Word classes and part of speech tagging

Reading: Ch. 5, Jurafsky & Martin (Ch. 10 3rd. Edition)

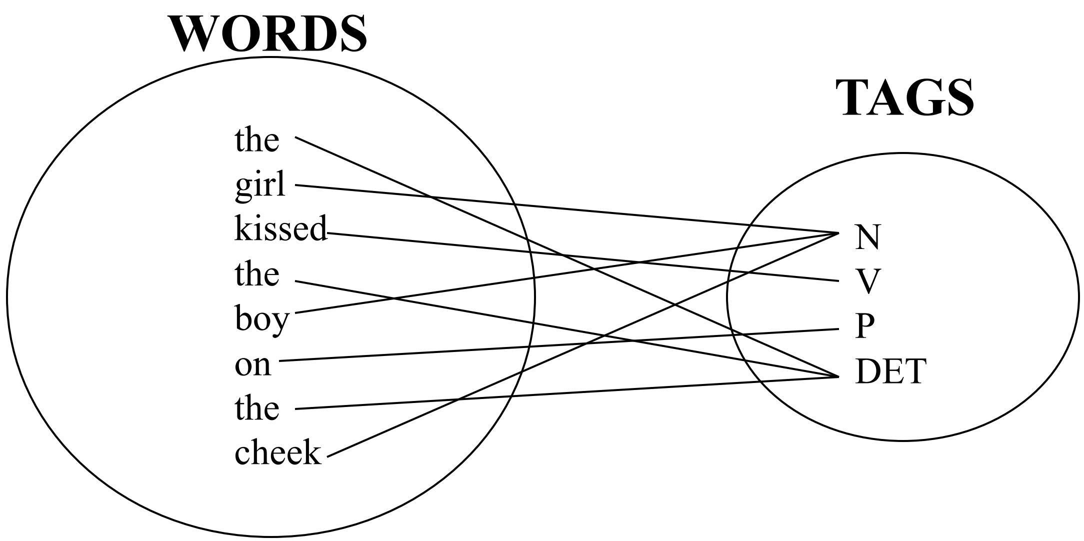
PI Disclosure: This set includes adapted material from Rada Mihalcea, Raymond Mooney and Dan Jurafsky

Outline

- Why part of speech tagging?
- Tag sets and problem definition
- Automatic approaches:
  - HMMs
  - MEMMs

Definition

"The process of assigning a part-of-speech or other lexical class marker to each word in a corpus" (Jurafsky and Martin)



An Example

WORD	LEMMA	TAG
the	the	+DET
girl	girl	+NOUN
kissed	kiss	+VPAST
the	the	+DET
boy	boy	+NOUN
on	on	+PREP
the	the	+DET
cheek	cheek	+NOUN

From: http://www.xrce.xerox.com/competencies/content-analysis/fsnlp/tagger.en.html

Why is POS Tagging Useful?

- First step of a vast number of practical tasks
- Speech synthesis
  - INsult		inSULT
  - OBject		obJECT
  - OVERflow		overFLOW
  - DIScount		disCOUNT
  - CONtent		conTENT
- Parsing
  - Need to know if a word is an N or V before you can parse
- Information extraction
  - Finding names, relations, etc.
- Machine Translation

Outline

- Why part of speech tagging?
- **Tag sets and problem definition**
- Automatic approaches:
  - HMMs
  - MEMMs

POS Tagging: Choosing a Tagset

- Could pick very coarse tagsets
  - N, V, Adj, Adv.
- More commonly used set is finer grained, the "Penn TreeBank tagset", 45 tags
  - PRP$, WRB, WP$, VBG
- Even more fine-grained tagsets exist

English POS Tagsets

- Original Brown corpus used a large set of 87 POS tags.
- Most common in NLP today is the Penn Treebank set of 45 tags.
- The C5 tagset used for the British National Corpus (BNC) has 61 tags.
- Universal POS tagset includes ~12 tags.

Penn TreeBank POS Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

POS Tagging: The Problem

- Words often have more than one word class:
  - *This* is a nice day = PRP
  - *This* day is nice = DT
  - You can go *this* far = RB
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB

POS Tagging: Definition

- The process of assigning a part-of-speech or other lexical class marker to each word in a corpus
- Often applied to punctuation markers as well
- Input: a string of words and a specified tagset
- Output: a single best tag for each word

How Hard is POS Tagging? Measuring Ambiguity

		WSJ		Brown	
Types:					
Unambiguous	(1 tag)	44,432	(**86%**)	45,799	(**85%**)
Ambiguous	(2+ tags)	7,025	(**14%**)	8,050	(**15%**)
Tokens:					
Unambiguous	(1 tag)	577,421	(**45%**)	384,349	(**33%**)
Ambiguous	(2+ tags)	711,780	(**55%**)	786,646	(**67%**)

Consistent Human Annotation is Crucial

- Disambiguating POS tags requires deep knowledge of syntax
- Development of clear heuristics in the form of tagging manuals
  - She told off/RP her friends
  - She told her friends off/RP
  - She stepped off/IN the train
  - *She stepped the train off/IN
- See Manning (2011) for a more in depth discussion

Part-of-Speech Tagging

- Manual Tagging
- Automatic Tagging
  - Two flavors:
    - Rule-based taggers (EngCC ENGTWOL)
    - Stochastic taggers (HMM, Max Ent, CRF-based, Tree-based)
  - Hodge-podge:
    - Transformation–based tagger (Brill, 1995)

Manual Tagging

- Agree on a Tagset after much discussion.
- Chose a corpus, annotate it manually by two or more people.
- Check on inter-annotator agreement.
- Fix any problems with the Tagset (if still possible).

Outline

- Why part of speech tagging?
- Tag sets and problem definition
- **Automatic approaches:**
  - HMMs
  - MEMMs

Classification Learning

- Typical machine learning addresses the problem of classifying a feature-vector description into a fixed number of classes.
- There are many standard learning methods for this task:
  - Decision Trees and Rule Learning
  - Naïve Bayes and Bayesian Networks
  - Logistic Regression / Maximum Entropy (MaxEnt)
  - Perceptron and Neural Networks
  - Support Vector Machines (SVMs)
  - Nearest-Neighbor / Instance-Based

Beyond Classification Learning

- Standard classification problem assumes individual cases are disconnected and independent.
- Many NLP problems do not satisfy this assumption (involve many connected decisions, ambiguity and dependence)

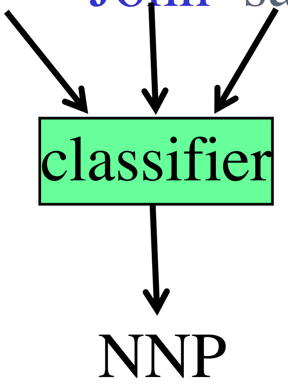
Sequence Labeling Problem

- Many NLP problems can viewed as sequence labeling.
- Each token in a sequence is assigned a label.
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors (not i.i.d).

Sequence Labeling as Classification

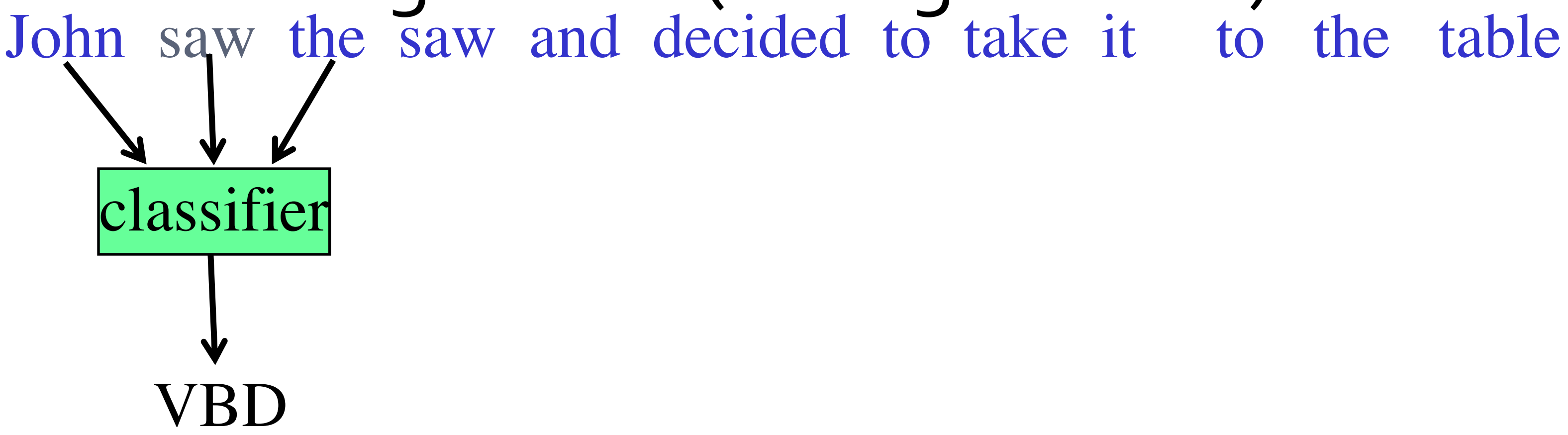
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.



