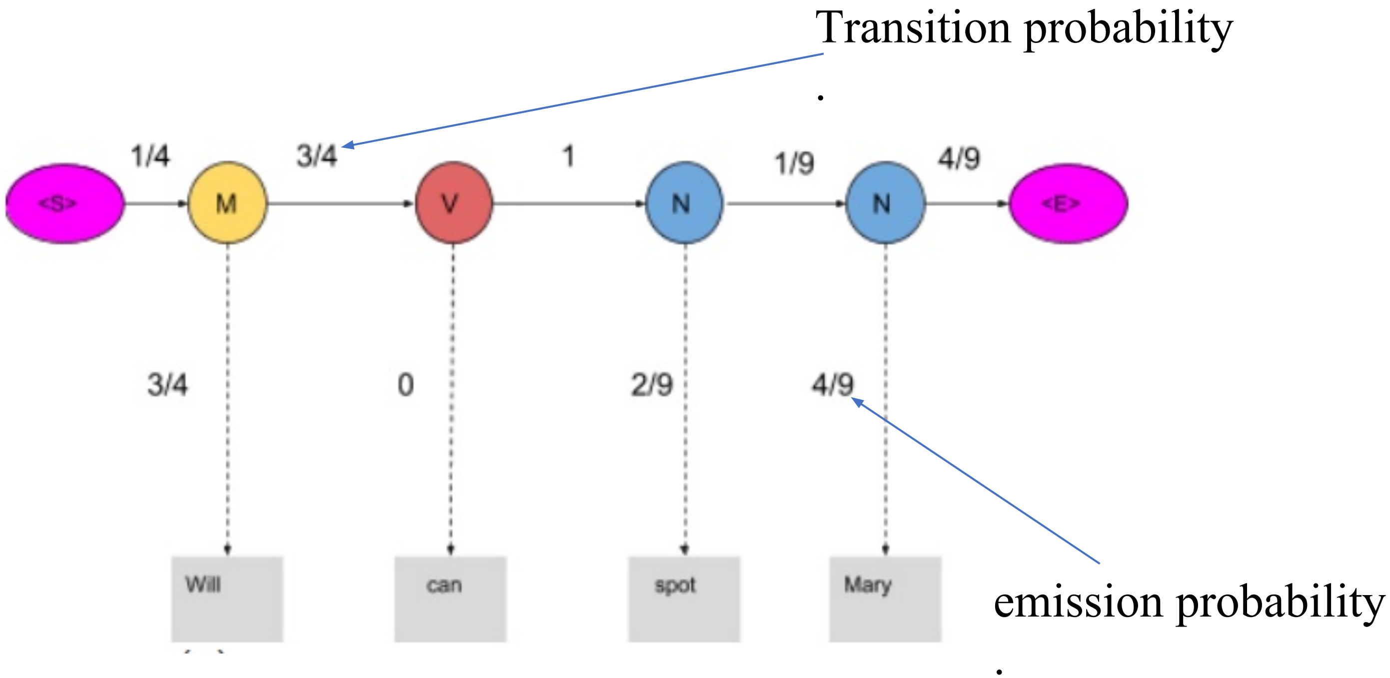
- Now in the table, we are required to check for the combination of parts of speeches for calculation of the transition probabilities.
- For example, we can see in the set of sentences modal before a verb has appeared 3 times and 1 time before a noun.
- This means it has appeared in the set four four-time and the probability of coming modal before any verb will be $\frac{3}{4}$ and before a noun will be $\frac{1}{4}$.
- Similarly performing this for every entity of the table:

Here the above values in the table are the respective transition values for a given set of sentences.

- Let’s take the sentence “**Will can spot Mary**” **Take a new sentence and tag them with the wrong tags.** Let’s see how HMM model helps us identify whether this tagging is correct or not.
- Let the sentence, ‘ Will can spot Mary’ be tagged as-
- Will as a modal (Actually will is a noun)
- Can as a verb (Actually can is a modal)
- Spot as a noun (Actually Spot is a verb)
- Mary as a noun

- Now calculate the probability of this sequence being correct in the following manner.

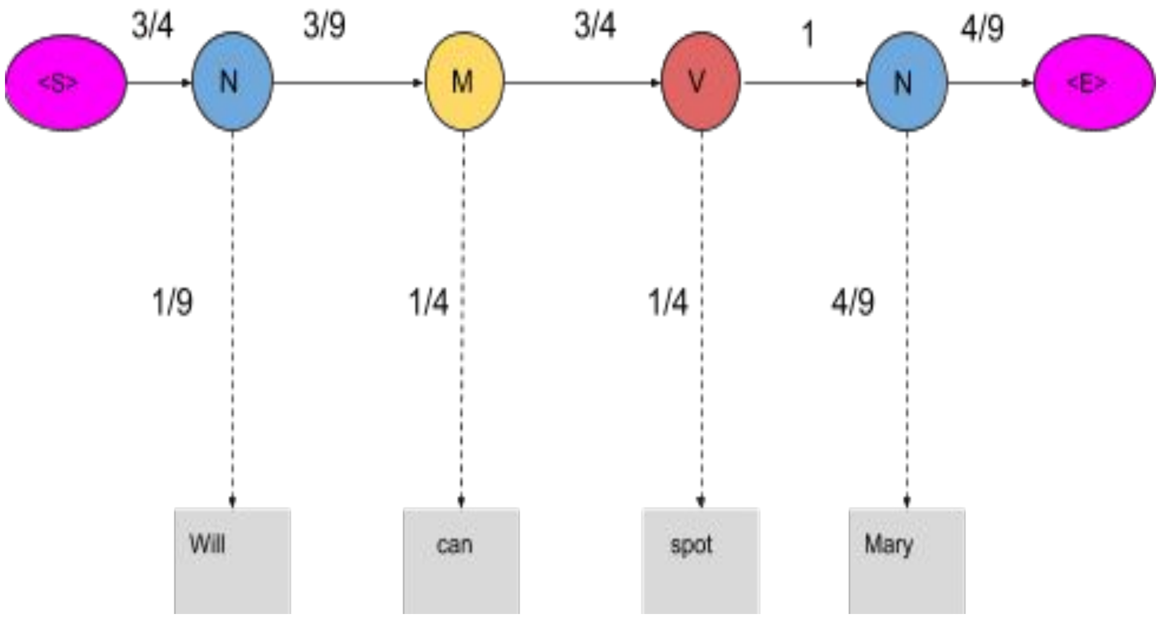
	Noun	Modal	Verb	End
Start	3/4	1/4	0	0
Noun	1/9	3/9	1/9	4/9
Model	1/4	0	3/4	0
Verb	4/4	0	0	0

