Part-of-speech Tagging

Bal Krishna Nyaupane
Assistant Professor
Department of Electronics and Computer Engineering
Paschimanchal Campus, IOE
bkn@wrc.edu.np

Part of speech Tagging

- Annotate each word in a sentence with a part-of-speech marker.
- Lowest level of syntactic analysis.
- John saw the saw and decided to take it to the table.
- NNP VBD DT NN CC VBD TO VB PRP IN DT NN
- Useful for subsequent syntactic parsing and word sense disambiguation.

- Original Brown corpus used a large set of 87 POS tags.
- Most common in NLP today is the Penn Treebank set of 45 tags.
  - *Tagset used in these slides.*
  - *Reduced from the Brown set for use in the context of a parsed corpus (i.e. treebank).*
- The C5 tagset used for the British National Corpus (BNC) has 61 tags.

English Part of Speech

- Noun (person, place or thing)
  - Singular (NN): dog, fork
  - Plural (NNS): dogs, forks
  - Proper (NNP, NNPS): John, Springfields
  - Personal pronoun (PRP): I, you, he, she, it
  - Wh-pronoun (WP): who, what
- Verb (actions and processes)
  - Base, infinitive (VB): eat
  - Past tense (VBD): ate
  - Gerund (VBG): eating
  - Past participle (VBN): eaten
  - Non 3rd person singular present tense (VBP): eat
  - 3rd person singular present tense: (VBZ): eats
  - Modal (MD): should, can
  - To (TO): to (to eat)
- Adjective (modify nouns)
  - Basic (JJ): red, tall
  - Comparative (JJR): redder, taller
  - Superlative (JJS): reddest, tallest
- Adverb (modify verbs)
  - Basic (RB): quickly
  - Comparative (RBR): quicker
  - Superlative (RBS): quickest
- Preposition (IN): on, in, by, to, with
- Determiner:
  - Basic (DT) a, an, the
  - WH-determiner (WDT): which, that
- Coordinating Conjunction (CC): and, but, or,
- Particle (RP): off (took off), up (put up)

POS tags used in the Penn Treebank Project

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

A Universal Part-of-Speech Tagset

Tag	Meaning	English Examples
ADJ	adjective	*new, good, high, special, big, local*
ADP	adposition	*on, of, at, with, by, into, under*
ADV	adverb	*really, already, still, early, now*
CONJ	conjunction	*and, or, but, if, while, although*
DET	determiner, article	*the, a, some, most, every, no, which*
NOUN	noun	*year, home, costs, time, Africa*
NUM	numeral	*twenty-four, fourth, 1991, 14:24*
PRT	particle	*at, on, out, over per, that, up, with*
PRON	pronoun	*he, their, her, its, my, I, us*
VERB	verb	*is, say, told, given, playing, would*
.	punctuation marks	*. , ; !*
X	other	*ersatz, esprit, dummo, gr8, univeristy*

Ambiguity in POS Tagging

- "Like" can be a verb or a preposition
  - *I like/VBP candy.*
  - *Time flies like/IN an arrow.*
- "Around" can be a preposition, particle, or adverb
  - *I bought it at the shop around/IN the corner.*
  - *I never got around/RP to getting a car.*
  - *A new Prius costs around/RB $25K.*

What is Part of speech tagging?

- Part-of-speech tagging is the ***process of converting a sentence***, in the form of a list of words, into a list of tuples, where each tuple is of the form (word, tag). ***The tag is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on***.

- Without the part-of-speech tags, a ***chunker*** cannot know how to extract phrases from a sentence. But with part-of-speech tags, you can tell a ***chunker*** how to identify phrases based on tag patterns.
- ***Chunk extraction*** is the process of extracting short phrases from a part-of-speech tagged sentence.

What is Part of speech tagging?

- ***Wikipedia definition*** : in corpus linguistics, POS tagging or POST, also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context — i.e., its relationship with adjacent and related words in a phrase, sentence, or paragraph.

- A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc.

Why Part-of-Speech tagging?

- ***Text to Speech Conversion***
  - Let us look at the following sentence:

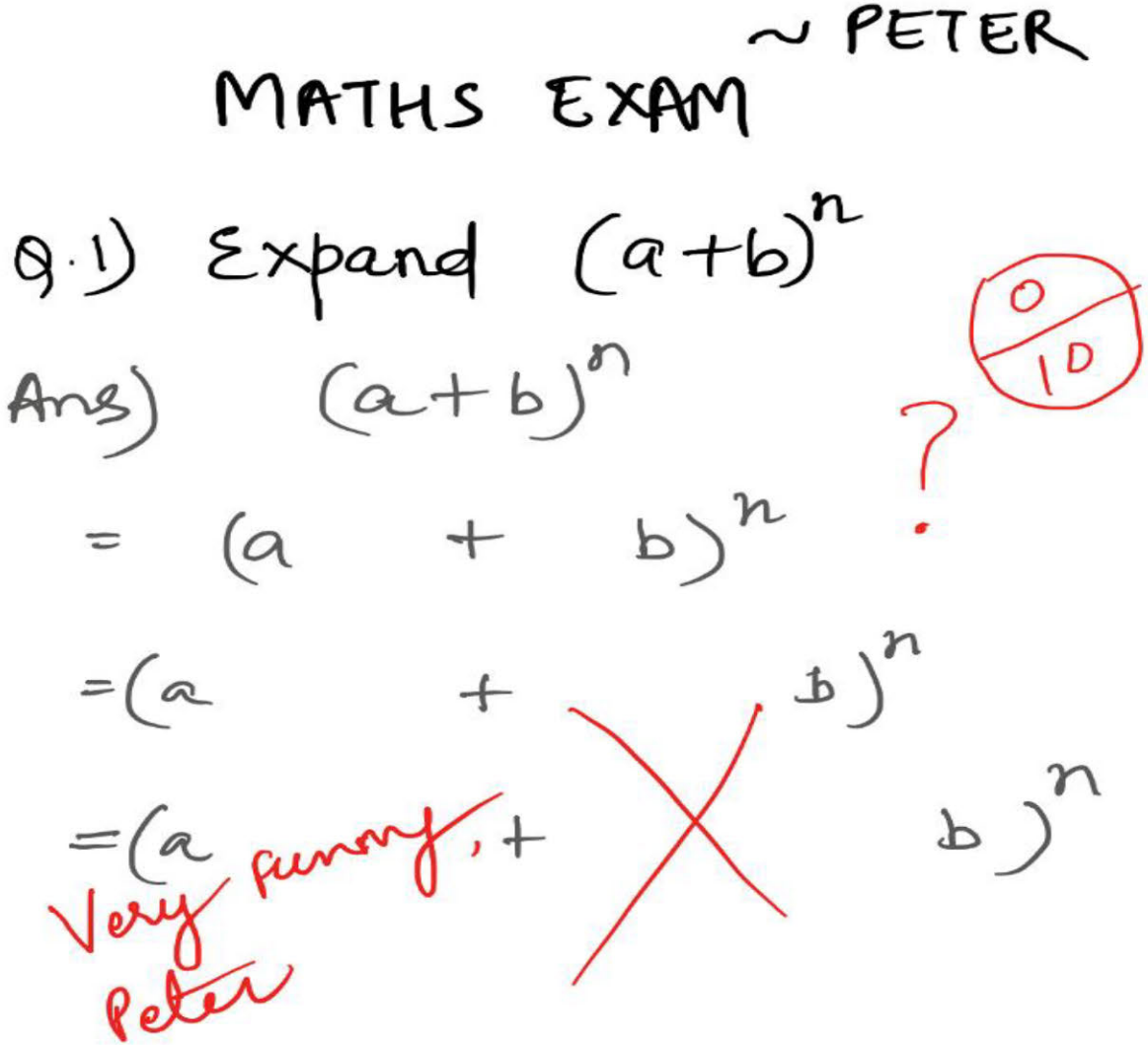
    "They refuse to permit us to obtain the refuse permit."
  - The word refuse is being used twice in this sentence and has two different meanings here. Refuse "/rɪˈfjuːz/" is a verb meaning "deny," while Refuse "/ˈrɛfjuːs/" is a noun meaning "trash" (that is, they are not homophones).

