HMM Review (continued)

Class-Based Sequence Models

- From Brown et al. (1990):

$$p(\text{start}, w_1, w_2, \ldots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i \mid \text{cl}(w_i)) \times \eta(\text{cl}(w_i) \mid \text{cl}(w_{i-1}))$$

- "cl" is a deterministic function from words to a smaller set of classes.
  - Each word only gets one class; known in advance.
  - Discovered from data using a clustering algorithm.

start

start C53

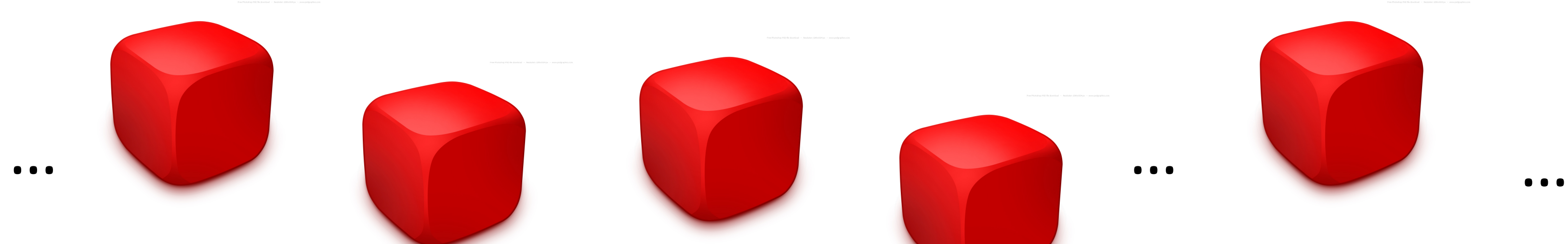
one “next class” die per class:

Each word appears on
only one of the word dice.

start C53

I

one word die per class:

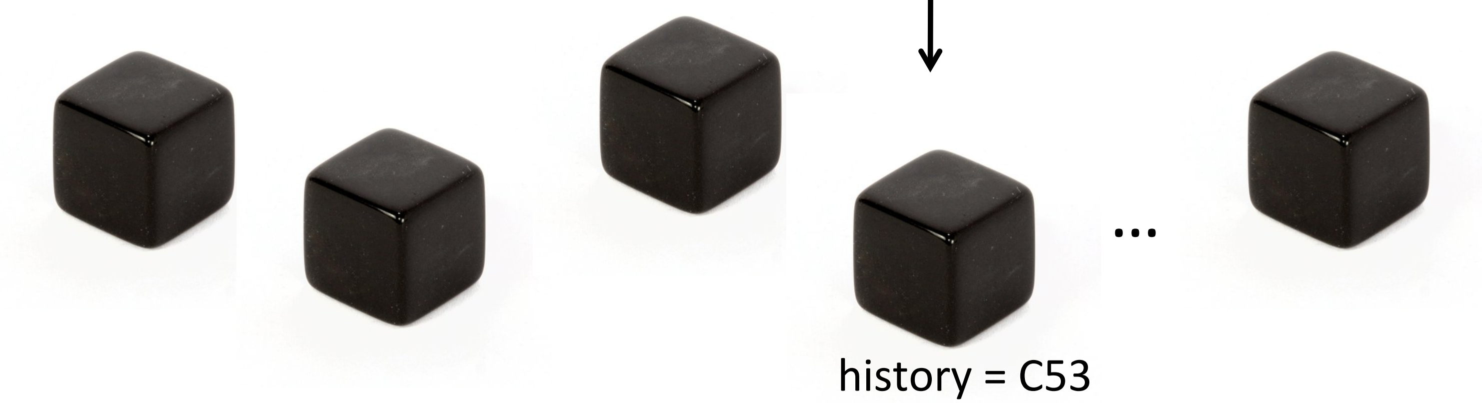


class = C53

start C53 C23

one “next class” die per class:

...