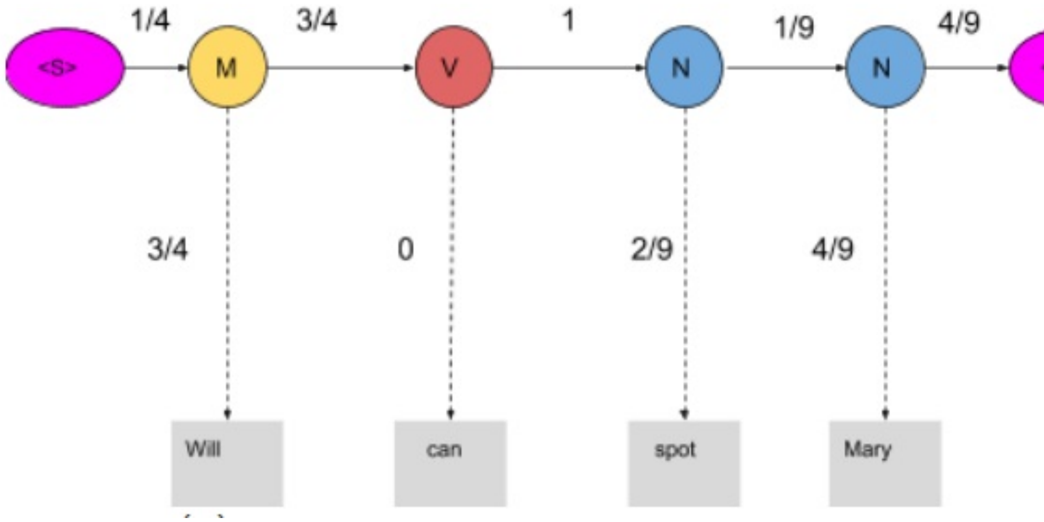


1/4*3/4*3/4*0*1*2/9*1/9*4/9*4/9=0

- The probability of the tag Model (M) comes after the tag <S> is ¼ as seen in the transition table.
- Also, the probability that the word Will is a Model is ¾ (refer emission table) .
- In the same manner, we calculate each and every probability in the graph.
- Now the product of these probabilities is the likelihood that this sequence is right. Since the tags are not correct, the product is zero.

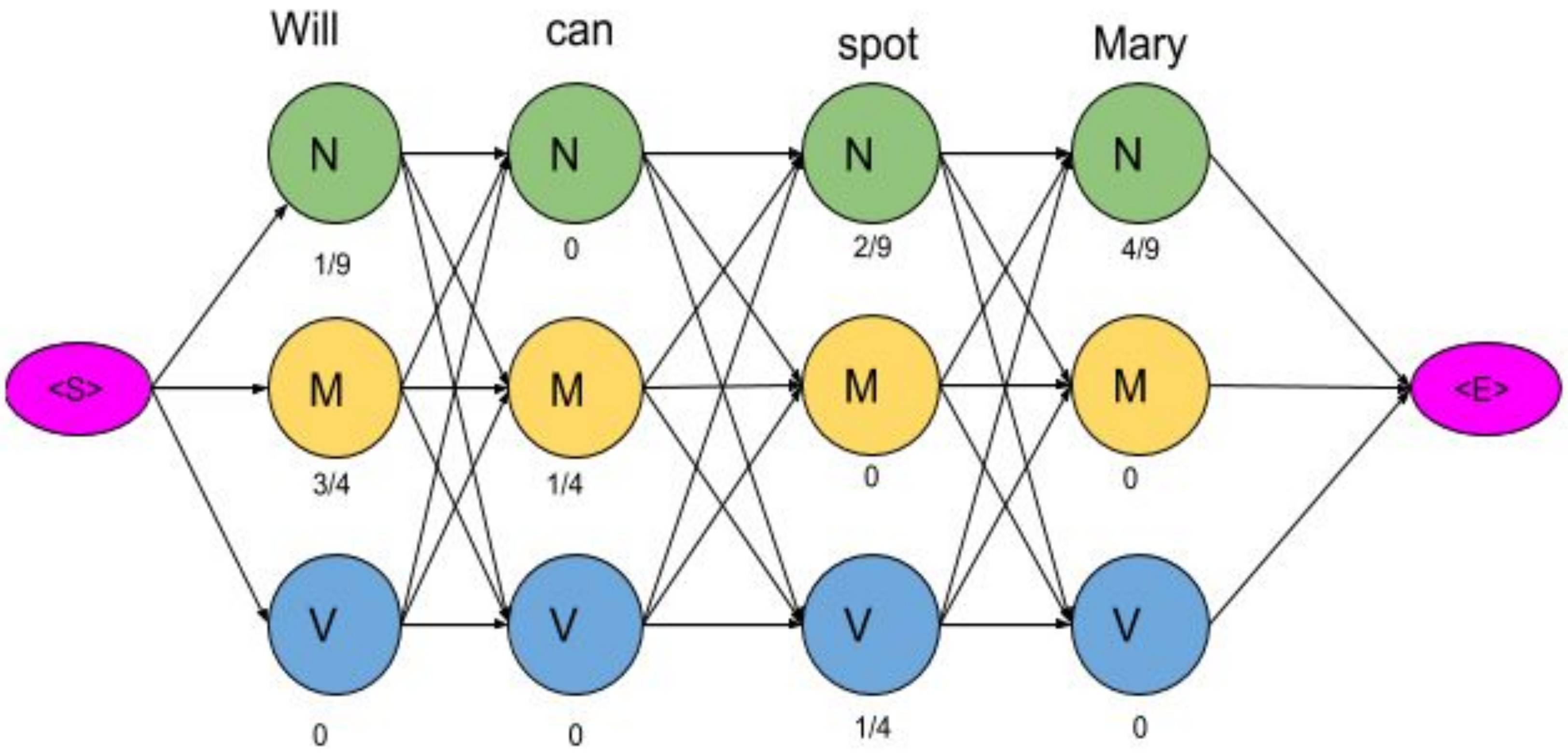
When these words are correctly tagged, we get a probability greate