  - Thus, we need to know which word is being used in order to pronounce the text correctly. (For this reason, text-to-speech systems usually perform POS-tagging.)

Why Part-of-Speech tagging?

- ***Word Sense Disambiguation:*** Word-sense disambiguation (WSD) is identifying which sense of a word (that is, which meaning) is used in a sentence, when the word has multiple meanings.

- The word ***"expand"***



POS Tagging Process

- Usually assume a separate initial tokenization process that separates and/or disambiguates punctuation, including detecting sentence boundaries.

- Degree of ambiguity in English (based on Brown corpus)
  - ***11.5% of word types are ambiguous.***
  - ***40% of word tokens are ambiguous.***
- Average POS tagging disagreement amongst expert human judges for the Penn treebank was 3.5%
  - ***Based on correcting the output of an initial automated tagger, which was deemed to be more accurate than tagging from scratch.***
- **Baseline:** Picking the most frequent tag for each specific word type gives about 90% accuracy
  - ***93.7% if use model for unknown words for Penn Treebank tagset.***

Types of POS taggers: Rule-Based POS Taggers

- One of the ***oldest techniques*** of tagging is rule-based POS tagging.
- Rule-based taggers ***use dictionary or lexicon*** for getting possible tags for tagging each word.

- Typical rule-based approaches ***use contextual information to assign tags to unknown or ambiguous words***. 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 an article, *then the word in question must be a noun*. This information is coded in the form of rules.

Types of POS taggers: Rule-Based POS Taggers

- ***Example of a rule:*** If an ambiguous/unknown word X is preceded by a determiner and followed by a noun, tag it as an adjective.

- The Brill's tagger is a rule-based tagger that goes through the training data and finds out the set of tagging rules that best define the data and minimize POS tagging errors.

- The most important point to note here about Brill's tagger is that the rules are not hand-crafted, but are instead found out using the corpus provided. The only feature engineering required is a set of rule templates that the model can use to come up with new features.

Types of POS taggers: Rule-Based POS Taggers

- We can also understand Rule-based POS tagging by its two-stage architecture -
  - ***First stage:*** In the first stage, it uses a dictionary to assign each word a list of potential parts-of-speech.
  - ***Second stage:*** In the second stage, it uses large lists of hand-written disambiguation rules to sort down the list to a single part-of-speech for each word.

Types of POS taggers: Stochastic POS Tagging

- The model that includes frequency or probability (statistics) can be called stochastic.
- Any model which ***somehow incorporates frequency or probability*** may be properly labelled stochastic.
- **Word Frequency Approach:**
  - It disambiguate ***words based solely on the probability that a word occurs with a particular tag***.
  - We can also say that the tag encountered most frequently in the training set with the word is the one assigned to an ambiguous instance of that word.
  - The main issue with this approach is that it may yield inadmissible sequence of tags.

Types of POS taggers: Stochastic POS Tagging

- **Tag Sequence Probabilities**
  - The tagger calculates the probability of a given sequence of tags occurring.
  - This is sometimes referred to as the n-gram approach, referring to the fact that the best tag for a given word is determined by the probability that it occurs with the n previous tags.
  - This approach makes much more sense than the one defined before, because it considers the tags for individual words based on context.

Markov chains

- If the future states of a process are independent of the past and depend only on the present , the process is called a Markov process. A Markov Chain is a random process with the property that the next state depends only on the current state.

- Markov chains is a mathematical tools for statistical modeling in modern applied mathematics, information science.

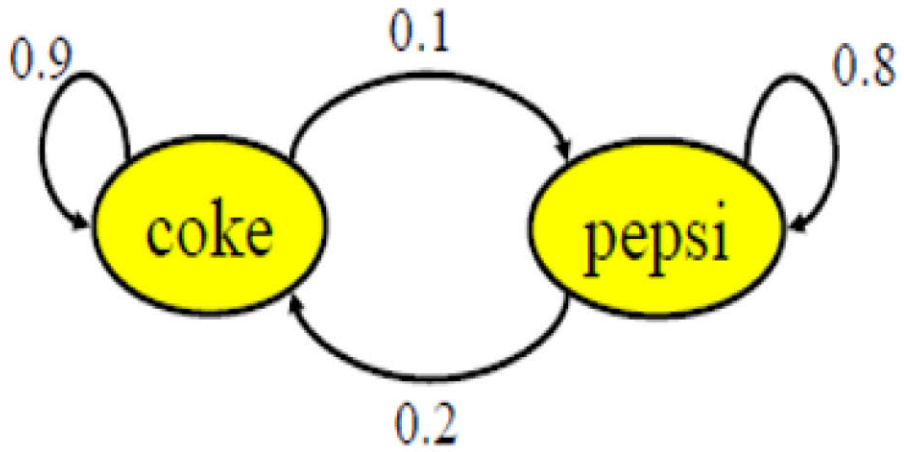
- Markov chains are used to analyze trends and predict the future. (Weather, stock market, genetics, product success, etc.)

Markov Chain Model: Transition Matrix

- A Transition matrix has following features:
  1) It is square, since all possible states must be used both as rows and as columns.
  2) All entries are between 0 and 1.
  3) The sum of entries of any row must be 1, since the number in the row give the probability of changing from one state at the left to one of state across the top.

transition matrix:

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$



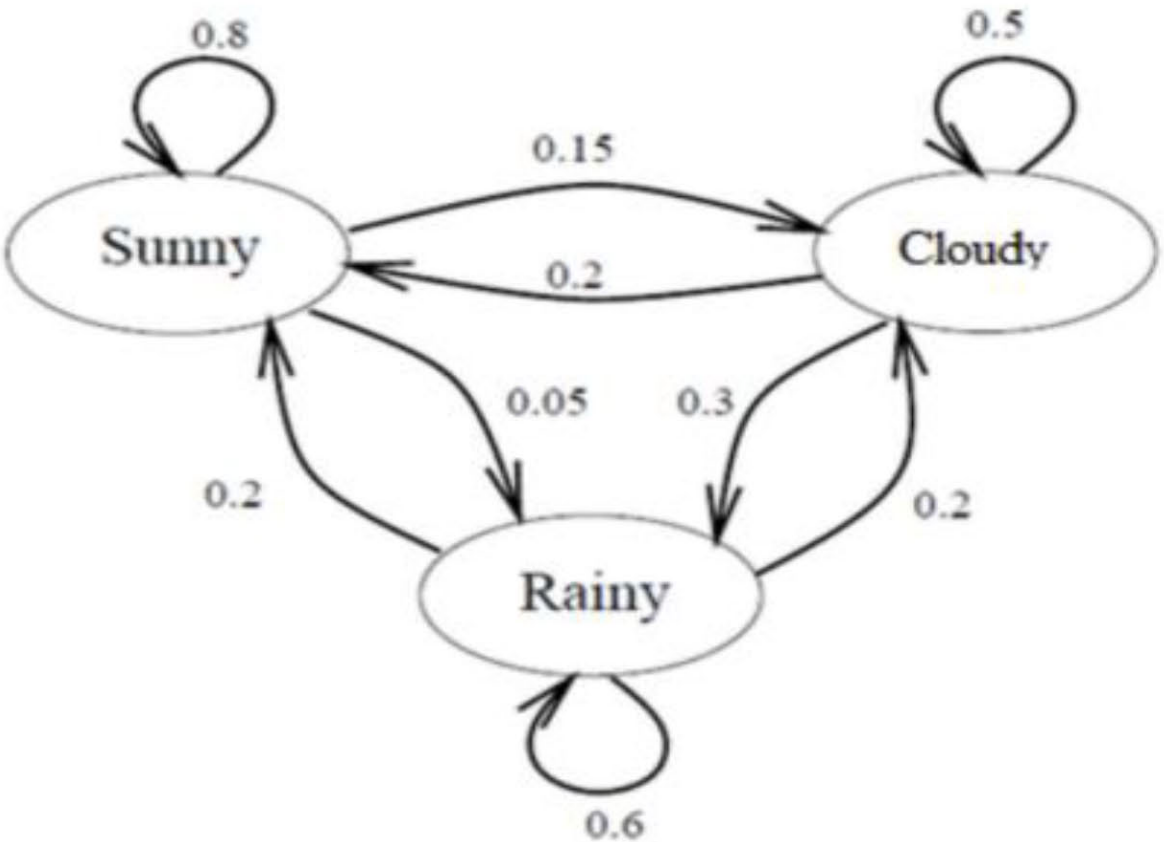
Markov Chain Example

- Exercise 2: Assume that yesterday's weather was Rainy, and today is Cloudy, what is the probability that tomorrow will be Sunny?