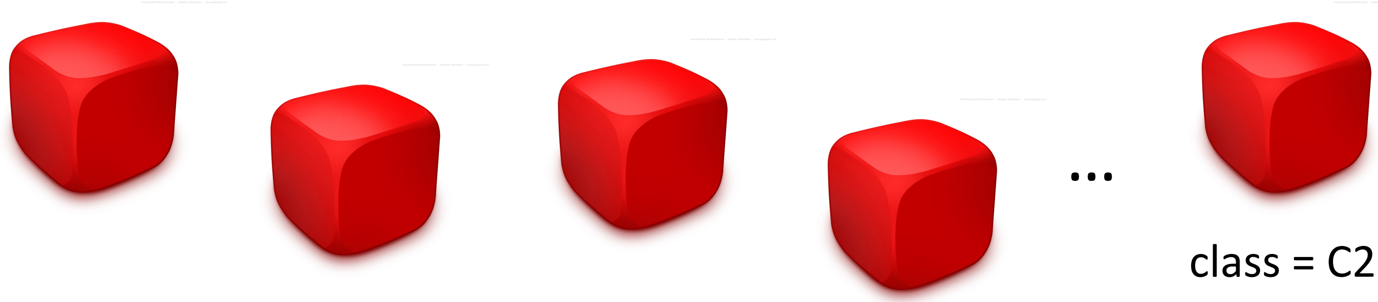
... ...

history = C53

start C53 $\boxed{\text{C23}}$

I want

one word die per class:

...  ... ...

class = C23

start C53 C23 C2

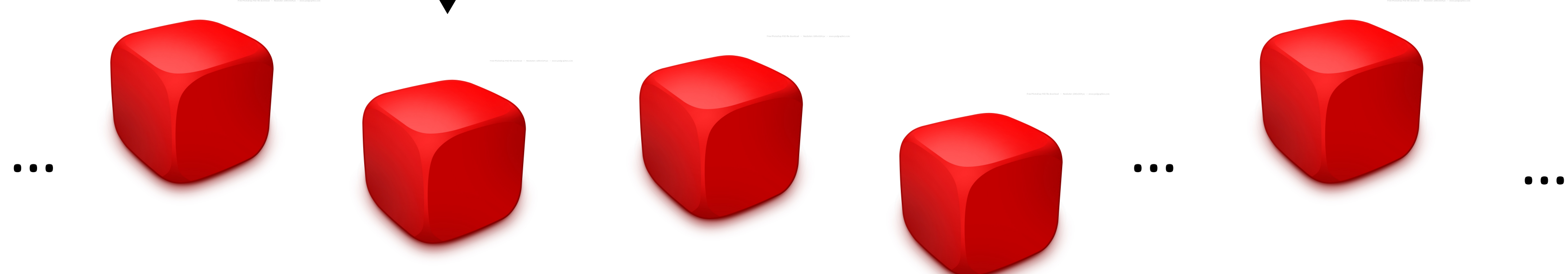
I want

one “next class” die per class:

start C53 C23 $\boxed{\text{C2}}$

I want a

one word die per class:



class = C2

start C53 C23 $\boxed{\text{C2}}$ C5

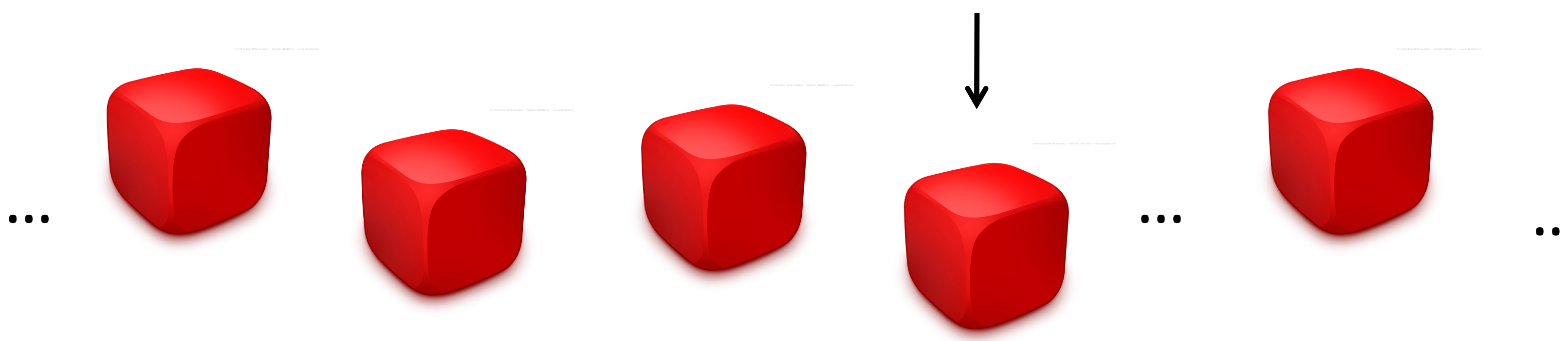
I want a

one “next class” die per class:

start C53 C23 C2 C5

I want a flight

one word die per class:



class = C5

Class-Based Sequence Models

- From Brown et al. (1990):

$$p(\text{start}, w_1, w_2, \ldots, w_n \text{stop}) \quad = \quad \prod_{i=1}^{n+1} \eta(w_i \mid \text{cl}(w_i)) \times \gamma(\text{cl}(w_i) \mid \text{cl}(w_{i-1}))$$

- Independence assumptions?
- Number of parameters?
- Generalization ability?

Lecture Outline

- ✓ Markov models
2. Hidden Markov models
3. Viterbi algorithm

HIDDEN MARKOV MODELS

Hidden Markov Model

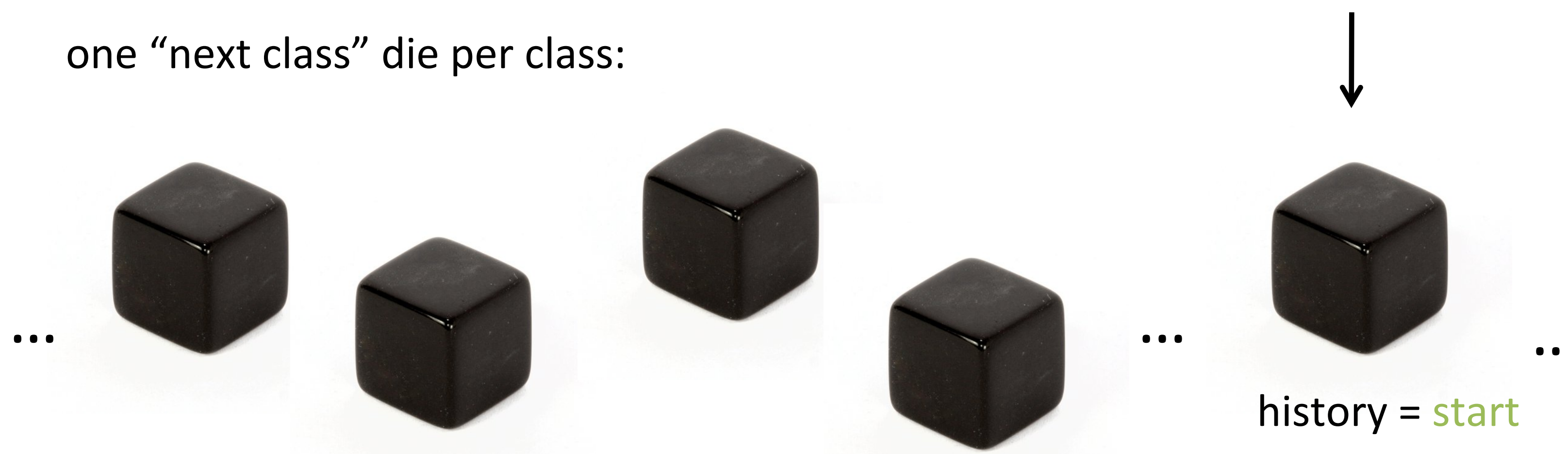
- A model over sequences of symbols, but there is missing information associated with each symbol: its "state."
  - Assume a finite set of possible states, Λ.

$$p(\text{start}, s_1, w_1, s_2, w_2, \ldots, s_n, w_n \text{stop}) = \prod_{i=1}^{n+1} \eta(w_i \mid s_i) \times \gamma(s_i \mid s_{i-1})$$