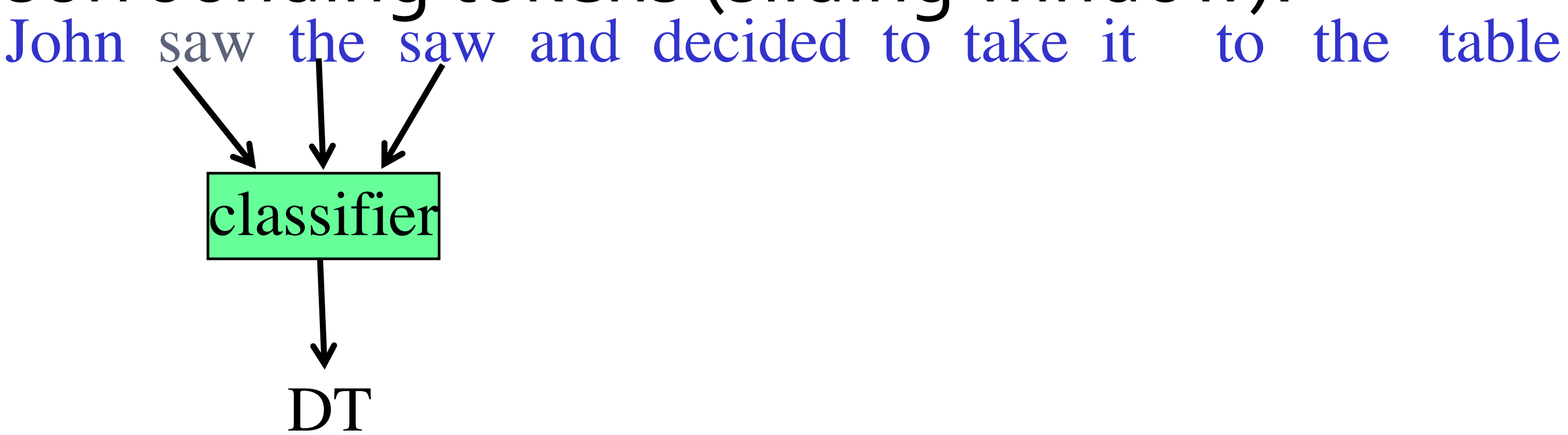
Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Sequence Labeling as Classification

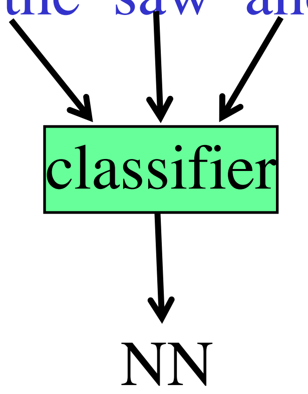
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

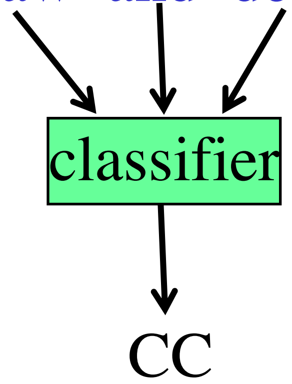
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

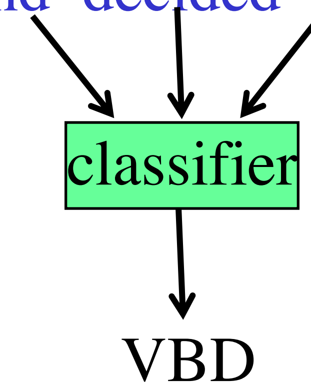
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

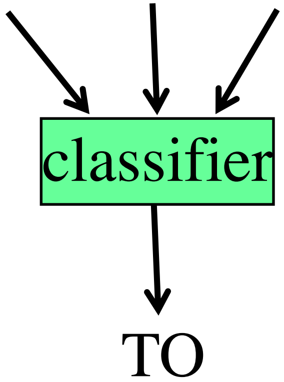
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

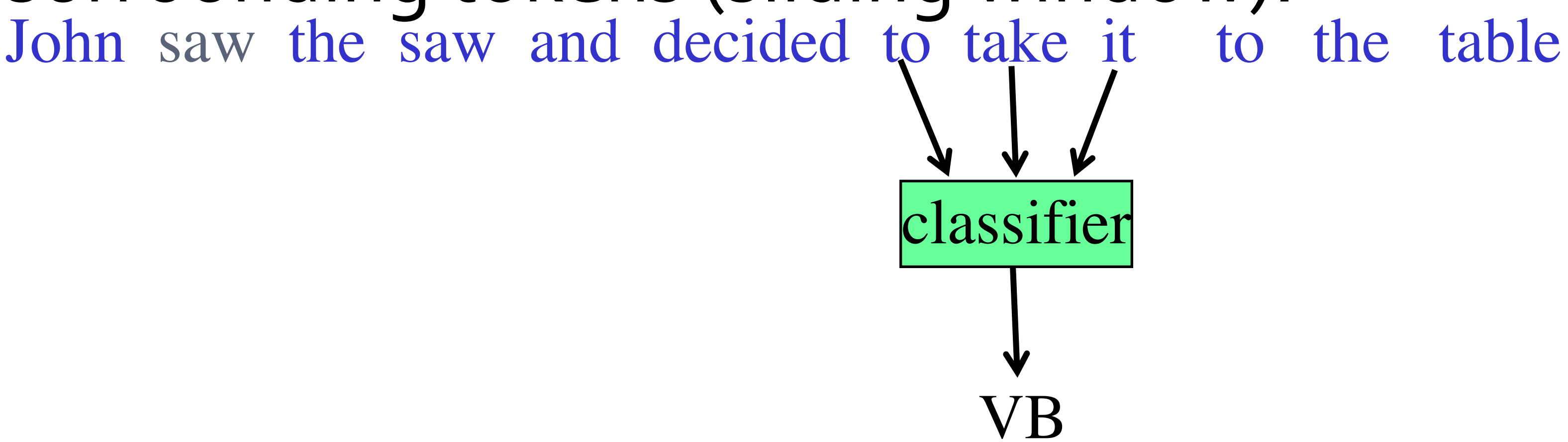
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

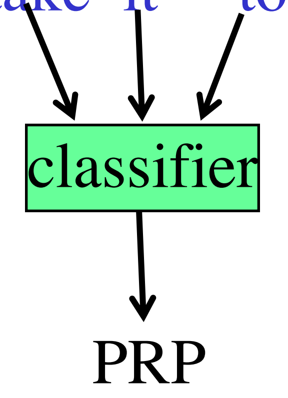
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

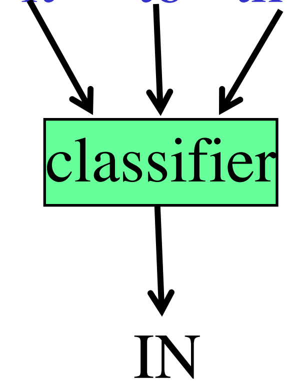
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

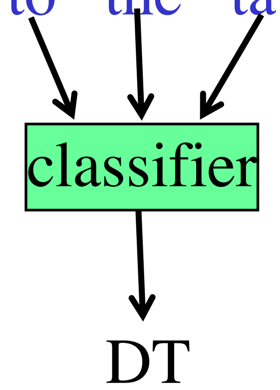
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

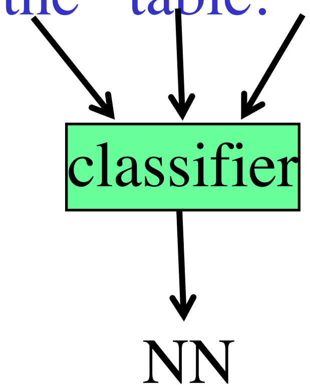
John saw the saw and decided to take it to the table.



Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.



Problems with Sequence Labeling as Classification

- Not easy to integrate information from category of tokens on both sides.
- Difficult to propagate uncertainty between decisions and "collectively" determine the most likely joint assignment of categories to all of the tokens in a sequence.