$$P(q_3|q_1, q_2) = P(q_3|q_2)$$

$$= P(Sunny|Cloudy)$$

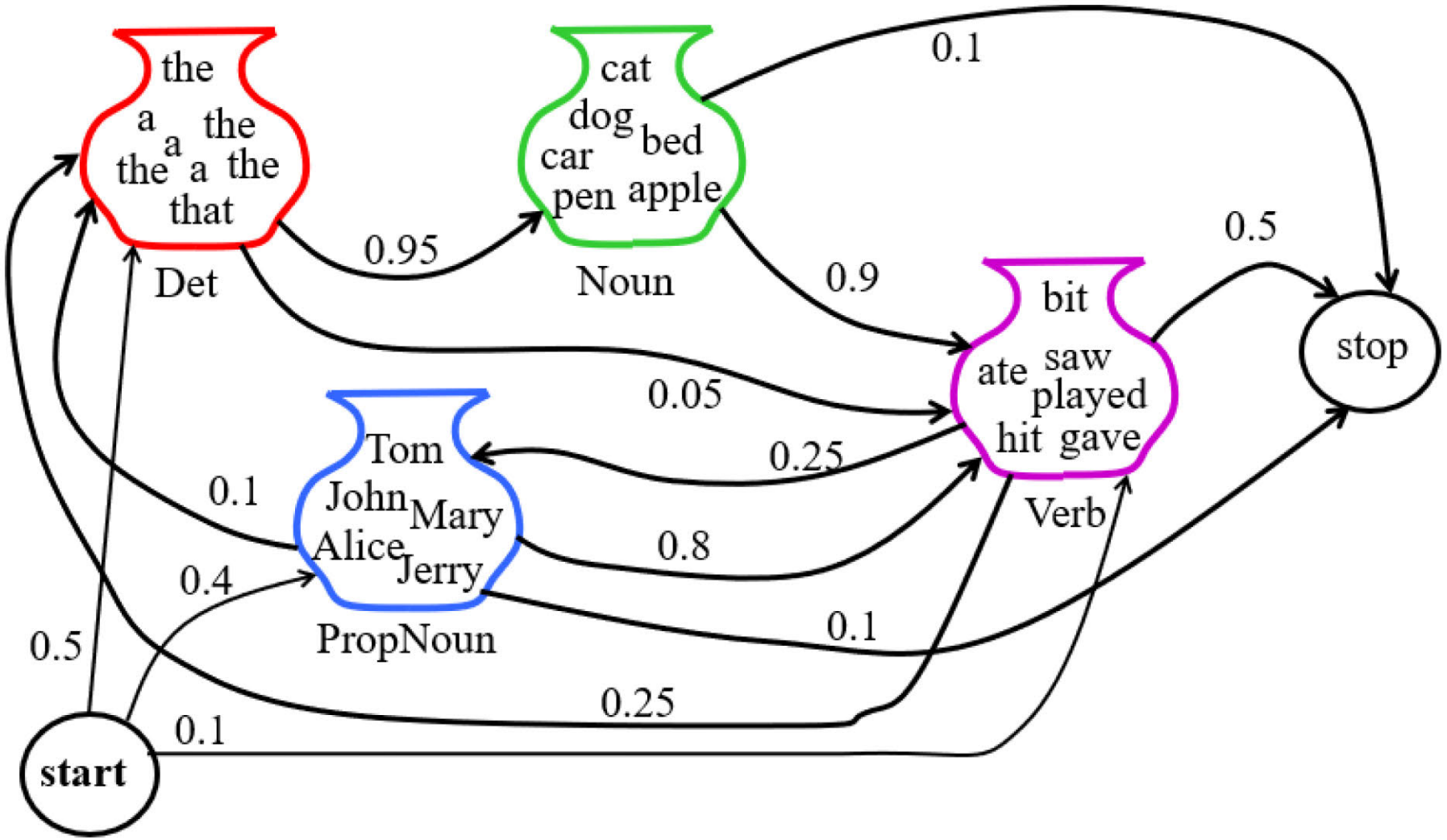
$$= 0.2$$



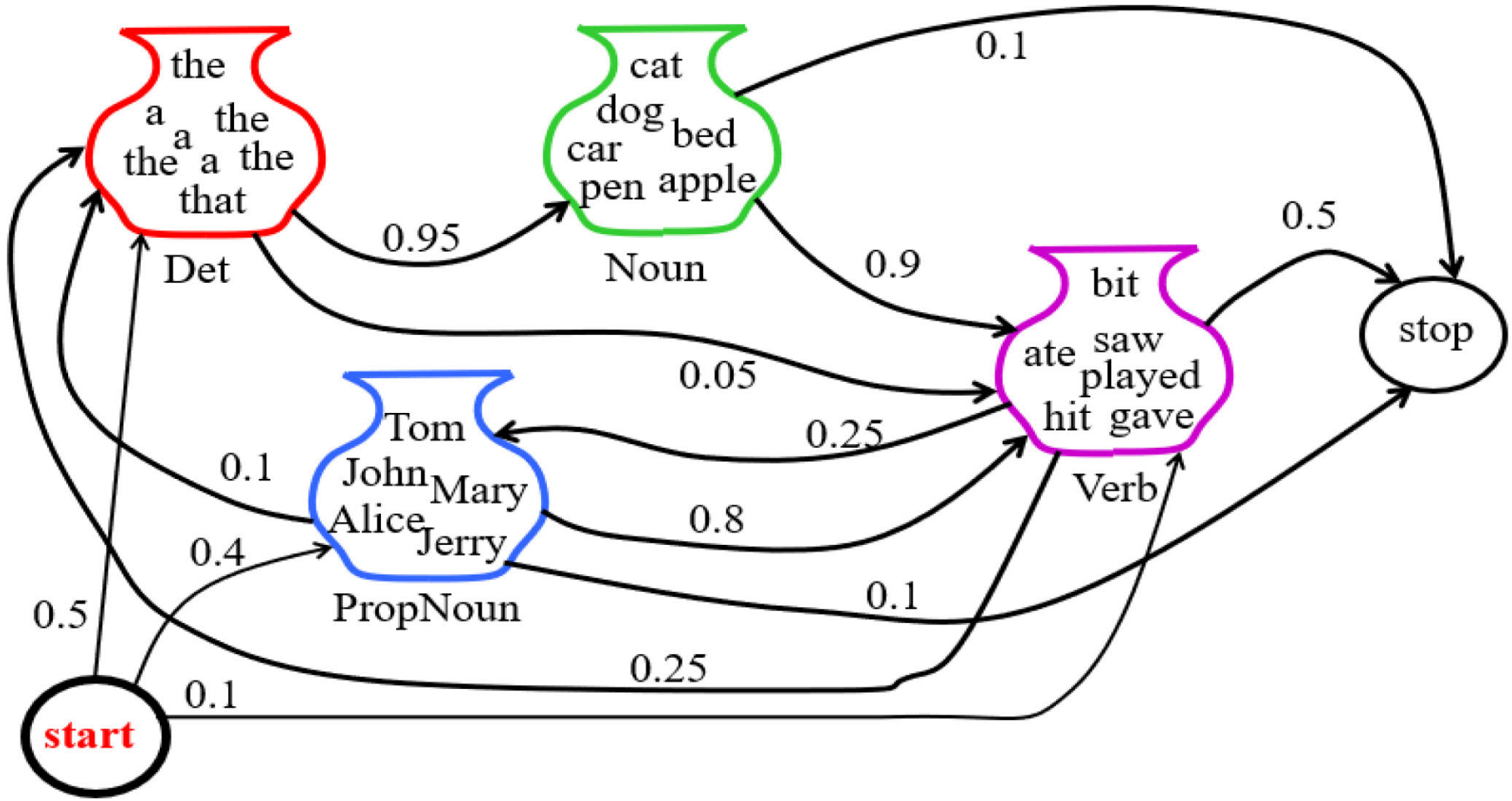
Hidden Markov Model (HMM) POS Tagging

- HMMs are based on Markov chains.
- Combines two approaches: tag sequence probabilities and word frequency measurements.
- The POS tagging process is the process of finding the sequence of tags which is most likely to have generated a given word sequence.
- We can model this POS process by using a Hidden Markov Model (HMM), where ***tags are the hidden states*** that produced the observable output, i.e., the words.
- HMMs are used in reinforcement learning and have wide applications in *cryptography, text recognition, speech recognition, bioinformatics, and many more*.