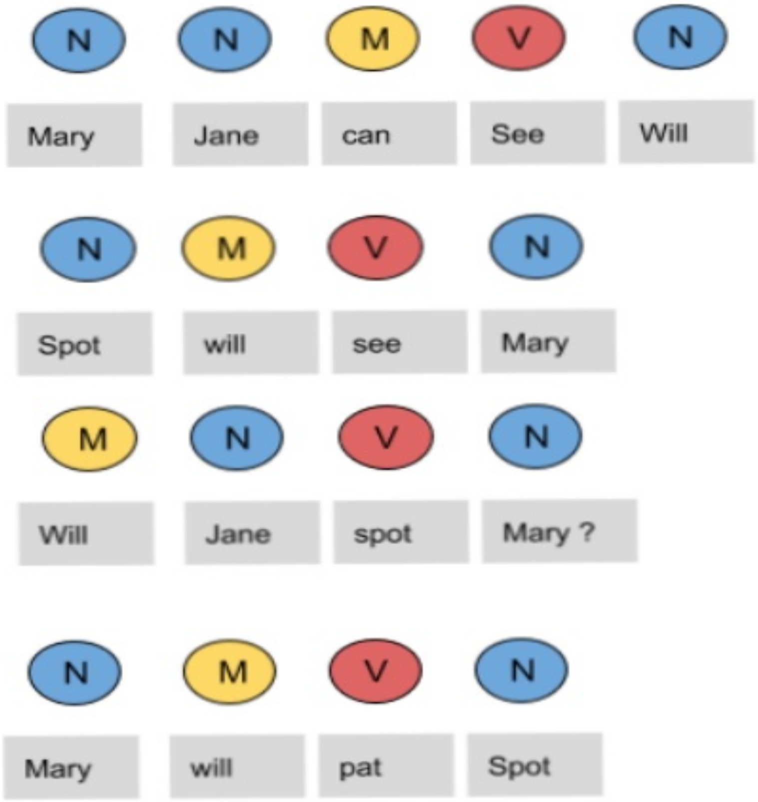


Calculating the product of these terms we get,

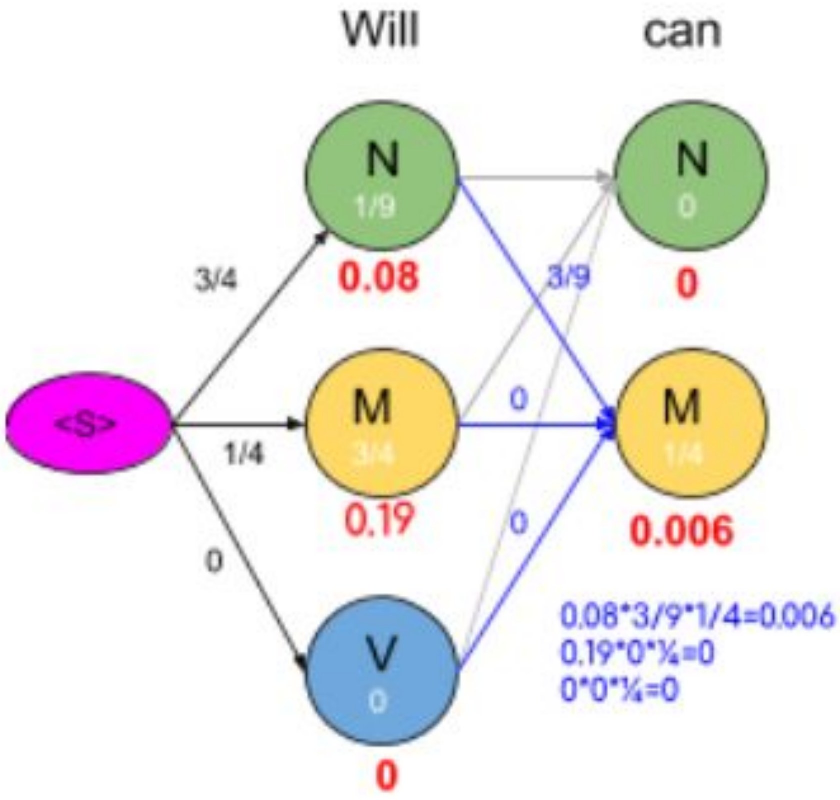
3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164