Probabilistic Sequence Models

- Probabilistic sequence models allow interdependent classifications and collectively determine the most likely global assignment.
- Two standard models
  - Hidden Markov Model (HMM)
  - Conditional Random Field (CRF)

POS Tagging as Sequence Classification

- We are given a sentence (an "observation" or "sequence of observations")
  - *Secretariat is expected to race tomorrow*
- What is the best sequence of tags that corresponds to this sequence of observations?
- Probabilistic view:
  - Consider all possible sequences of tags
  - Out of this universe of sequences, choose the tag sequence which is most probable given the observation sequence of n words $w_1 \ldots w_n$.

Getting to HMMs

- We want, out of all sequences of n tags $t_1 \ldots t_n$ the single tag sequence such that $P(t_1 \ldots t_n | w_1 \ldots w_n)$ is highest.

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

- Hat ^ means "our estimate of the best one"
- Argmax_x f(x) means "the x such that f(x) is maximized"

Likelihood and Prior

HMM Taggers choose tag sequence that maximizes this formula:

– P(word|tag) × P(tag|previous n tags)

$$\max_{t_1^n} \overbrace{P(w_1^n|t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

$$P(w_1^n|t_1^n) \approx \prod_{i=1} P(w_i|t_i)$$

$$\prod^{n} P(t_i|t_{i-1})$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Two Kinds of Probabilities

- Tag transition probabilities $p(t_i|t_{i-1})$
  - Determiners likely to precede adjs and nouns
    - That/DT flight/NN
    - The/DT yellow/JJ hat/NN
    - So we expect P(NN|DT) and P(JJ|DT) to be high
    - But P(DT|JJ) to be:
  - Compute P(NN|DT) by counting in a labeled corpus:

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

$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

Two Kinds of Probabilities

- Word likelihood probabilities $p(w_i|t_i)$
  - VBZ (3sg Pres verb) likely to be "is"
  - Compute P(is|VBZ) by counting in a labeled corpus:

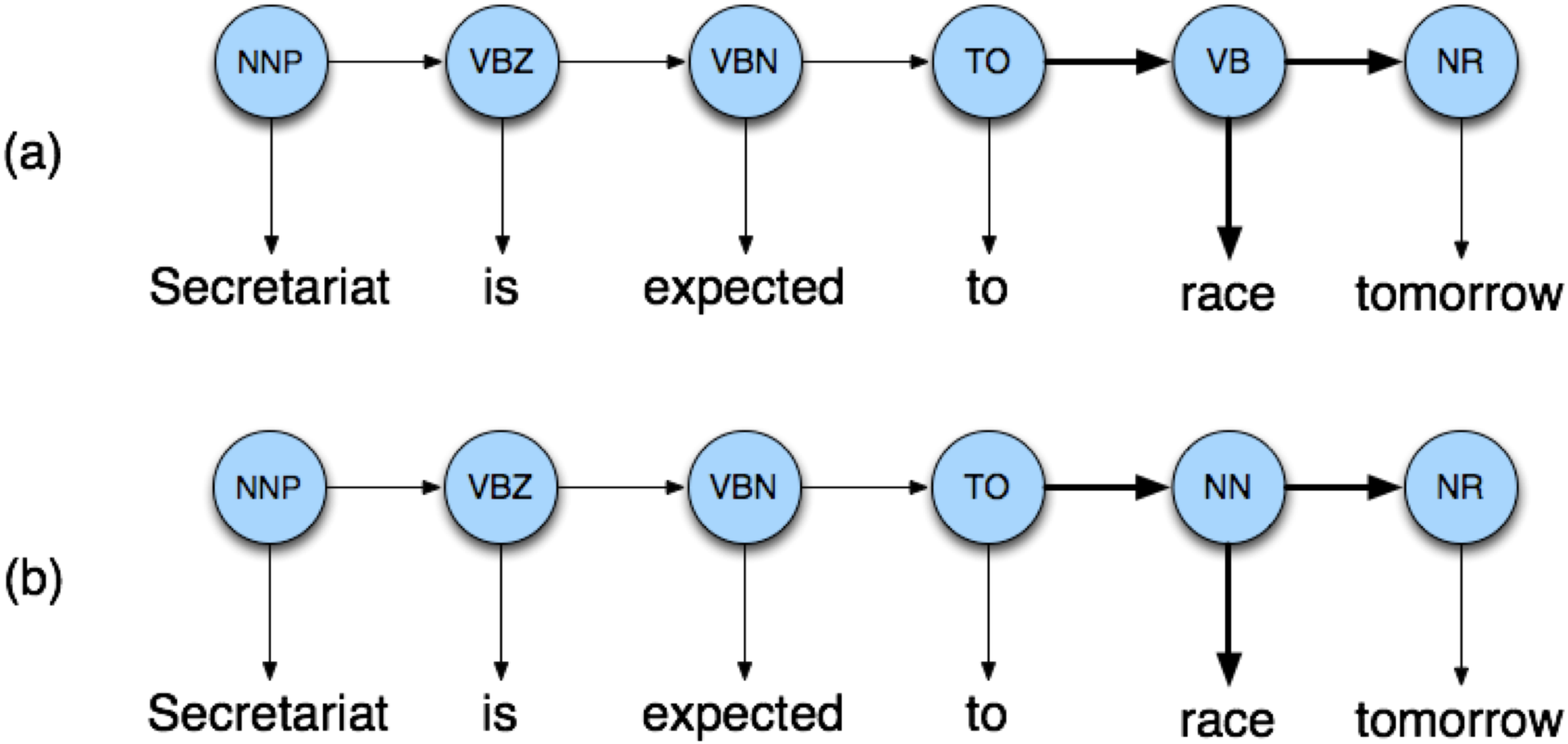
$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

Example: The Verb "race"

- Secretariat/NNP is/VBZ expected/VBN to/TO **race**/VB tomorrow/NR
- People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT **race**/NN for/IN outer/JJ space/NN
- How do we pick the right tag?

Disambiguating "race"



Example

- $P(NN|TO) = .00047$
- $P(VB|TO) = .83$
- $P(race|NN) = .00057$
- $P(race|VB) = .00012$
- $P(NR|VB) = .0027$
- $P(NR|NN) = .0012$
- $P(VB|TO)P(NR|VB)P(race|VB) = .00000027$
- $P(NN|TO)P(NR|NN)P(race|NN)=.00000000032$