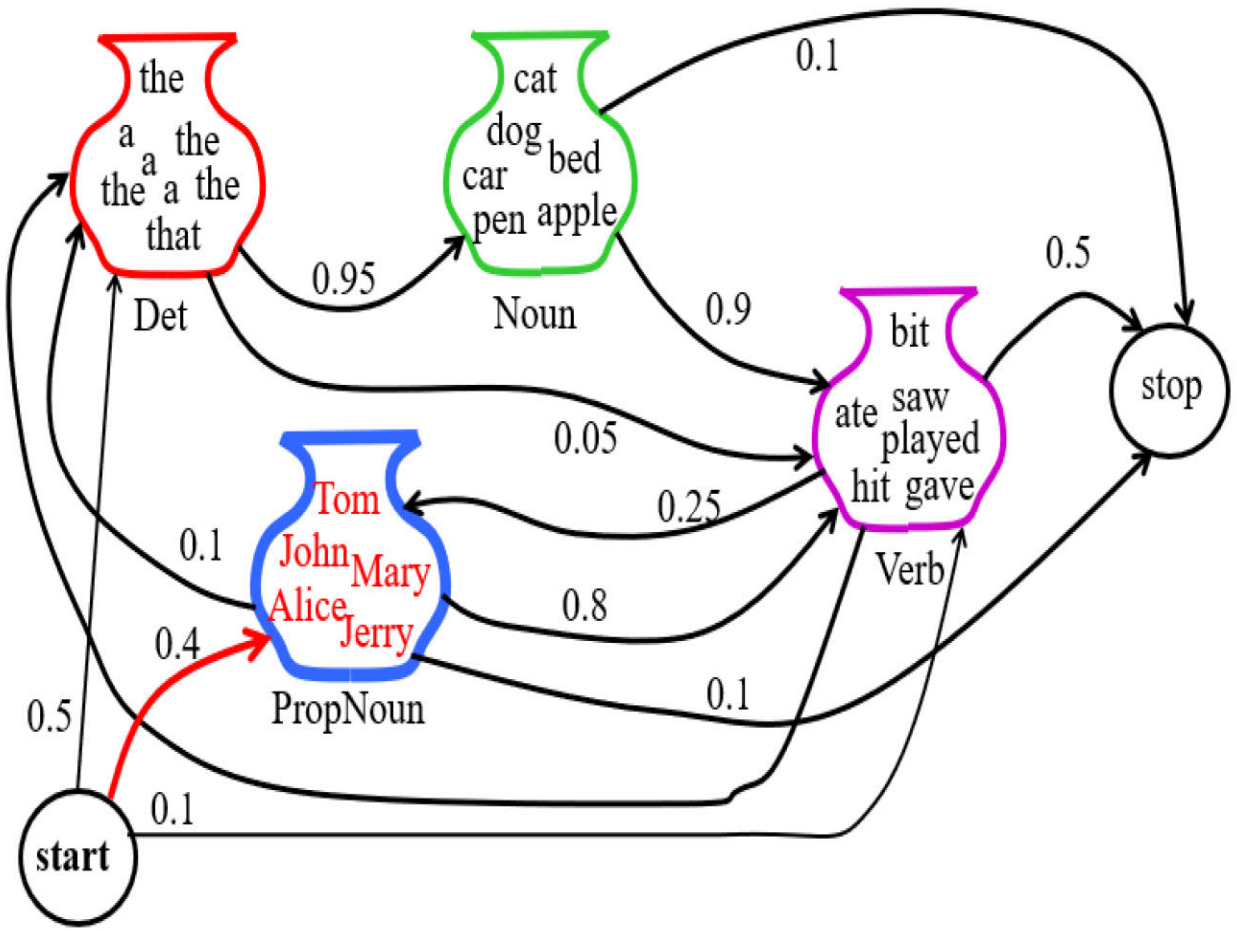
Sample HMM for POS

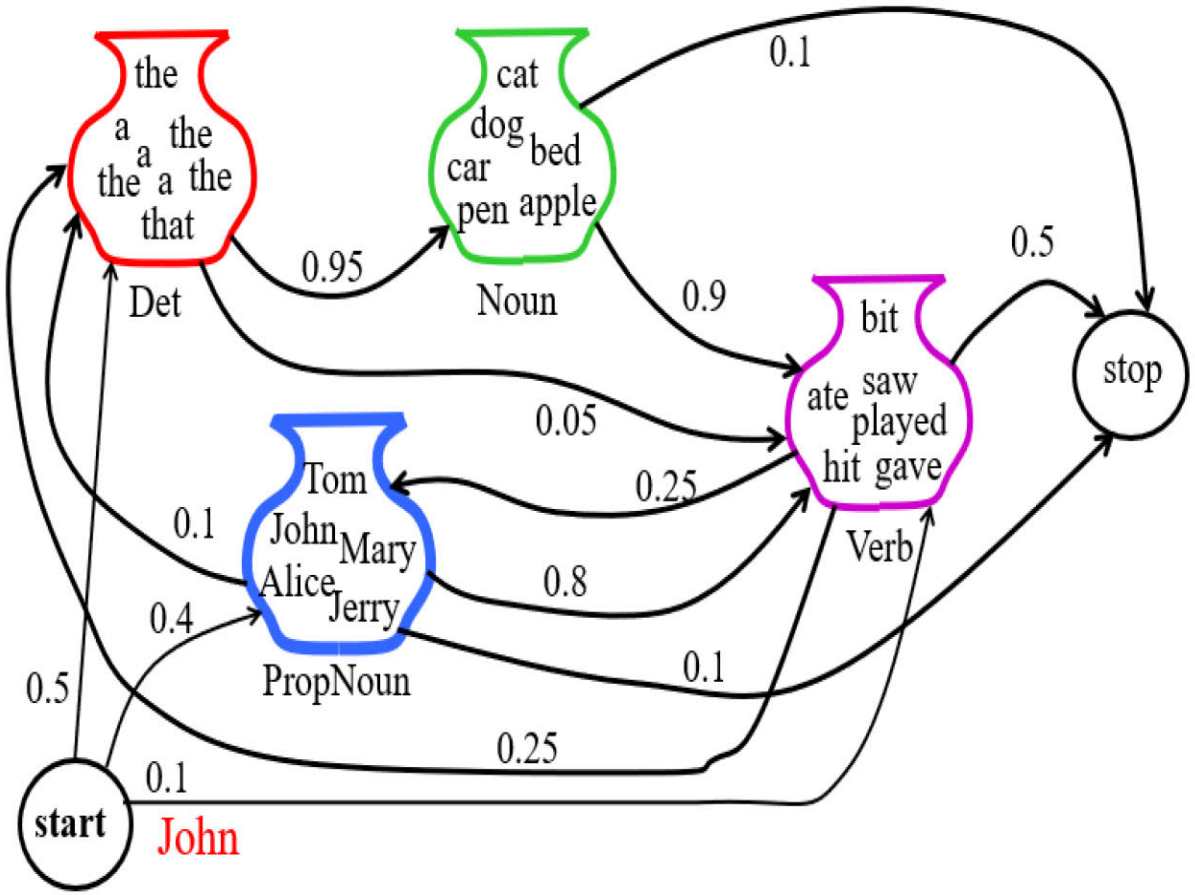


Sample HMM Generation

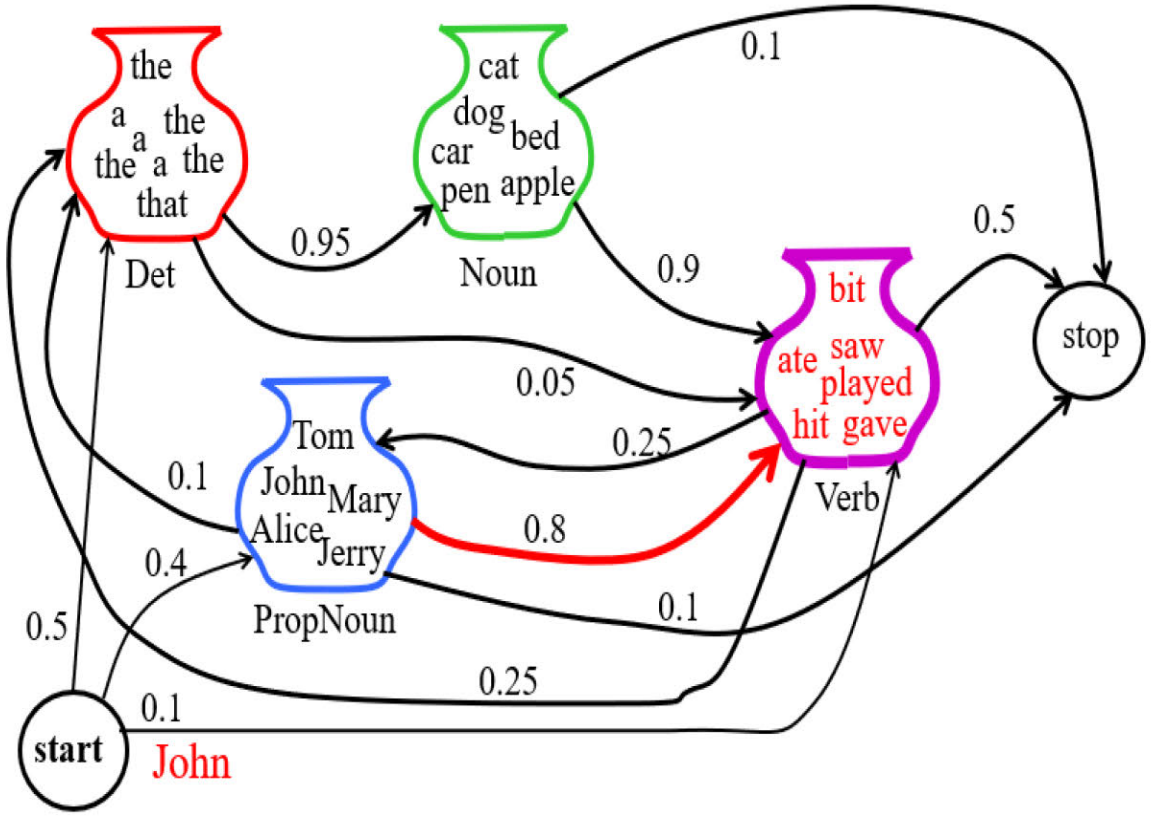


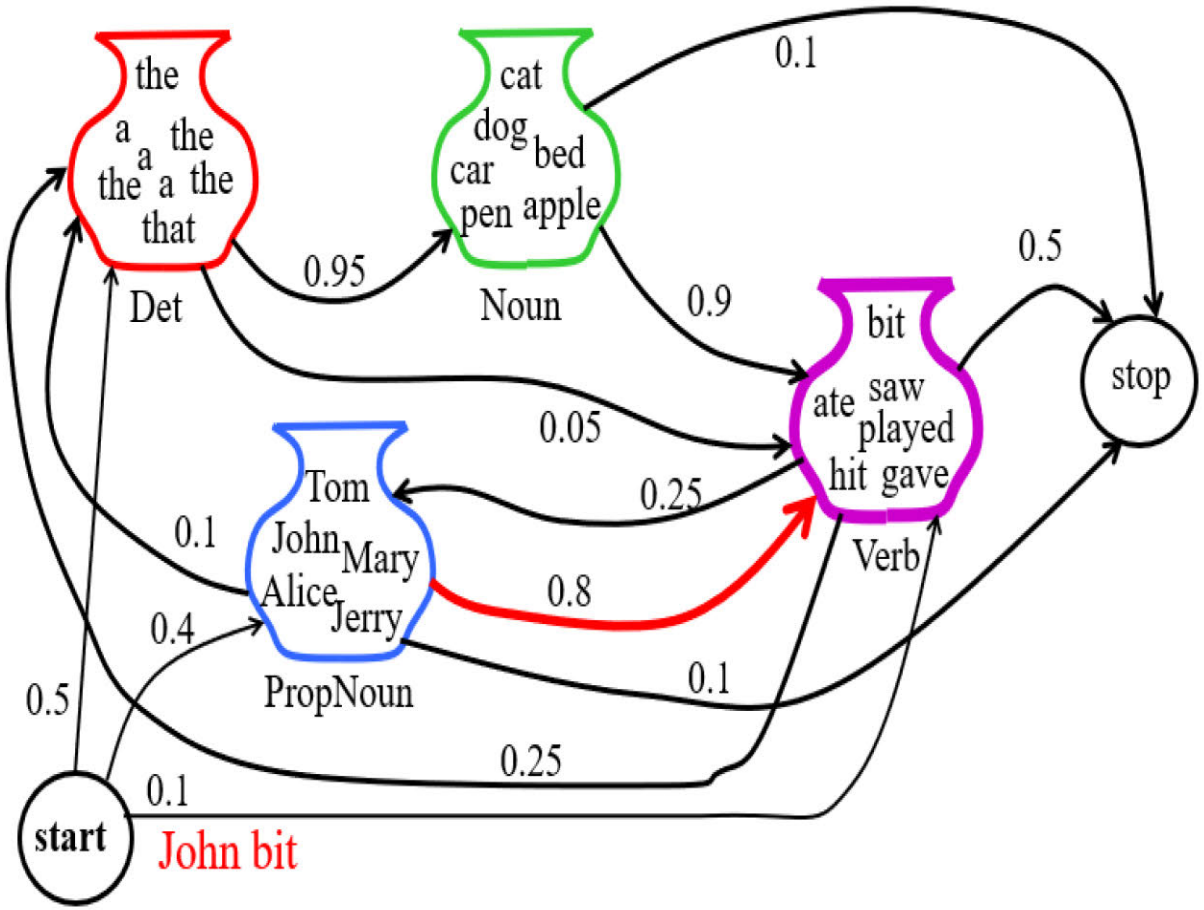
Sample HMM Generation



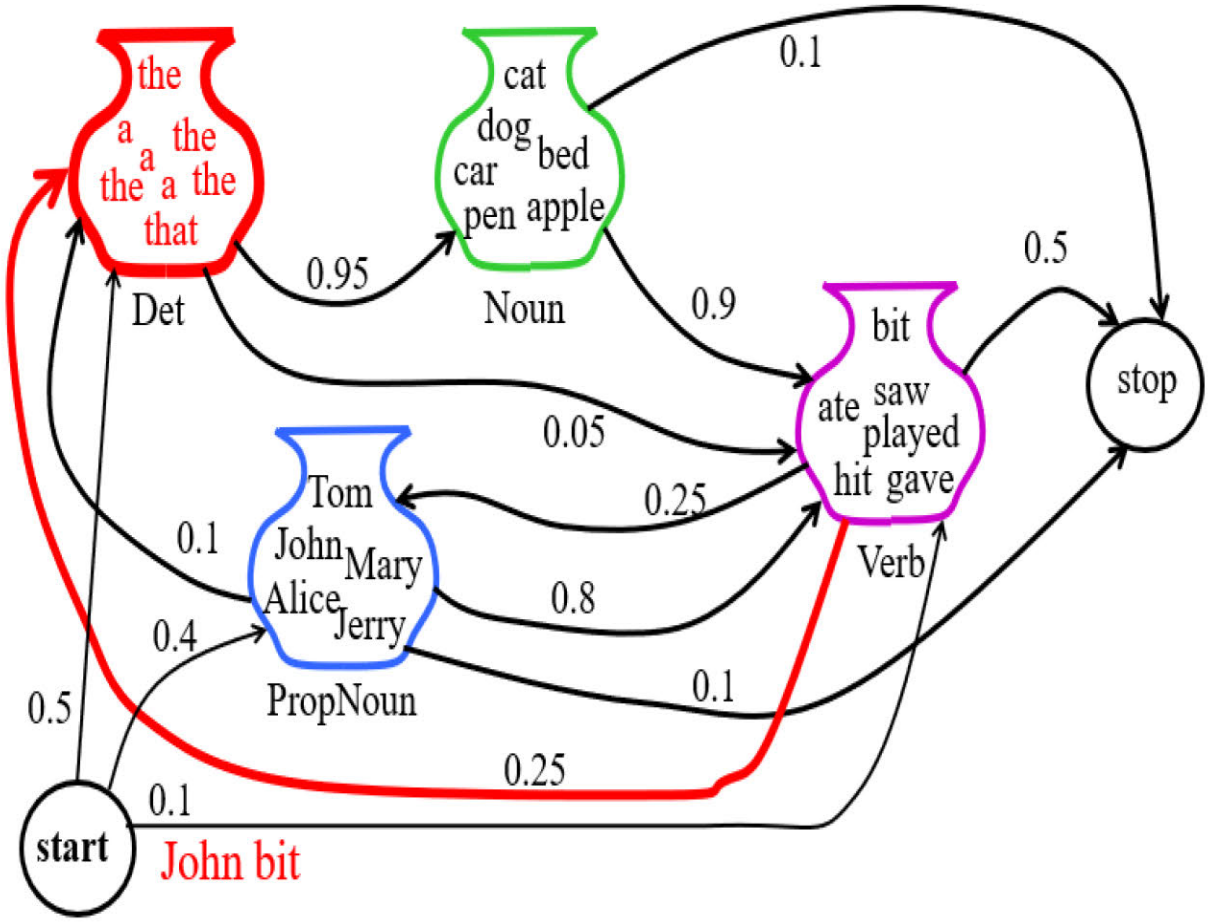


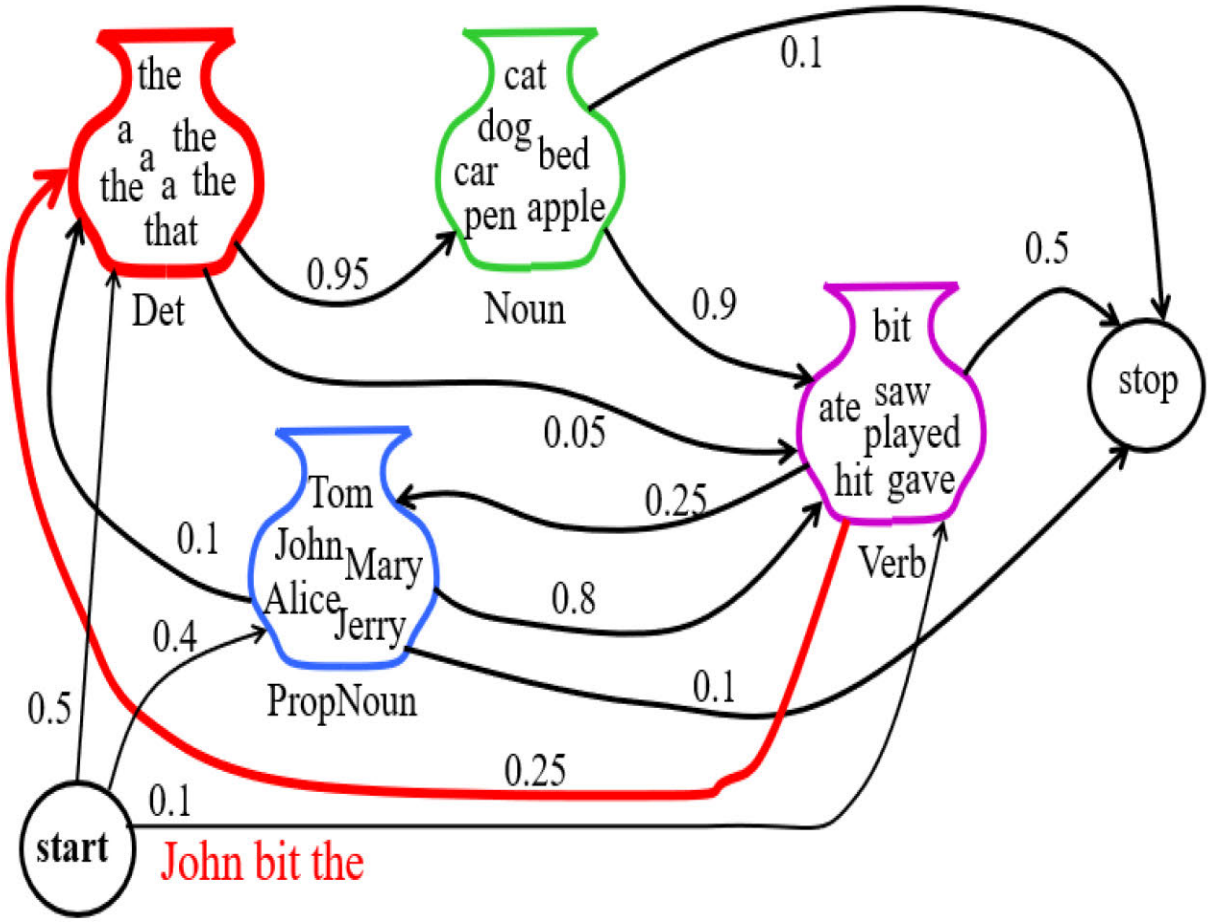
Sample HMM Generation



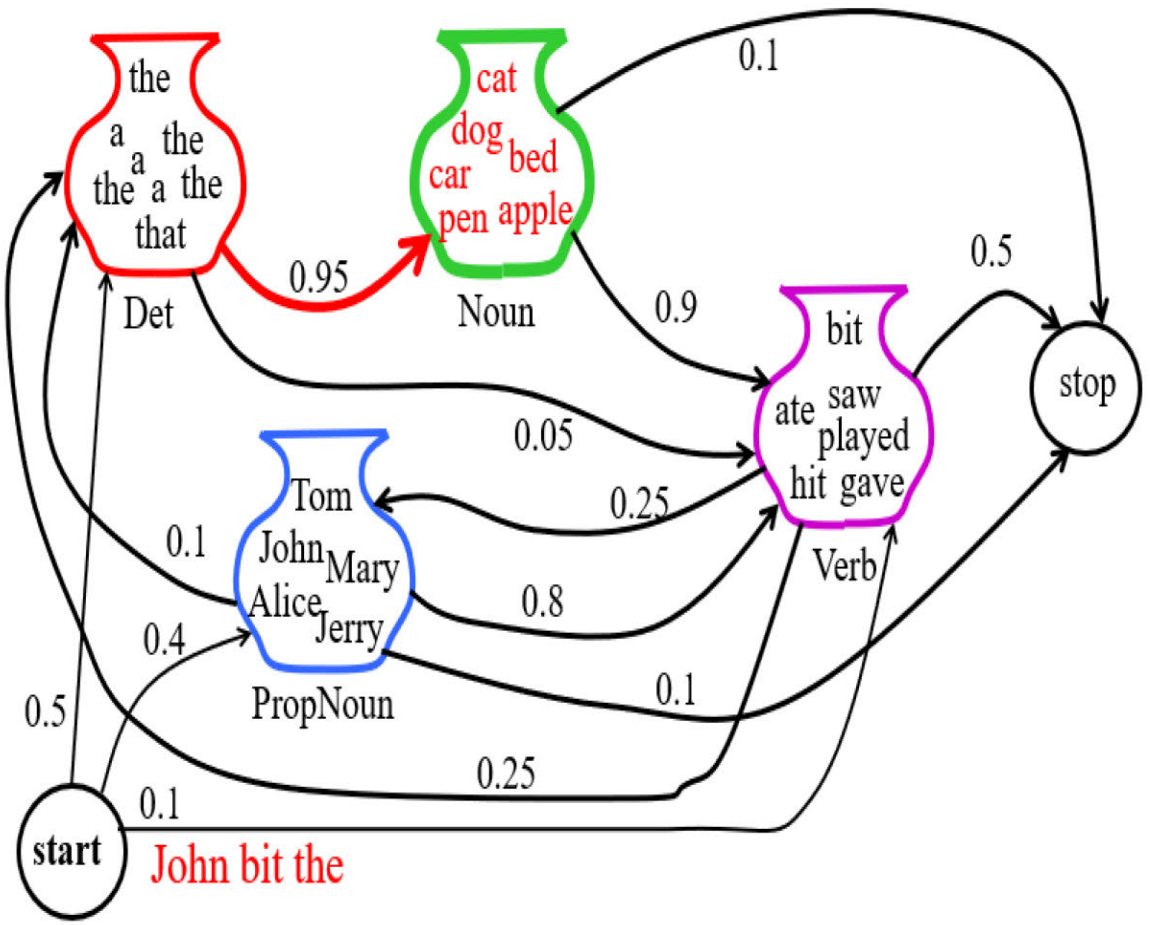


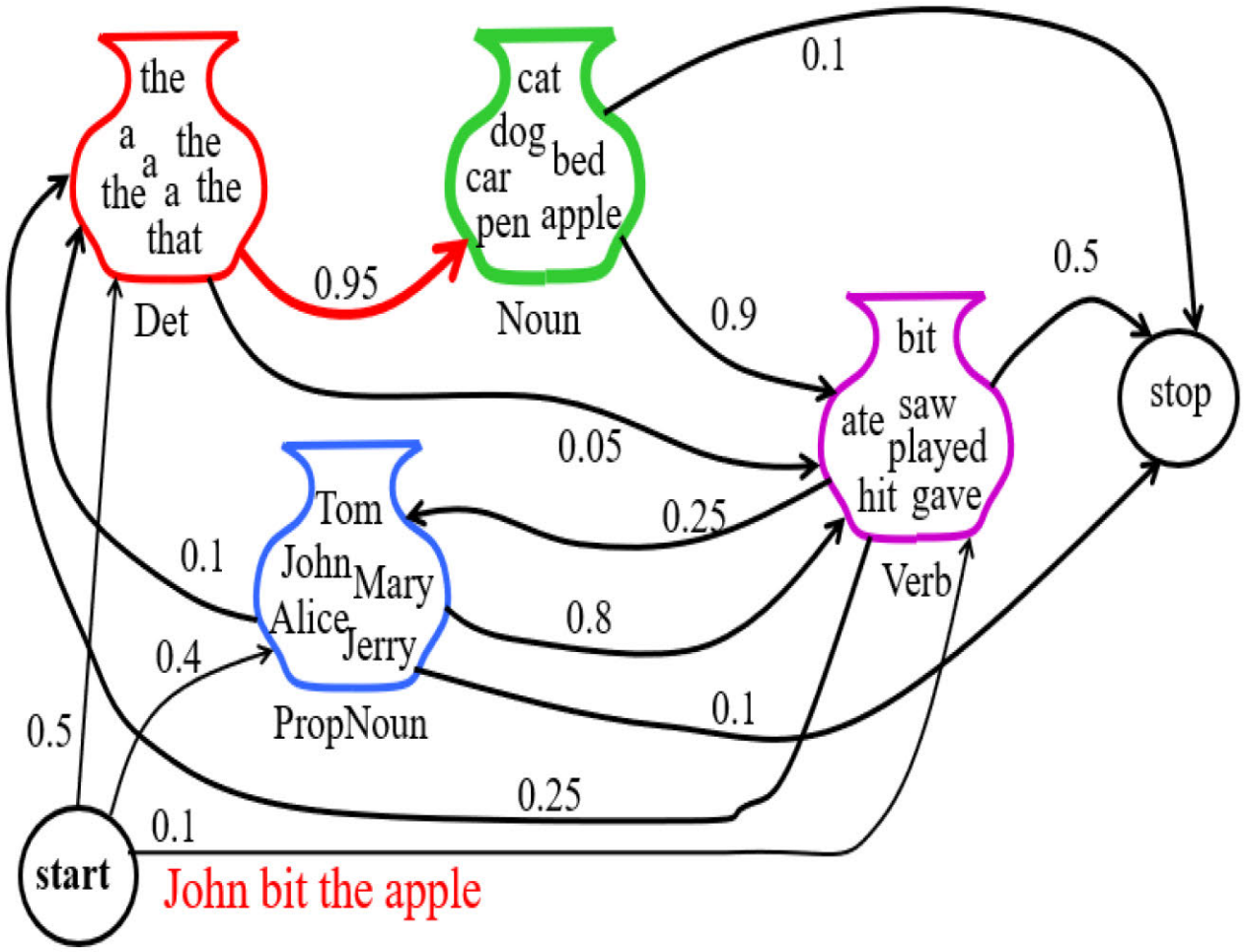
Sample HMM Generation



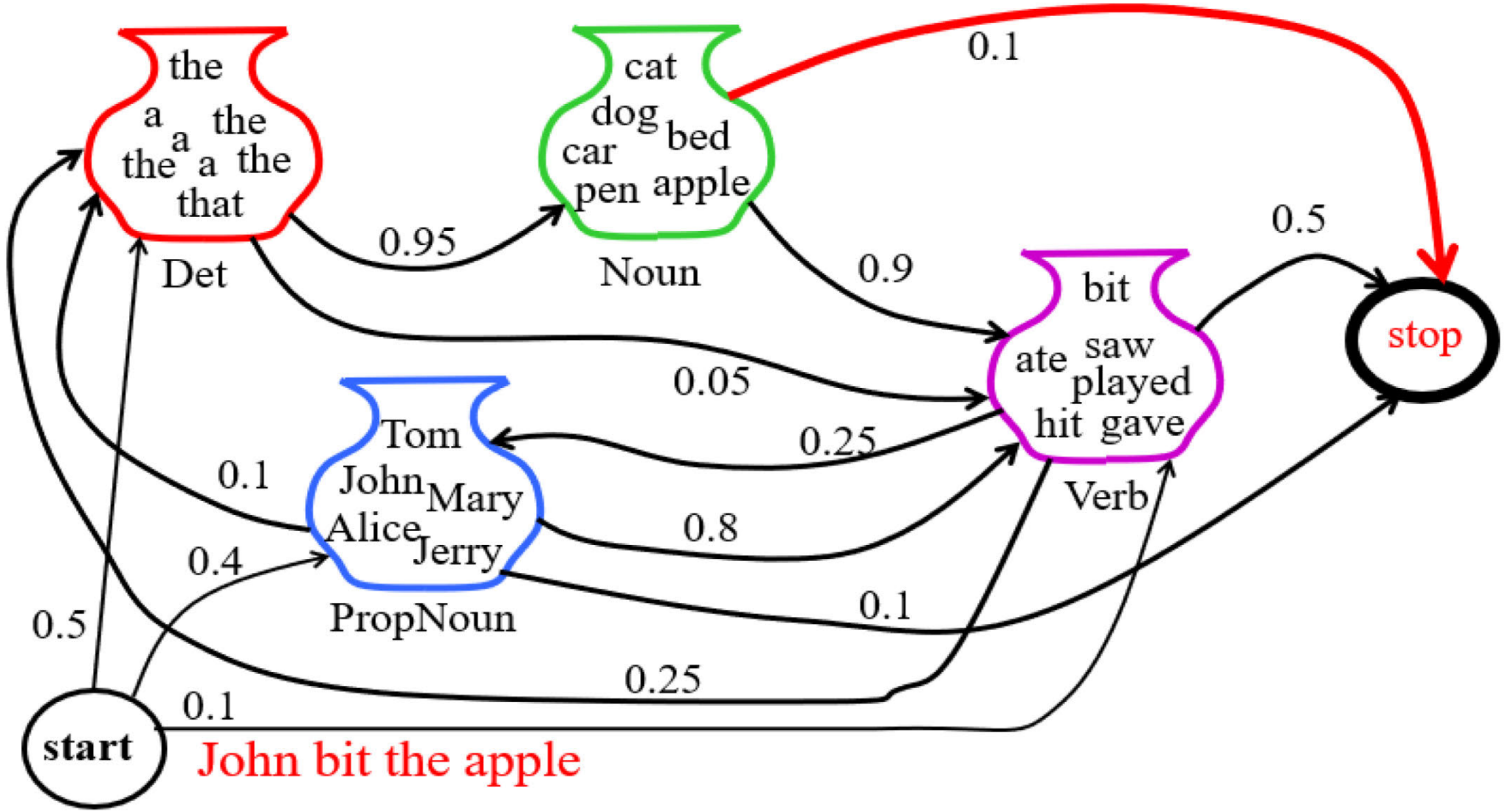


Sample HMM Generation





Sample HMM Generation



Hidden Markov Model (HMM) POS Tagging

- We have a sentence "I love Artificial Intelligence" and we need to assign POS tags to each word.
- To calculate the probabilities associated with the tags,
  - First find how likely it is for a pronoun to be followed by a verb, then an adjective, and finally a noun. These probabilities are typically called ***transitions probabilities.***