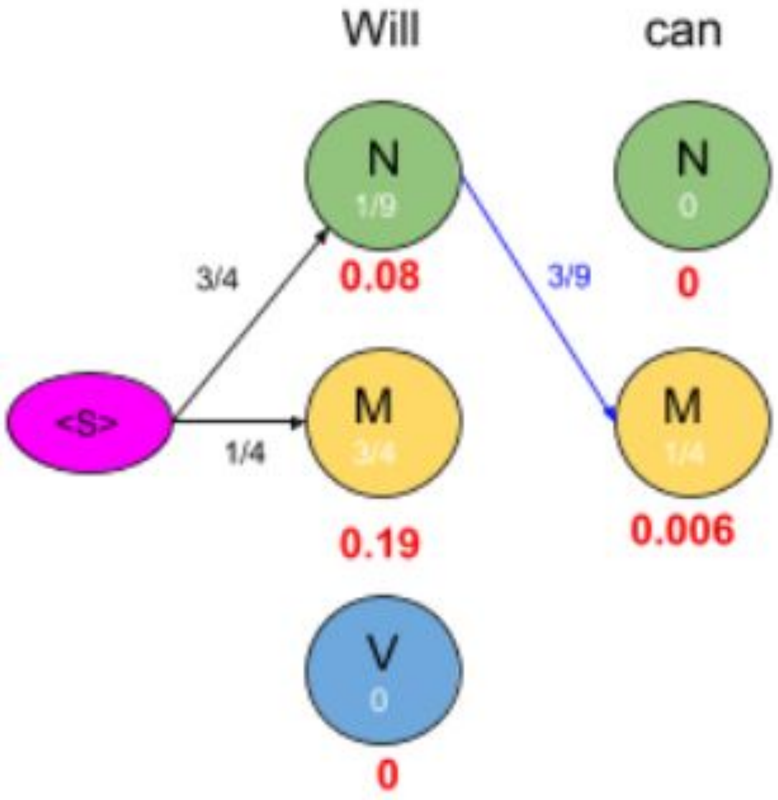


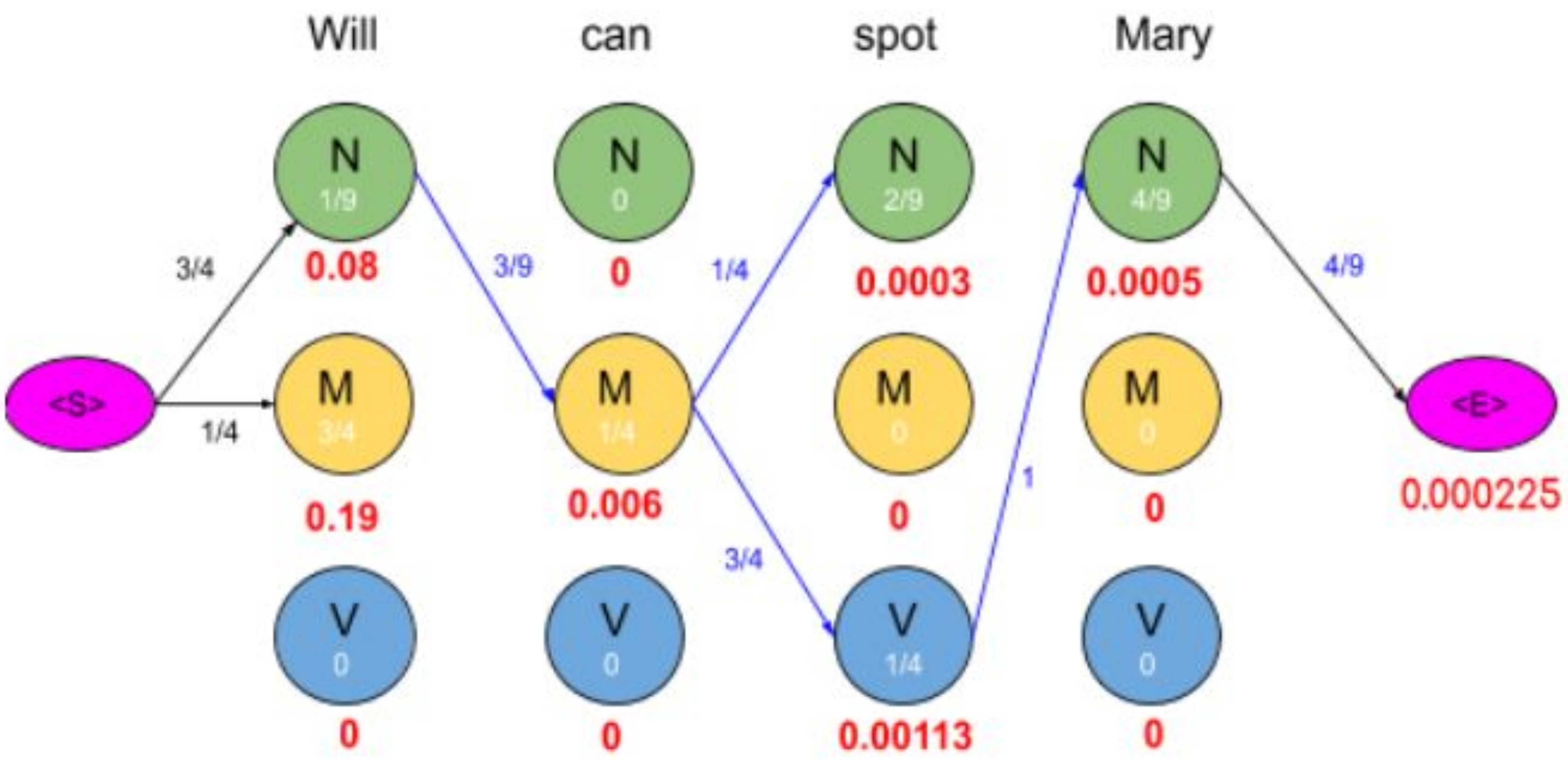
Will
can
spot
Mary
<S>
N
1/9
M
3/4
V
0
N
0
M
1/4
V
0
N
2/9
M
0
V
1/4
N
4/9
M
0
V
0
<E>



N N M V N
Mary Jane can See Will
N M V N
Spot will see Mary
M N V N
Will Jane spot Mary ?
N M V N
Mary will pat Spot