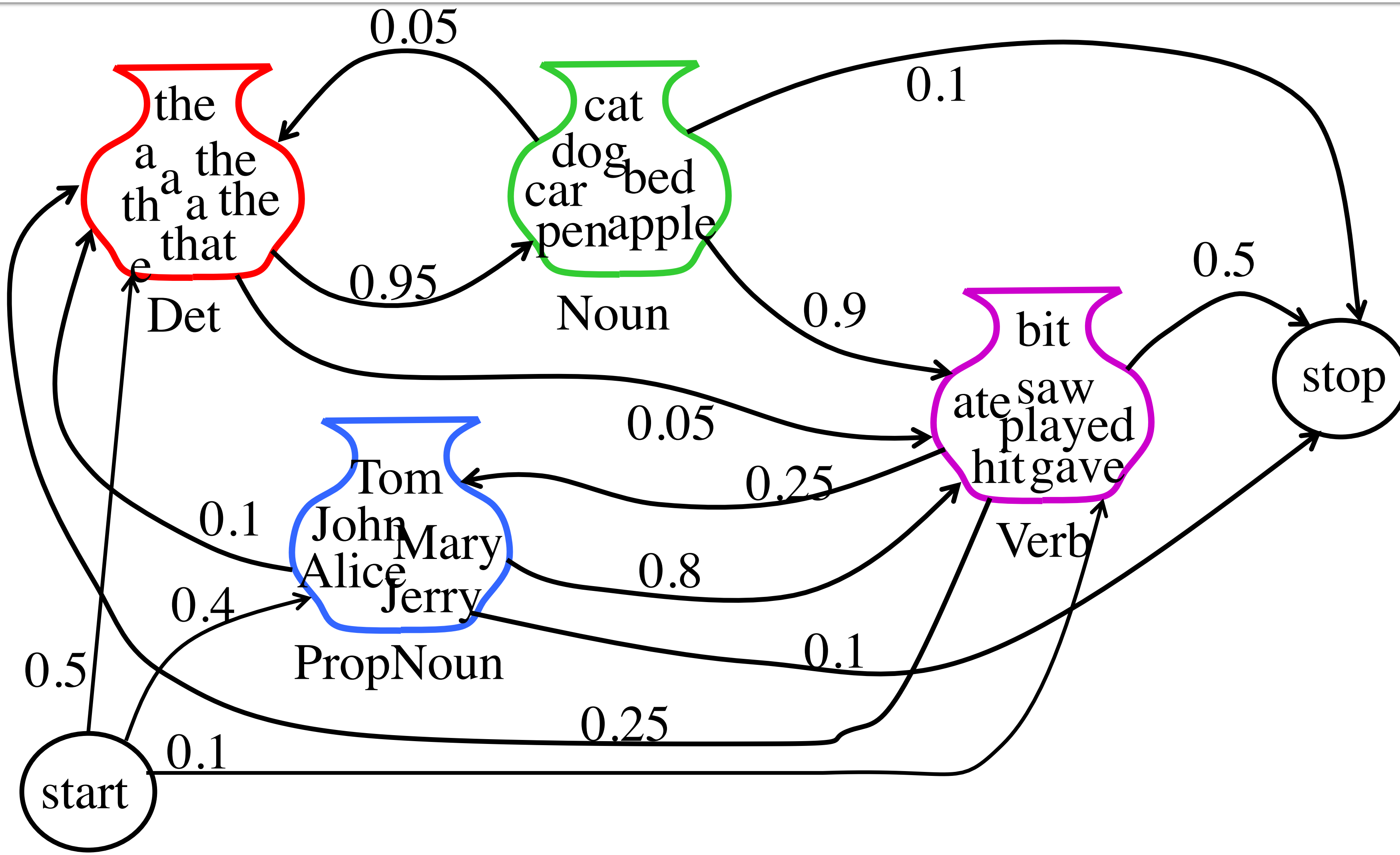
Hidden Markov Models

- What we've described with these two kinds of probabilities is a Hidden Markov Model (HMM)
- MM vs HMM
  - Both use the same algorithms for tagging they differ on the training algorithms

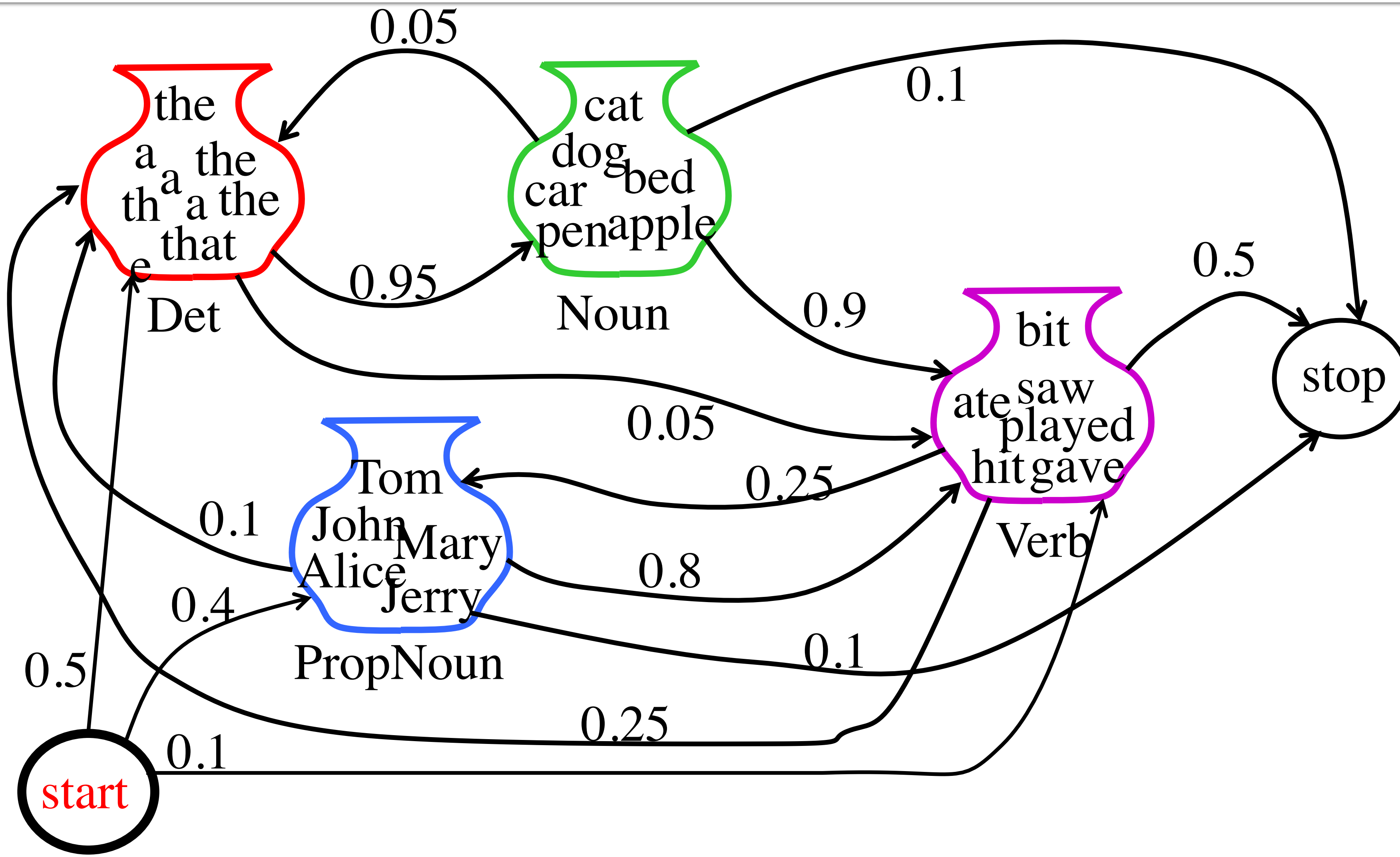
An Early Approach to Statistical POS Tagging

- PARTS tagger (Church, 1988): Stores probability of tag given word instead of word given tag.
  - P(tag|word) × P(tag|previous n tags)
- Compare to:
  - P(word|tag) × P(tag|previous n tags)
- What is the main difference between PARTS tagger (Church) and the HMM tagger?