  - Secondly, we need to know how likely that the word ''I" would be a pronoun, the word "love" would be a verb, the word "Artificial" would be an adjective, and the word "Intelligence" would be a noun. These probabilities are ***called emission probabilities.***
- The ***transition probability*** is the probability that connects the change from one state to the next in the system.
- The ***emission probability*** is the probability that quantifies the possibility of making a particular observation given a defined state.

Hidden Markov Model (HMM) POS Tagging

An HMM consists of two components, the A and the B probabilities. The A matrix contains the tag transition probabilities $P(t_i \vee t_{i-1})$ and B the emission probabilities $P(w_i \vee t_i)$ where w denotes the word and t denotes the tag. The transition probability, given a tag, how often is this tag is followed by the second tag in the corpus is calculated as (3):

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \tag{3}$$

The emission probability, given a tag, how likely it will be associated with a word is given by (4):

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)} \tag{4}$$

Hidden Markov Model (HMM) POS Tagging

Figure 2 shows an example of the HMM model in POS tagging. For a given sequence of three words, "word1", "word2", and "word3", the HMM model tries to decode their correct POS tag from "N", "M", and "V". The A transition probabilities of a state to move from one state to another and B emission probabilities that how likely a word is either N, M, or V in the given example.

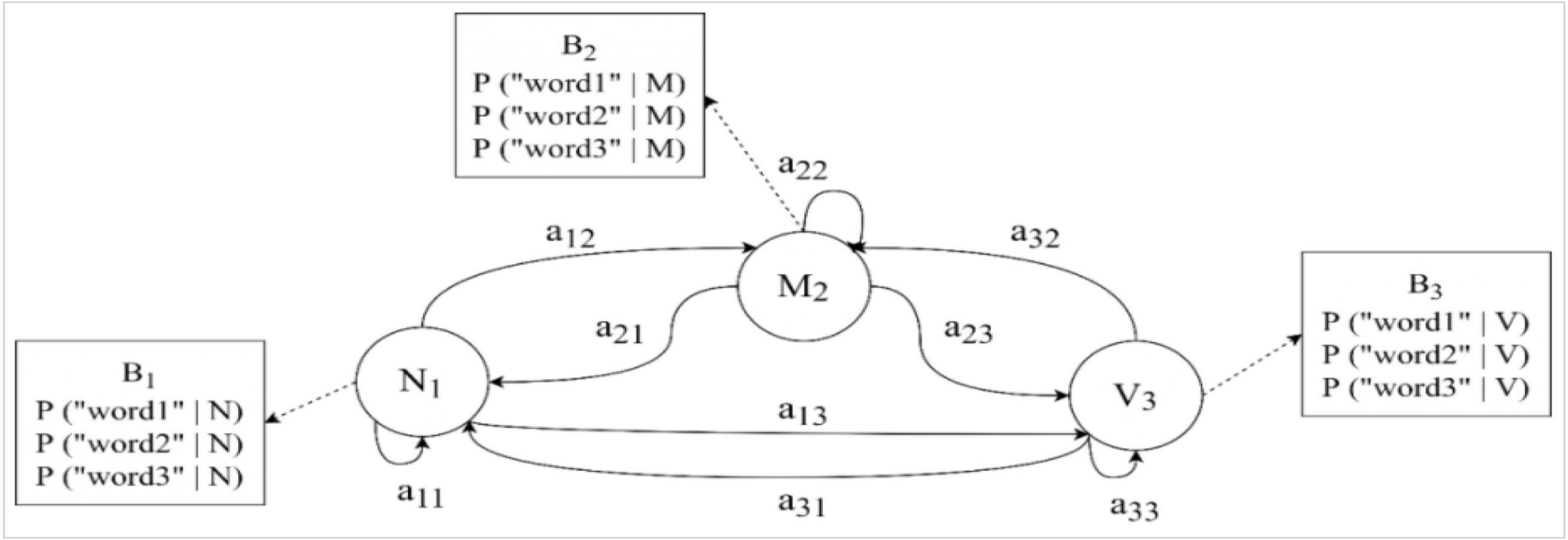