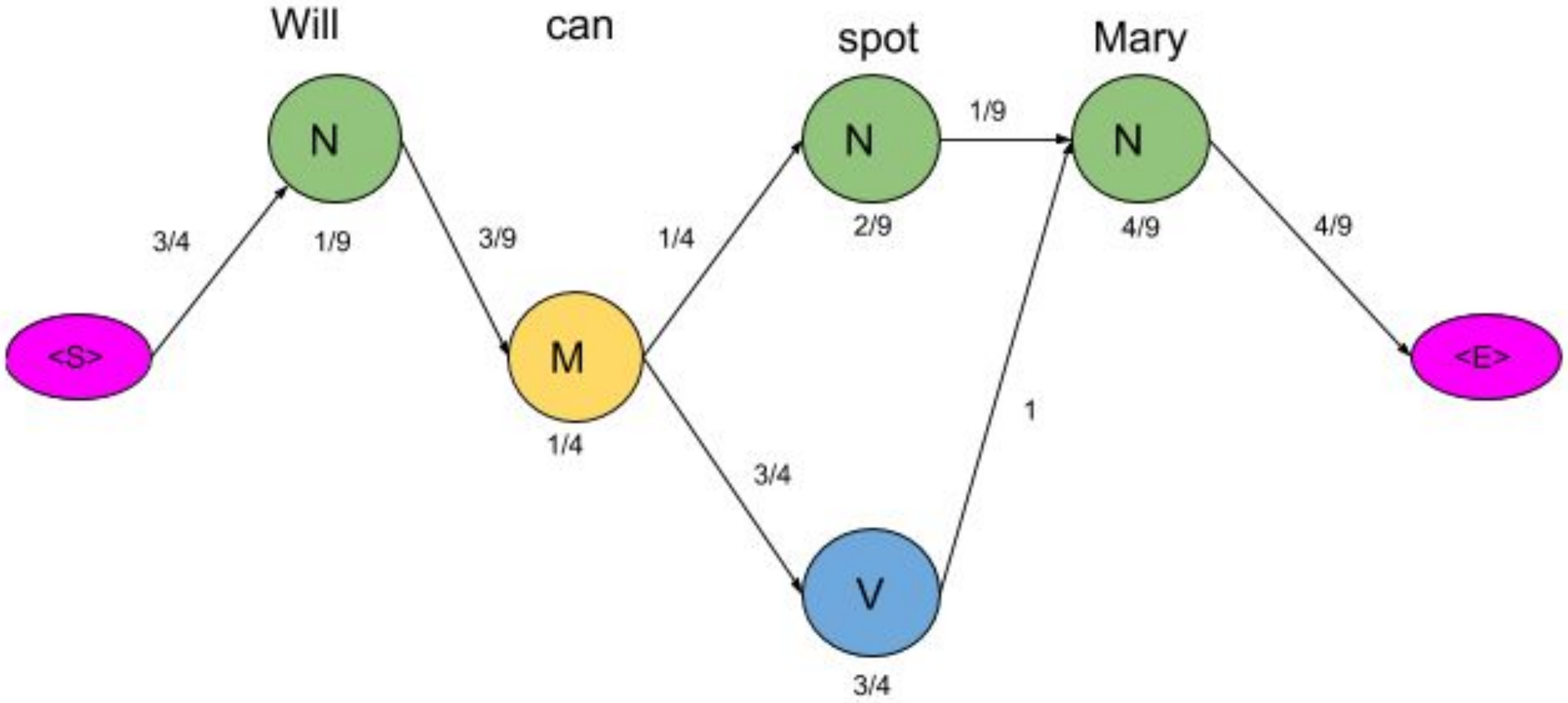

Will
can
<S>
N
1/9
0.08
M
3/4
0.19
V
0
0
N
0
0
M
1/4
0.006
3/4
1/4
0
3/9
0
0
0.08*3/9*1/4=0.006
0.19*0*¼=0
0*0*¼=0

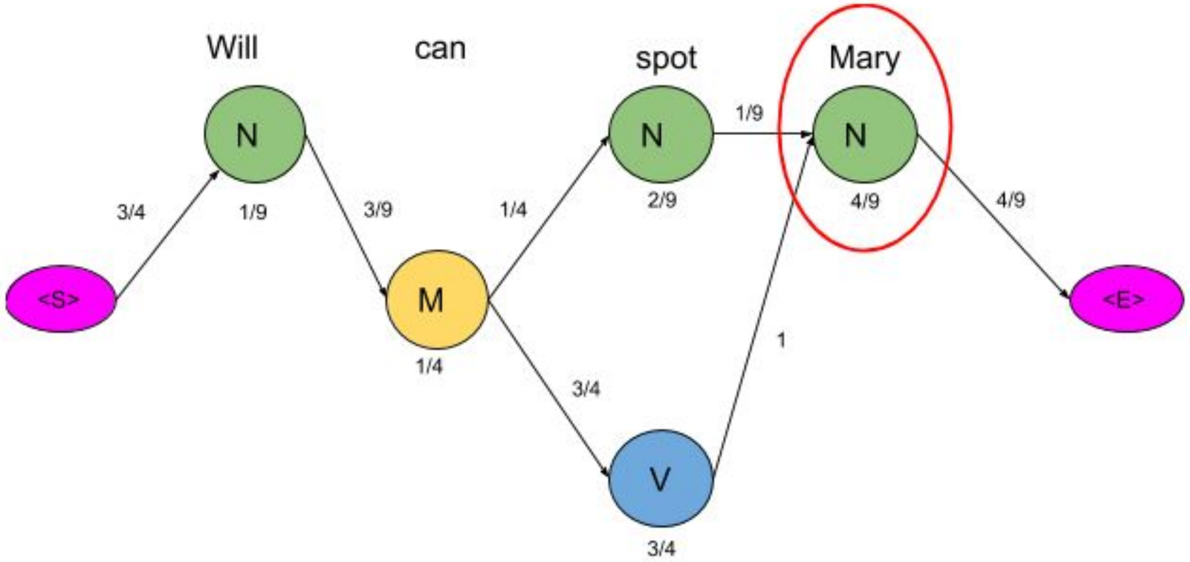

Will
can
<S>
N
1/9
0.08
M
3/4
0.19
V
0
0
N
0
0
M
1/4
0.006
3/4
1/4
3/9



Will
can
spot
Mary
N
1/9
0.08
N
0
0
N
2/9
0.0003
N
4/9
0.0005
3/4
3/9
1/4
4/9
<S>
M
3/4
0.19
1/4
M
1/4
0.006
M
0
0
M
0
0
<E>
0.000225
1
3/4
V
0
0
V
0
0
V
1/4
0.00113
V
0
0

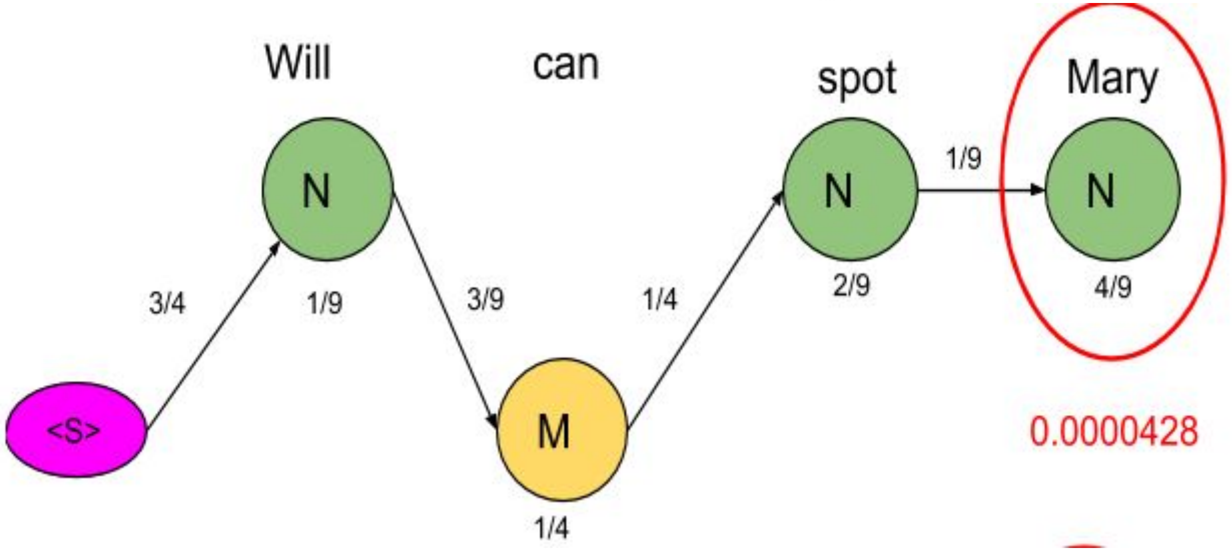
The next step is to delete all the vertices and edges with probability zero, also the vertices which do not lead to the endpoint are removed. Also, we will mention-