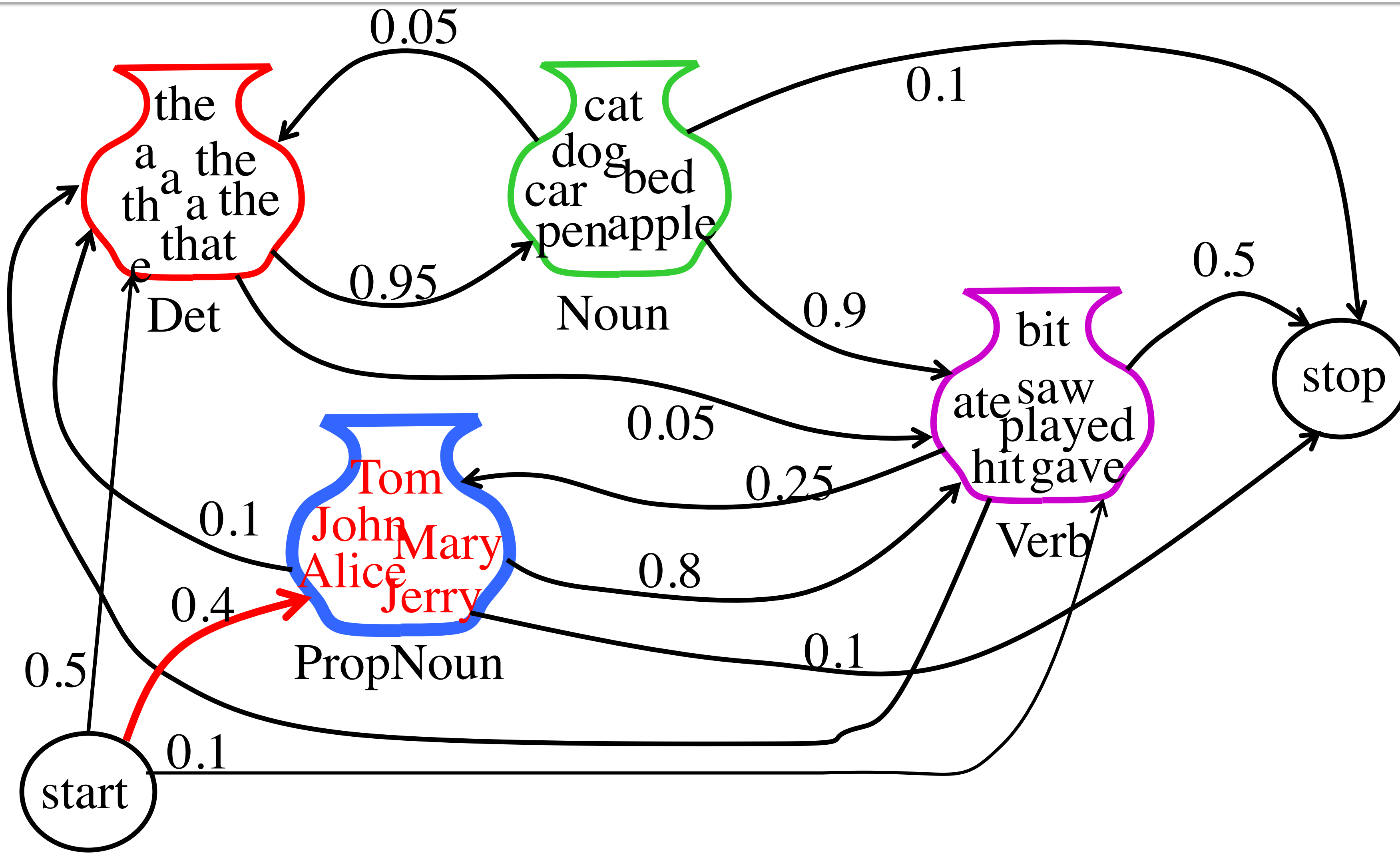
Sample HMM for POS



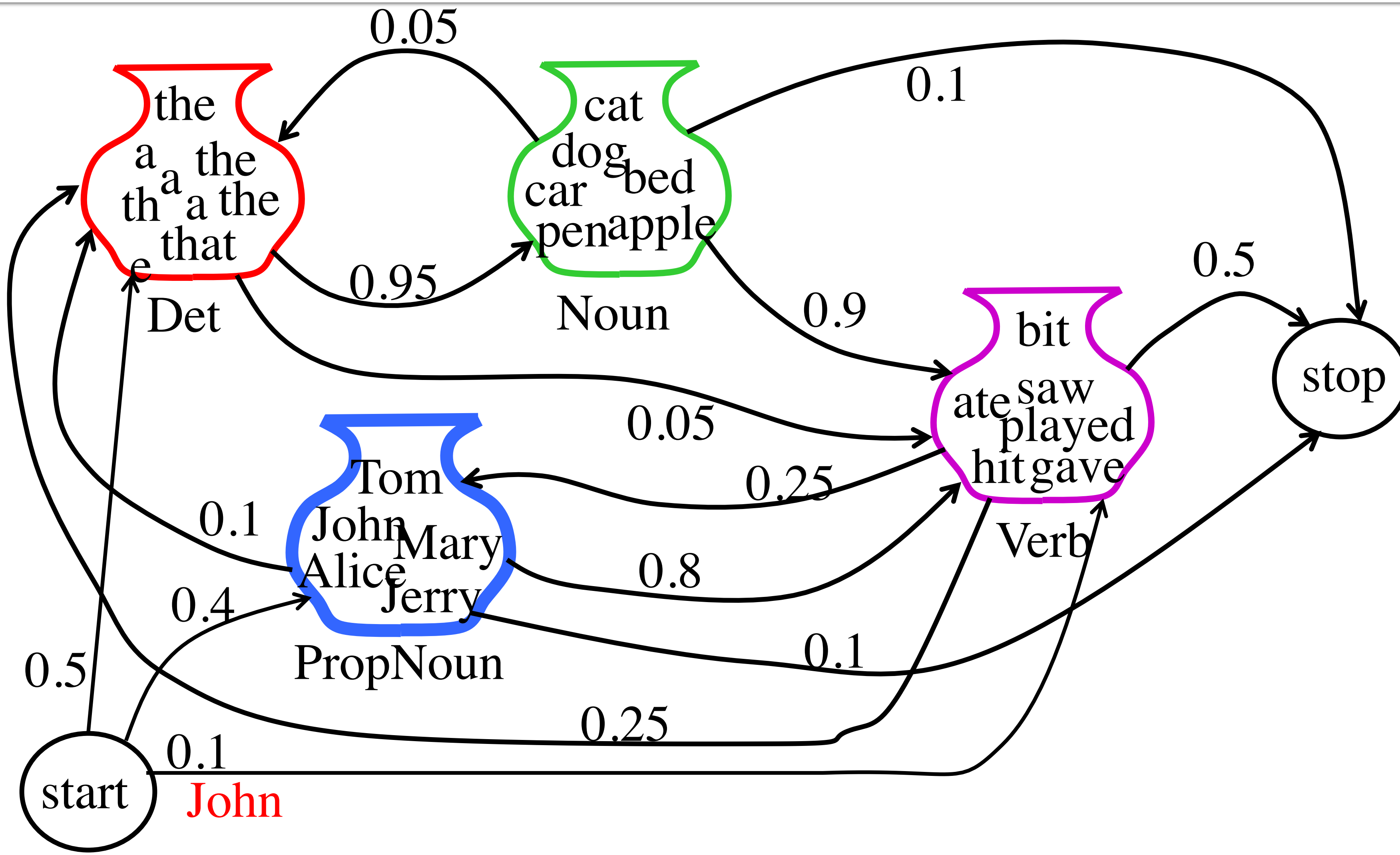
Sample HMM Generation



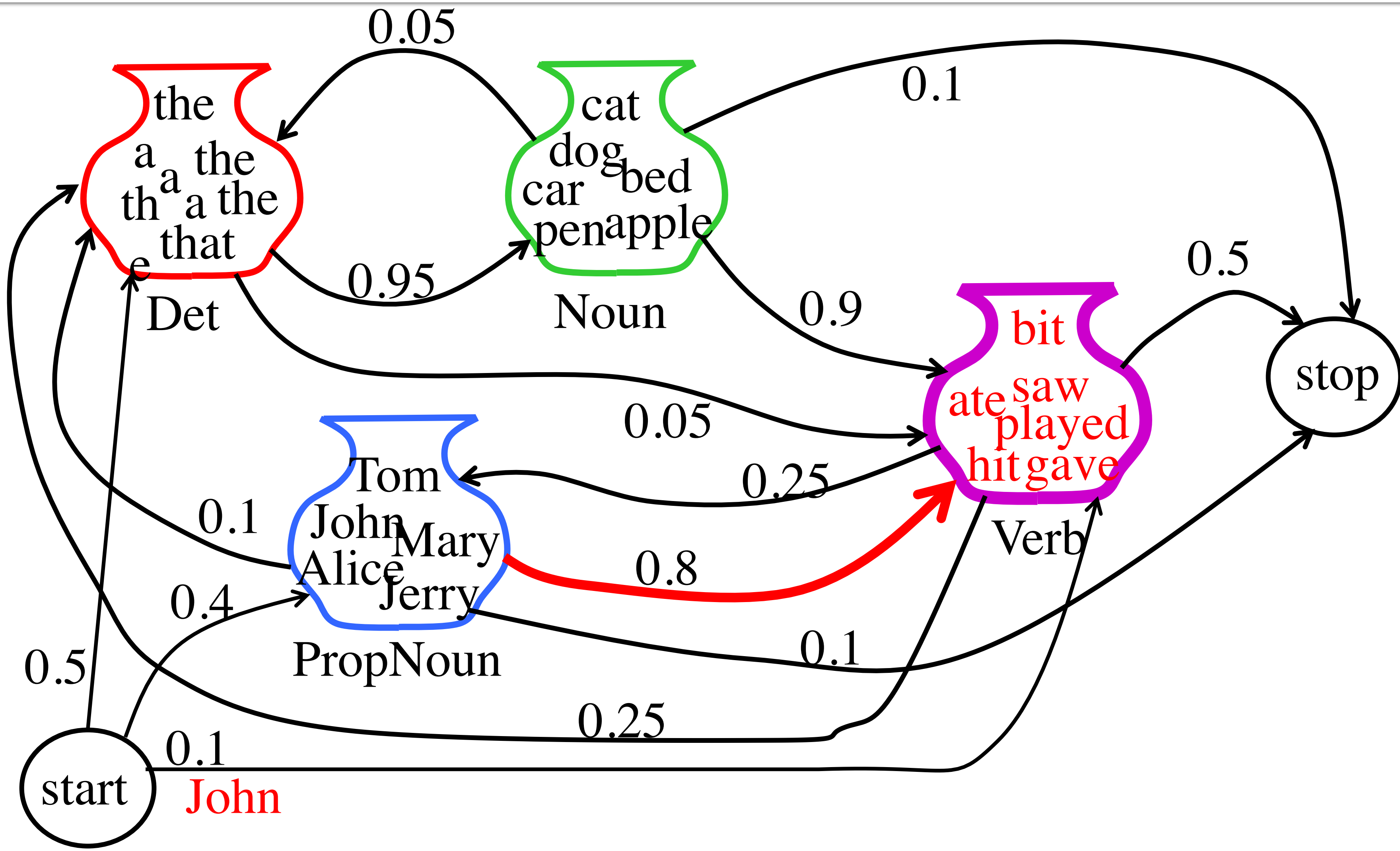
Sample HMM Generation



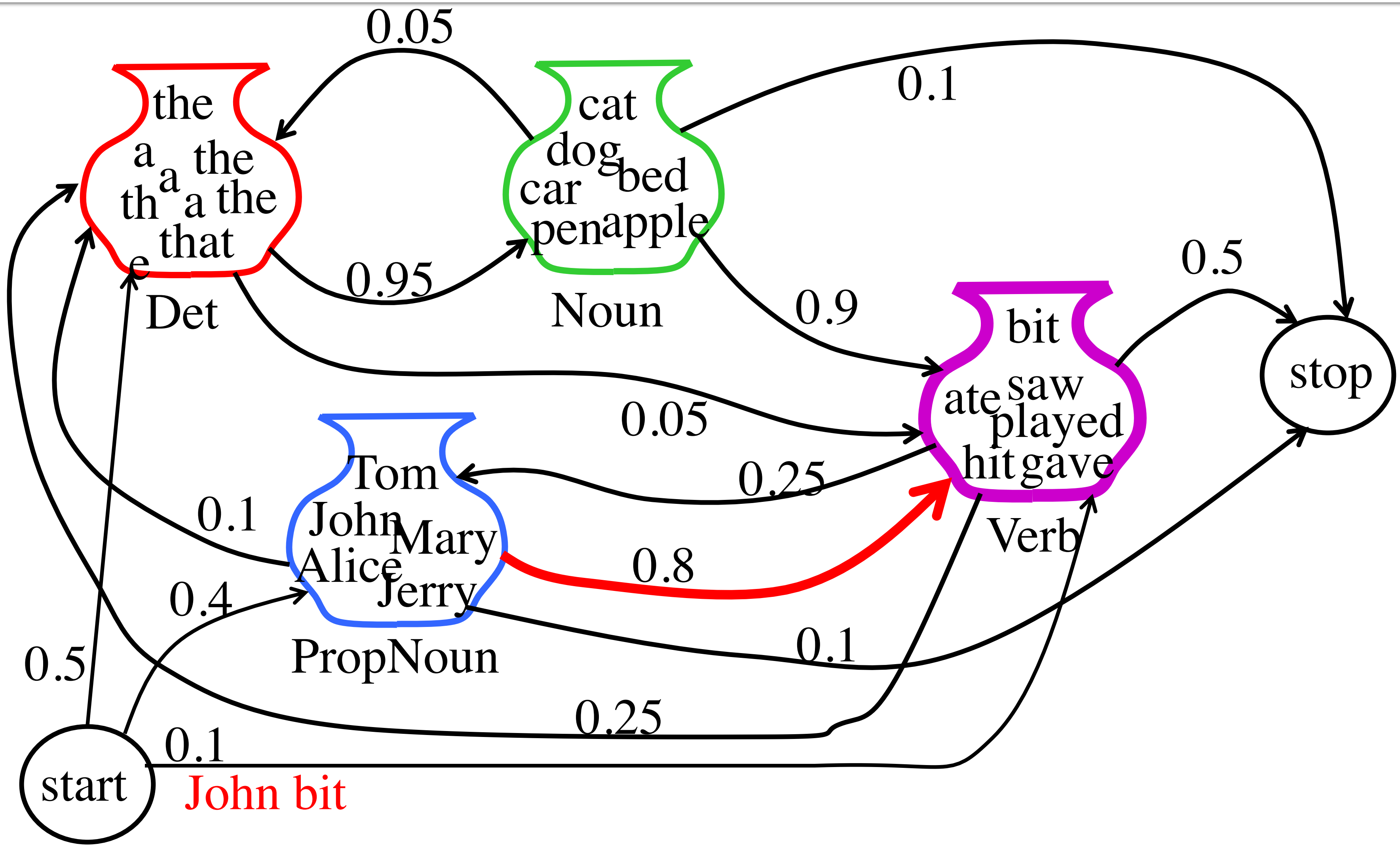
Sample HMM Generation



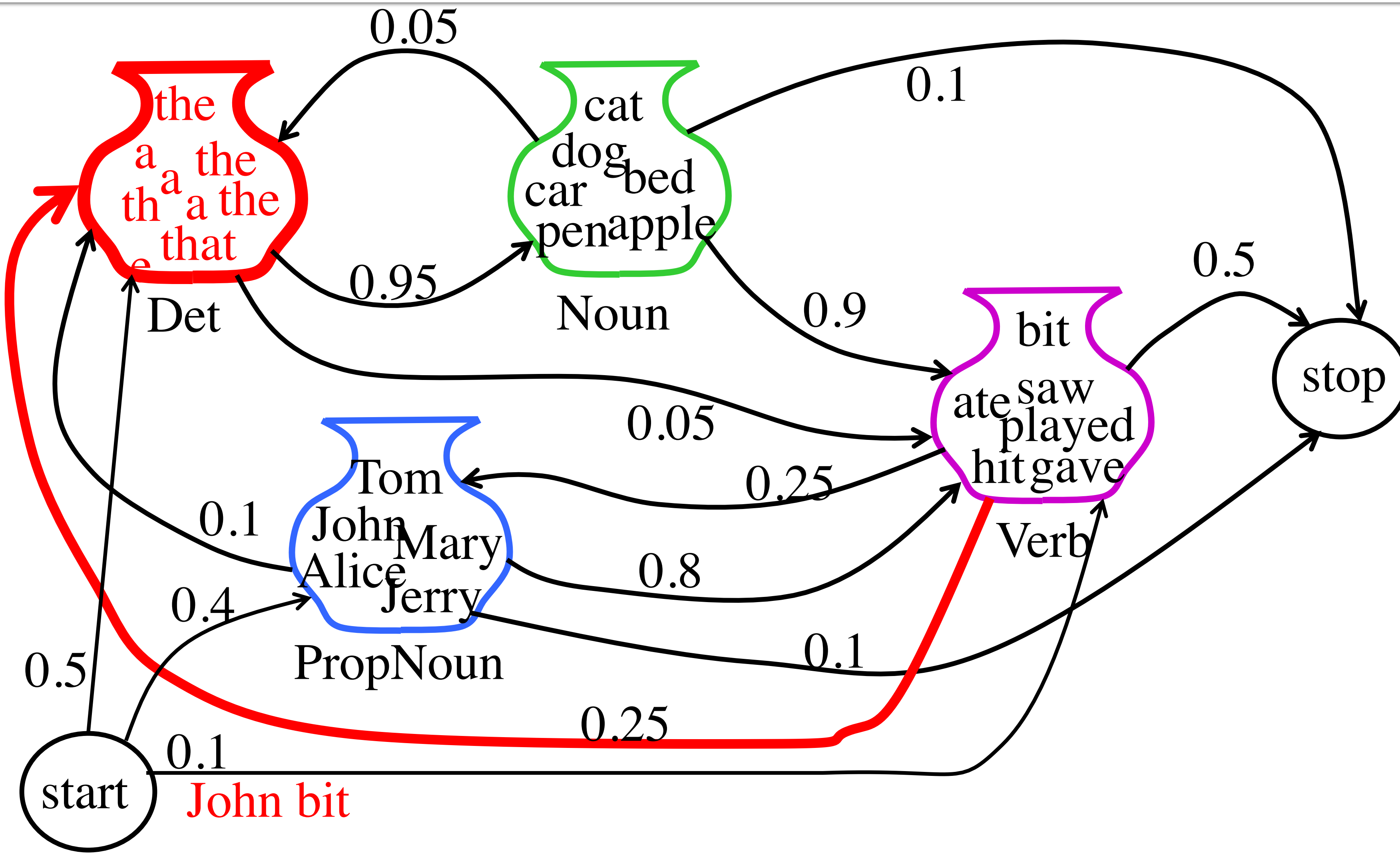
Sample HMM Generation



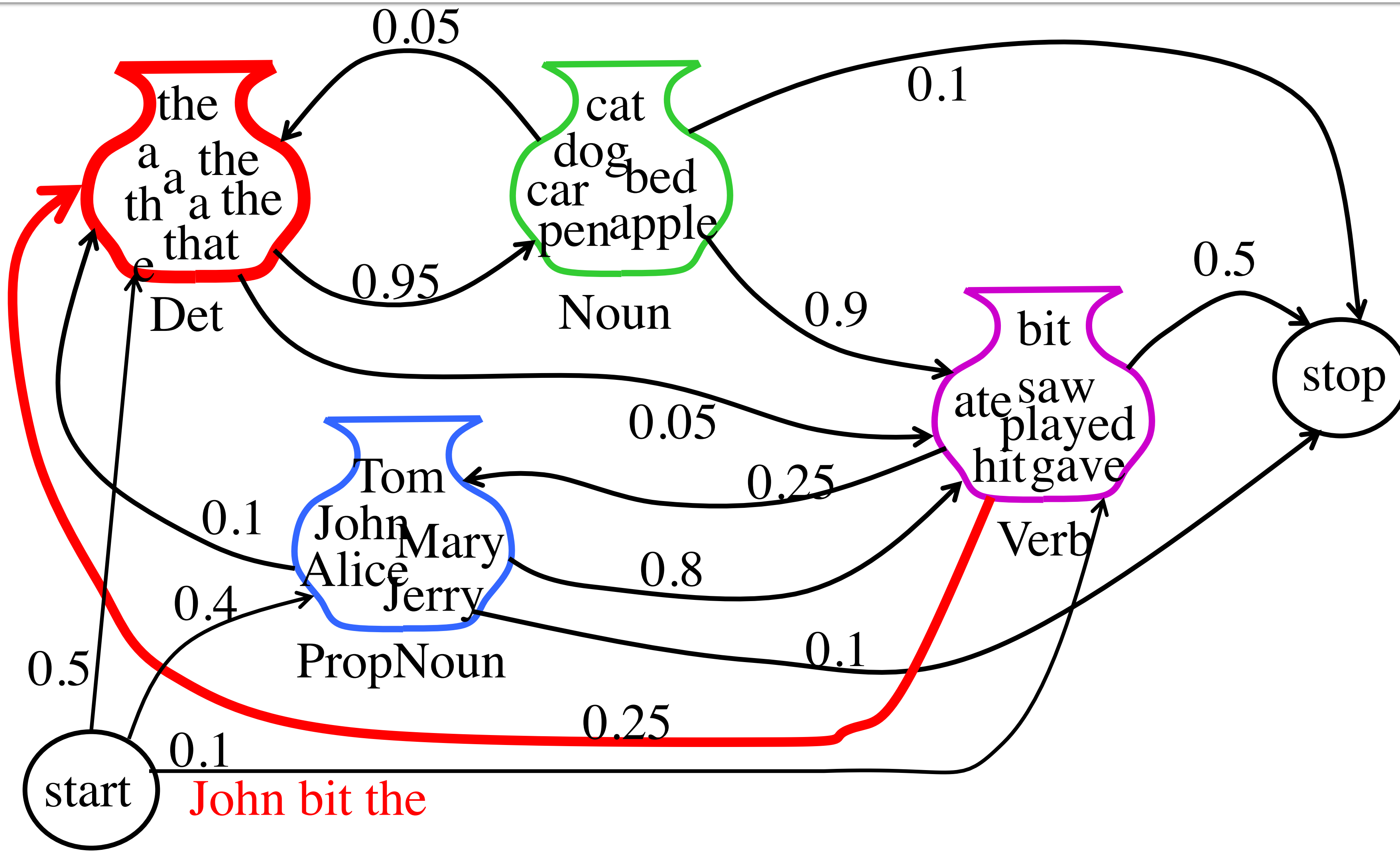
Sample HMM Generation



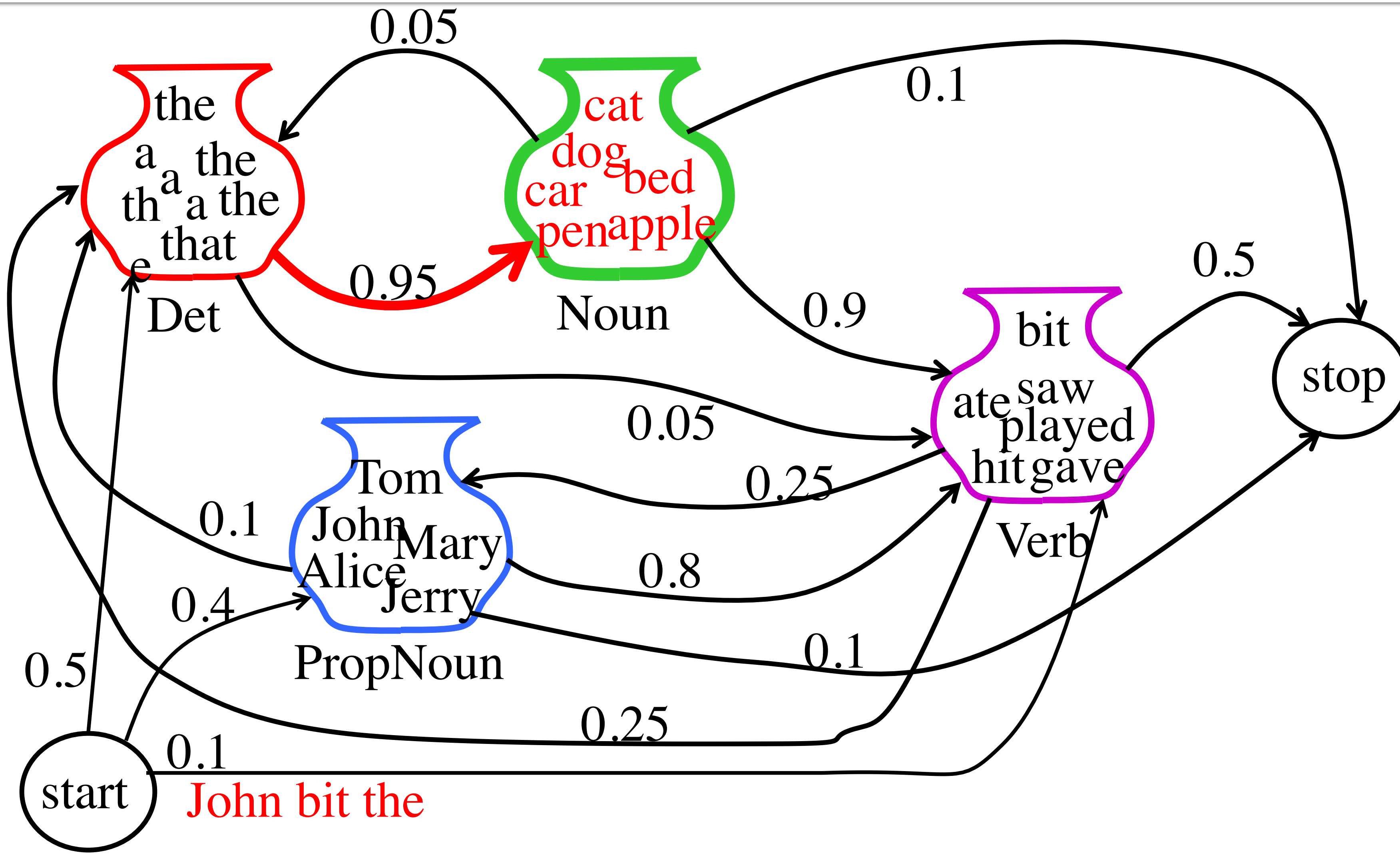
Sample HMM Generation



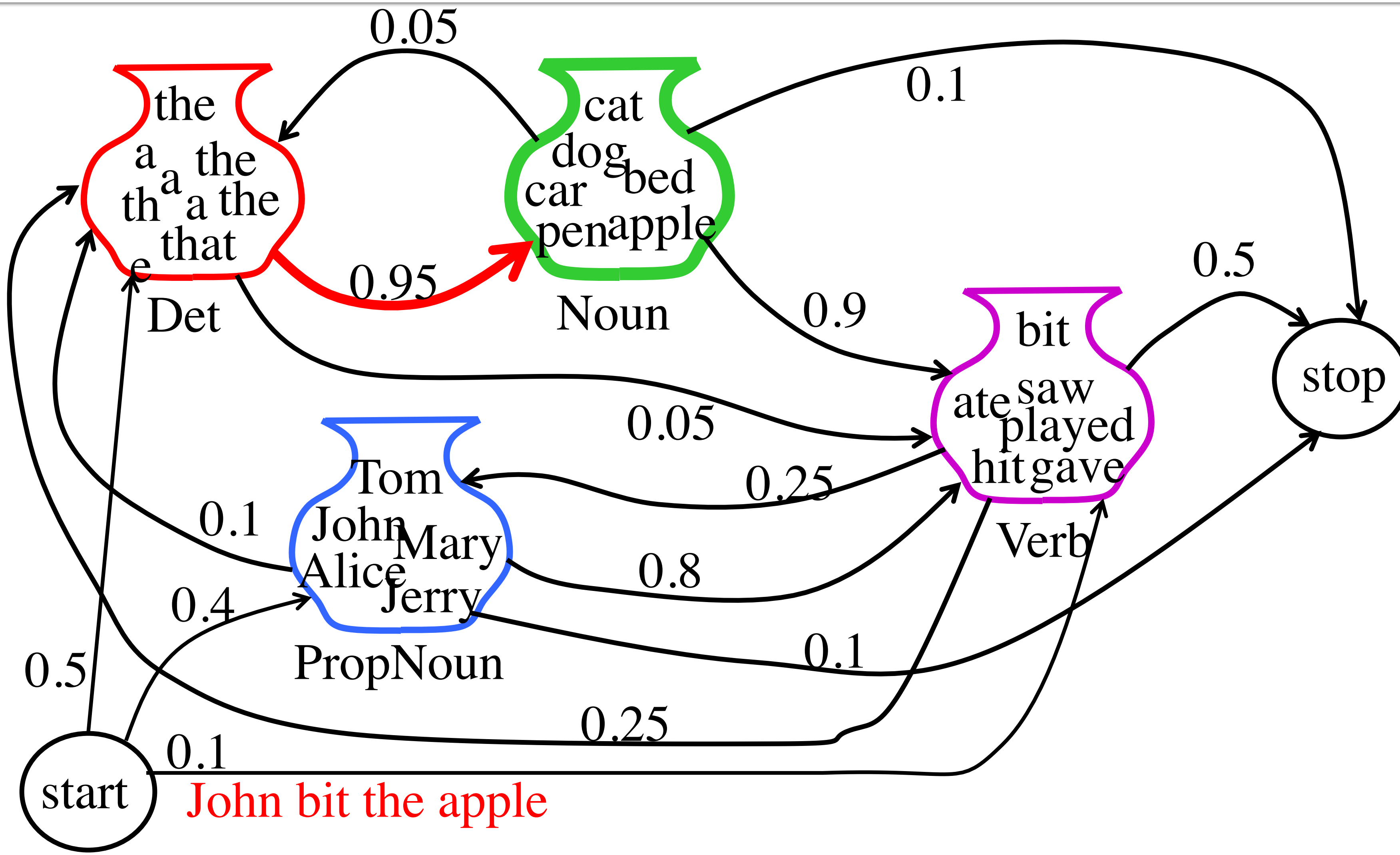