Figure 2. A Hidden Markov Model with A transition and B emission probabilities.

HMM Tagger

The process of determining hidden states to their corresponding sequence is known as decoding. More formally, given A, B probability matrices and a sequence of observations $O = o_1 \dots o_2, \dots o_T$, the goal of an HMM tagger is to find a sequence of states $Q = q_1 \dots q_2, \dots q_T$. For POS tagging the task is to find a tag sequence t_1^n that maximizes the probability of a sequence of observations of n words w_1^n (5).

$$t_1^n = \max_{t_1^n} P(t_1^n | w_1^n) \approx \max_{t_1^n} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1}) \tag{5}$$

The Viterbi Algorithm

The decoding algorithm for the HMM model is the Viterbi Algorithm. The algorithm works as setting up a probability matrix with all observations o_t in a single column and one row for each state q_i. A cell in the matrix $v_t(j)$ represents the probability of being in state j after first t observations and passing through the highest probability sequence given A and B probability matrices. Each cell value is computed by the following equation (6):

$$v_t(j) = \max_{q_1 \cdots q_{t-1}} P(q_1 \ldots q_{t-1}, o_1, o_2 \ldots o_t, q_t = j | (A, B)) \quad (6)$$

Thank You
???

- ***References:***

1) Foundations of Statistical Natural Language Processing - Christopher D. Manning.
2) Text Data Management and Analysis - A Practical Introduction to Information Retrieval and Text Mining
3) Natural Language Processing using Python- Apress (2019)
4) Applied Text Analysis with Python O'Reilly Media (2018)
5) Natural Language Processing Python and NLTK-Packt Publishing (2016)
6) Building Machine Learning Systems with Python-Packt Publishing (2015)