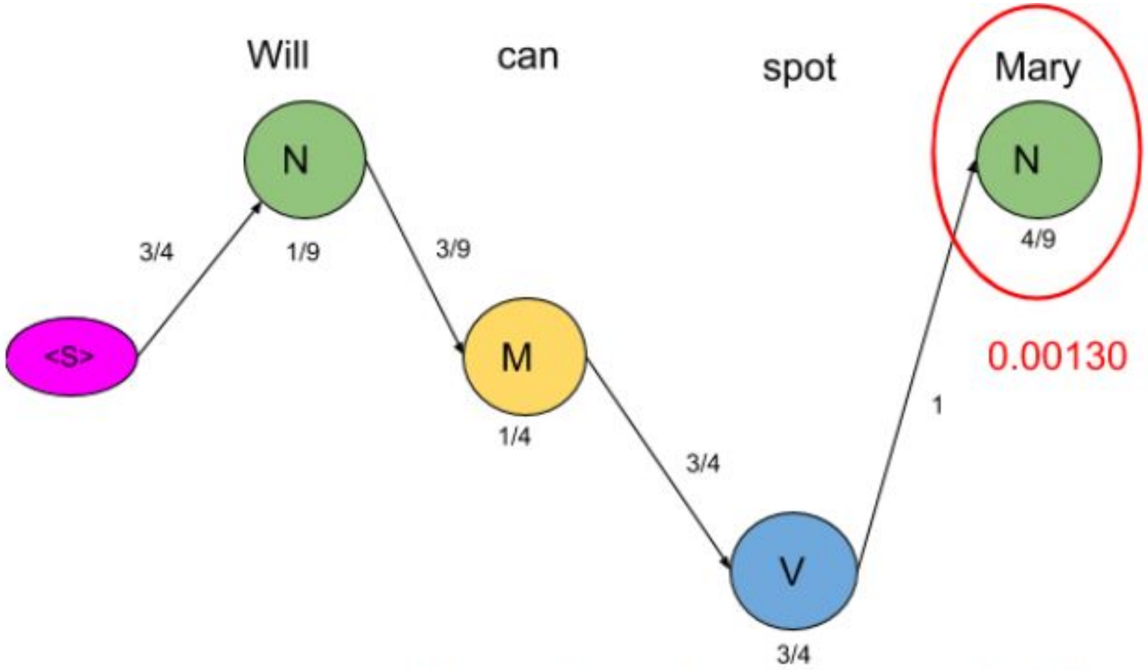




After applying Viterbi algorithm the model tags the sentence as following

- Will as a noun
- Can as a model
- Spot as a verb
- Mary as a noun





Step 1: Assign correct POS Tag to training corpus

Step 2: Creation of Emission Probability Matrix

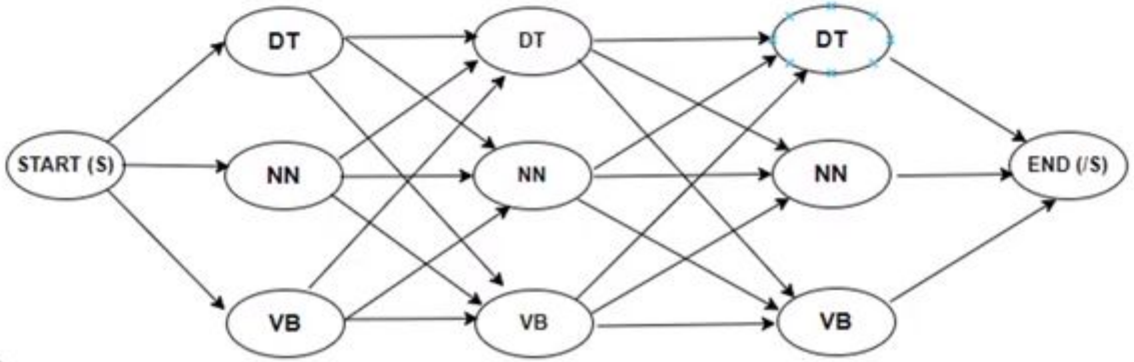
	DT	NN	VB
That	0.40	0.00	0.00
Girl	0.00	0.015	0.0031
Smiles	0.00	0.0004	0.20

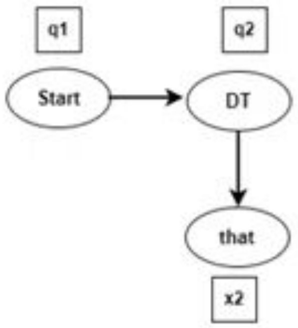
Step 3: Creation of State Transition Probability Matrix

	DT	NN	VB
<S>	0.50	0.40	0.1
DT	0.01	0.99	0.00
NN	0.30	0.30	0.40
VB	0.40	0.40	0.20

Test Data <S> That girl smiles </S>

That	Girl	Smiles
DT	DT	DT
NN	NN	NN
VB	VB	VB
3×3×3=27 ways		

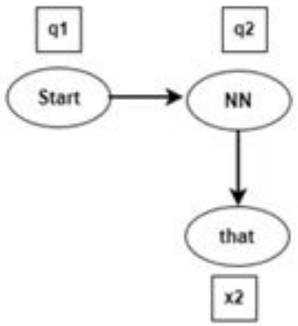




$$P(q2|x2, q1) = P(x2|q2) \times P(q2|q1)$$

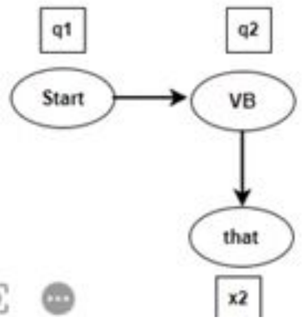
$$P(DT|that, <S>) = P(that|DT) \times P(DT|<S>)$$

$$\mathbf{P(DT|that, <S>) = 0.40 \times 0.50 = 0.2}$$



$$P(NN|that, <S>) = P(that|NN) \times P(NN|<S>)$$

$$\mathbf{P(NN|that, <S>) = 0.00 \times 0.40 = 0}$$



$$P(VB|that, <S>) = P(that|VB) \times P(VB|<S>)$$

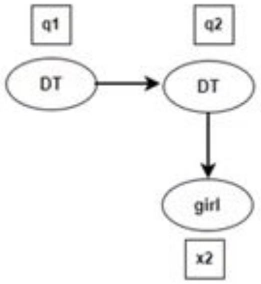
$$\mathbf{P(VB|that, <S>) = 0.00 \times 0.1 = 0}$$