Sample HMM Generation



Sample HMM Generation

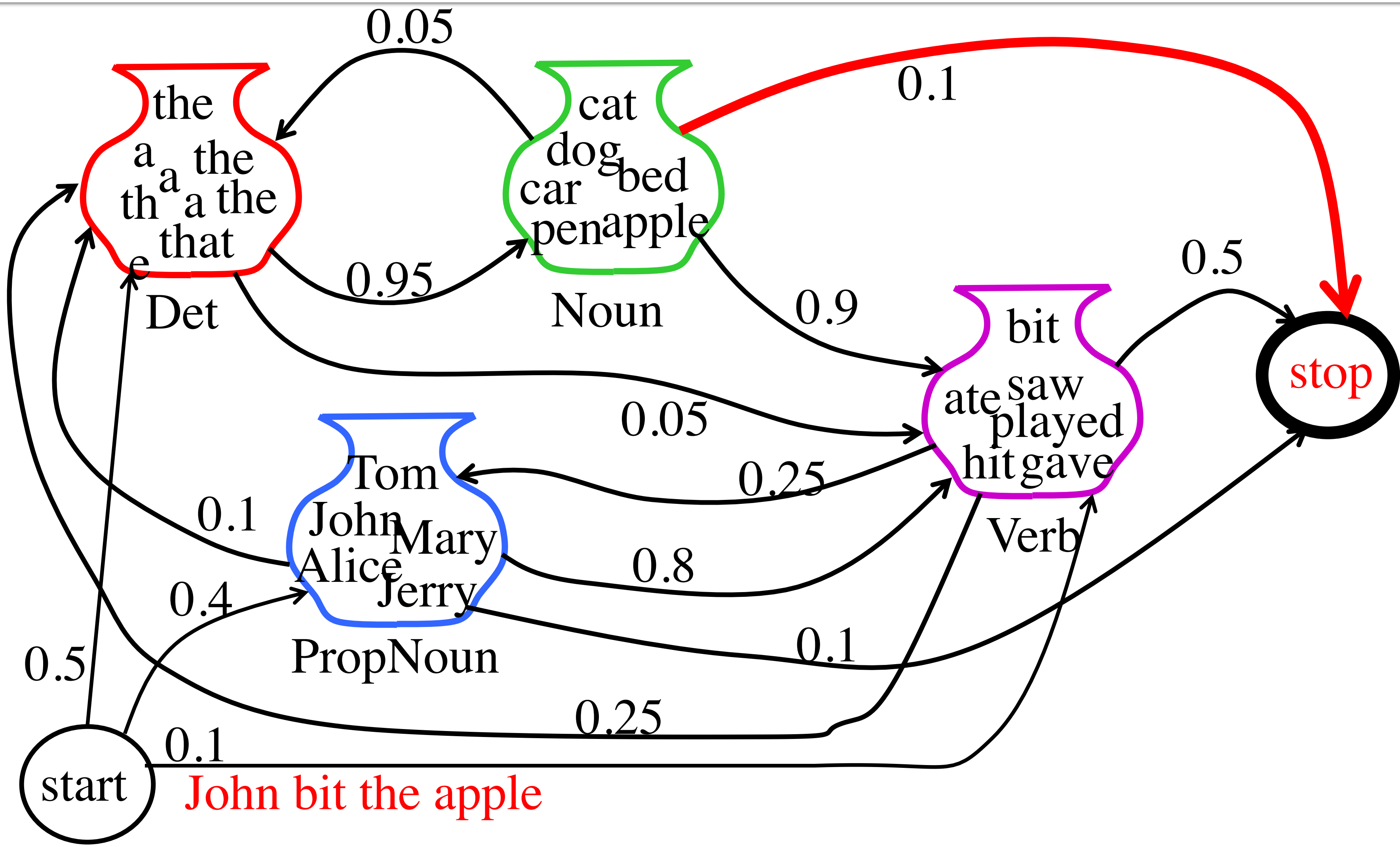


Sample HMM Generation



John bit the apple

Sample HMM Generation



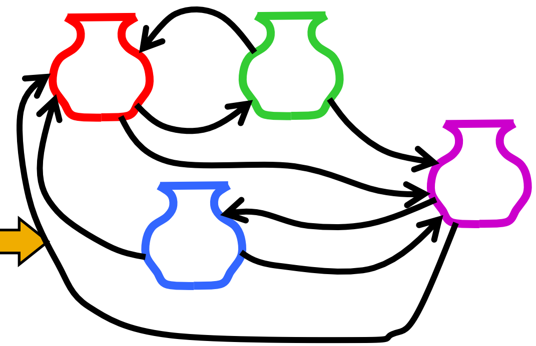
Supervised HMM Training

- If training sequences are labeled (tagged) with the underlying state sequences that generated them, then the parameters, λ={A,B} can all be estimated directly.

Training Sequences

John ate the apple
A dog bit Mary
Mary hit the dog
John gave Mary the cat.
.
.
.

Supervised HMM Training



Det Noun PropNoun Verb

Supervised Parameter Estimation

- Estimate state transition probabilities based on tag bigram and unigram statistics in the labeled data.

$$a_{ij} = \frac{C(q_t = s_i, \mathrm{q}_{\mathrm{t+1}} = s_j)}{C(q_t = s_i)}$$

- Estimate the observation probabilities based on tag/word co-occurrence statistics in the labeled data.

$$b_j(k) = \frac{C(q_i = s_j, o_i = v_k)}{C(q_i = s_j)}$$

- Use appropriate smoothing if training data is sparse.

Evaluating Taggers

- Train on ***training set*** of labeled sequences.
- Possibly tune parameters based on performance on a ***development set***.
- Measure accuracy on a disjoint ***test set***.
- Generally measure ***tagging accuracy***, i.e. the percentage of tokens tagged correctly.
- Accuracy of most modern POS taggers, including HMMs is 96–97% (for Penn tagset trained on about 800K words) . Most freq. baseline: ~92.4%
  - Generally matching human agreement level.