Probability of “that “ word is large for DT Part of Speech Tag

	DT	NN	VB
That	0.40	0.00	0.00
Girl	0.00	0.015	0.0031
Smiles	0.00	0.0004	0.20

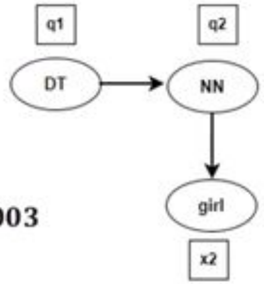
	DT	NN	VB
<S>	0.50	0.40	0.1
DT	0.01	0.99	0.00
NN	0.30	0.30	0.40
VB	0.40	0.40	0.20

Previous Probability is 0.2



$$P(DT|girl, DT) = P(girl|DT \times P(DT|DT)$$

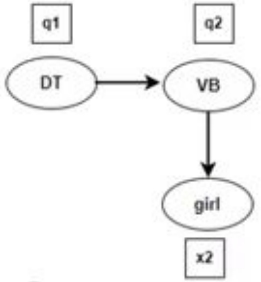
$$P(DT|girl, DT) = 0 \times 0.01 = 0$$



$$P(NN|girl, DT) = P(girl|NN) \times P(NN|DT)$$

$$P(NN|girl, DT) = 0.015 \times 0.99 = 0.01485$$

$$= 0.01458 \times 0.2 = 0.00297 \approx 0.003$$



$$P(VB|girl, DT) = P(girl|VB) \times P(VB|DT)$$

$$P(VB|girl, DT) = 0.0031 \times 0.00 = 0$$

Result: Girl is noun

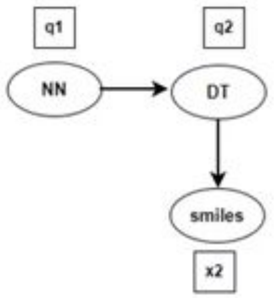
	DT	NN	VB
That	0.40	0.00	0.00
Girl	0.00	0.015	0.0031
Smiles	0.00	0.0004	0.20

	DT	NN	VB
<S>	0.50	0.40	0.1
DT	0.01	0.99	0.00
NN	0.30	0.30	0.40
VB	0.40	0.40	0.20

	DT	NN	VB
That	0.40	0.00	0.00
Girl	0.00	0.015	0.0031
Smiles	0.00	0.0004	0.20

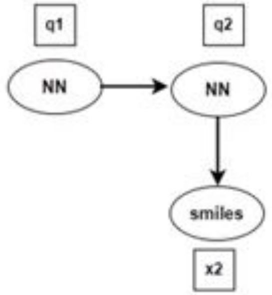
	DT	NN	VB
<S>	0.50	0.40	0.1
DT	0.01	0.99	0.00
NN	0.30	0.30	0.40
VB	0.40	0.40	0.20

Previous Probability is 0.003



$$P(DT|smiles, NN) = P(smiles|DT) \times P(DT|NN)$$

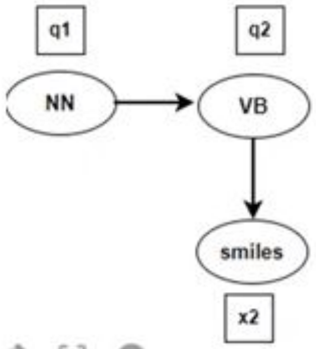
$$P(DT|smiles, NN) = 0 \times 0.30 = 0$$



$$P(NN|smiles, NN) = P(smiles|NN) \times P(NN|NN)$$

$$P(NN|smiles, NN) = 0.0004 \times 0.30 = 0.00012$$

$$= 0.00012 \times 0.003 = 0.00000036$$



$$P(VB|smiles, NN) = P(smiles|VB) \times P(VB|NN)$$

$$P(VB|smiles, NN) = 0.2 \times 0.4 = 0.08$$

$$= 0.08 \times 0.003 = 0.00024$$

Dr. Varsha Patil

Result: Smiles is verb

Result of POS tagging after applying HMM model

<S>	The	girl	smiles	</S>
	Det.	Noun	Verb	

Step 1: Assign correct POS Tag to training corpus

Step 2: Creation of Emission Probability Matrix

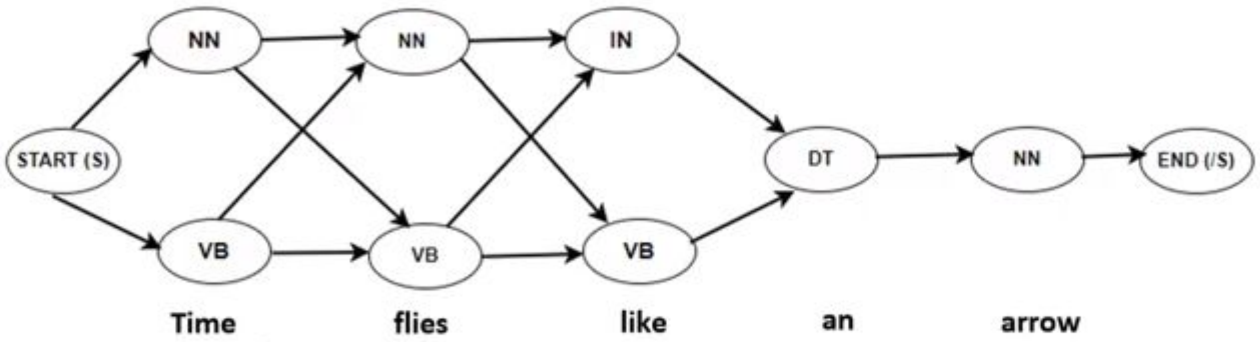
	Time	Flies	like	an	arrow
VB	0.1	0.2	0.2	0	0
NN	0.1	0.1	0	0	0.1
IN	0	0	0.25	0	0
DT	0	0	0	0.5	0

Step 3: Creation of State Transition Probability Matrix

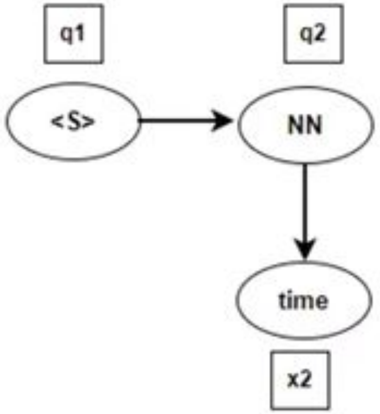
	VB	NN	IN	DT	<E>
<S>	0.2	0.8	0	0	0
VB	0	0.3	0.2	0.5	0
NN	0.4	0.5	0.1	0	0
IN	0	0.75	0	0.25	0
DT	0	1	0	0	0

Test Data **<S> Time flies like an arraow </S>**

Time	**flies**	**like**	**an**	**arrow**
NN	NN	IN	DT	NN
VB	VB	NN		
2×2×2×1×1=8 ways				

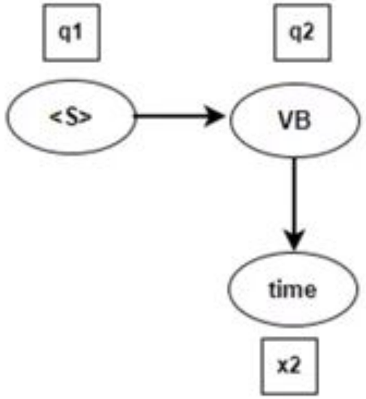


$$P(q2|x2, q1) = P(x2|q2) \times P(q2|q1)$$



$$P(NN|time, <S>) = P(time|NN) \times P(NN|<S>)$$

$$P(NN|time, <S>) = 0.1 \times 0.8 = 0.08$$



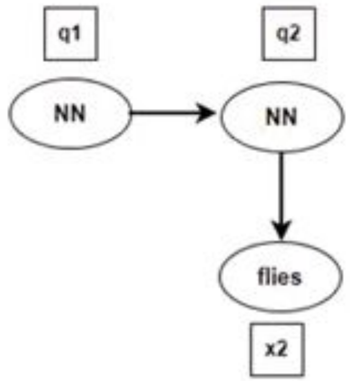
$$P(VB|time, <S>) = P(time|VB) \times P(VB|<S>)$$

$$P(VB|time, <S>) = 0.1 \times 0.2 = 0.02$$

$$0.08 > 0.02$$

Probability of "time " word is large for NN Part of Speech Tag

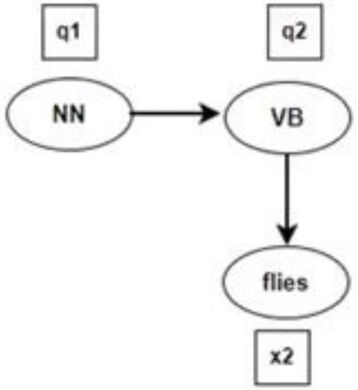
Previous Probability is 0.08



$$P(NN|flies, NN) = P(flies|NN) \times P(NN|NN)$$

$$P(NN|flies, NN) = 0.1 \times 0.5 = 0.05$$

$$= 0.05 \times 0.08 = 0.0040$$



$$P(VB|flies, NN) = P(flies|VB) \times P(VB|NN)$$

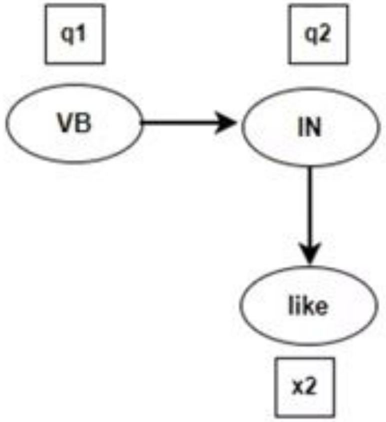
$$P(VB|flies, NN) = 0.2 \times 0.4 = 0.08$$

$$= 0.08 \times 0.08 = 0.0064$$

$0.0064 > 0.0040$

Result: flies is verb

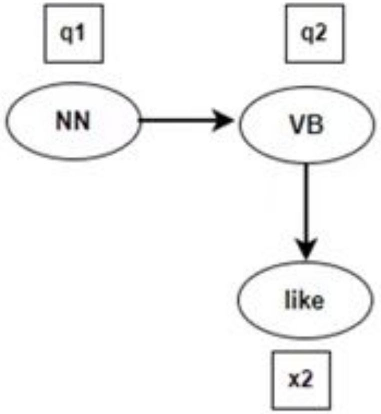
Previous Probability is 0.0064



$$P(\text{In}|\text{like}, \text{VB}) = P(\text{like}|\text{In}) \times P(\text{In}|\text{VB})$$

$$P(\text{IN}|\text{like}, \text{VB}) = 0.25 \times 0.2 = 0.05$$

$$= 0.05 \times 0.0064 = 0.00032$$



$$P(\text{VB}|\text{like}, \text{VB}) = P(\text{like}|\text{VB}) \times P(\text{VB}|\text{VB})$$

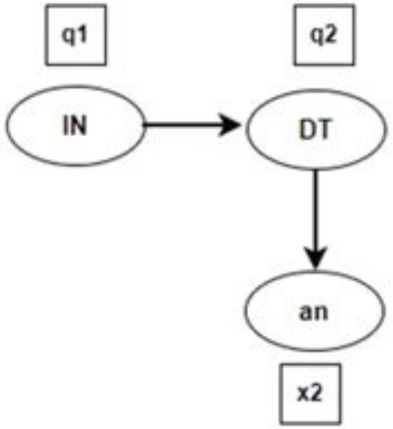
$$P(\text{IN}|\text{like}, \text{VB}) = 0.2 \times 0 = 0$$

$$= 0.2 \times 0.01 = 0.002$$

$$= 0.002 \times 0.0064 = 0.0000128$$

Result: like is preposition

Previous Probability is 0.00032

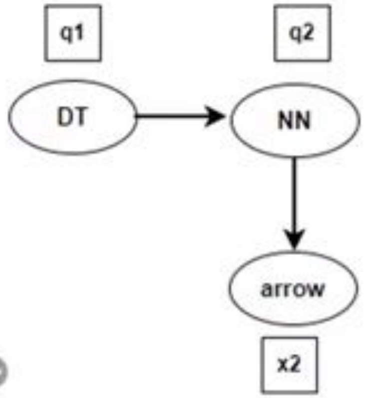


$$P(DT|a, IN) = P(a|DT) \times P(DT|IN)$$

$$P(DT|a, IN) = 0.5 \times 0.25 = 0.125$$
$$= 0.125 \times 0.00032 = 0.00004$$

Result: an is DT

Previous Probability is 0.00004



$$P(NN|arrow, DT) = P(arrow|NN) \times P(NN|DT)$$
$$= 0.1 \times 1 = 0.1$$
$$= 0.1 \times 0.00004 = 0.000004$$

Result:arrow is noun

Test Data <S> Time flies like an arraow </S>

<S>	Time	flies	like	an	arrow	</S>
	NN	VB	IN	DT	NN	