- A *joint* model over the observable symbols and their hidden/latent/unknown classes.

start C53

one “next class” die per class:



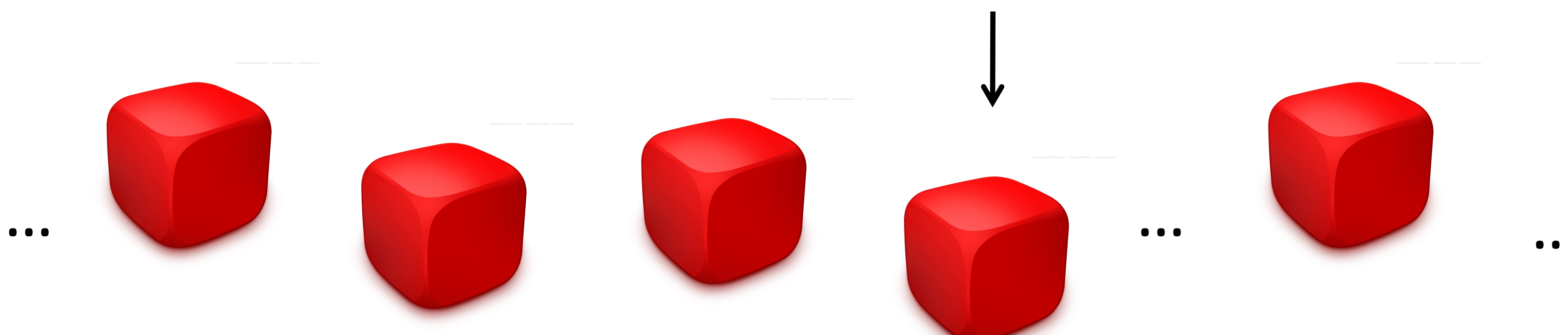
history = start

The only change to the class-based model is that now, the different word dice can *share words*!

start C53

I

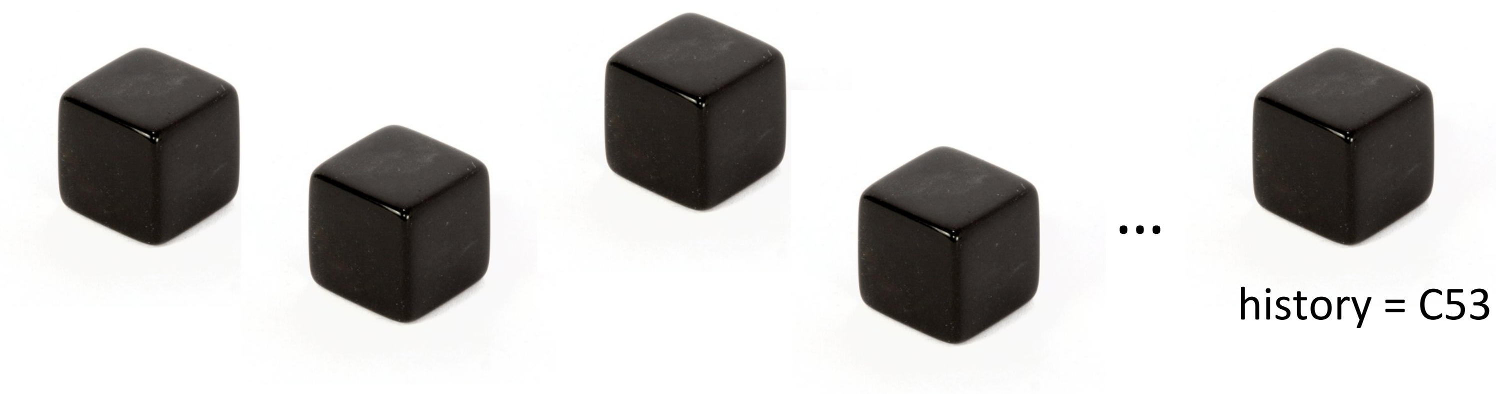
one word die per class:



class = C53

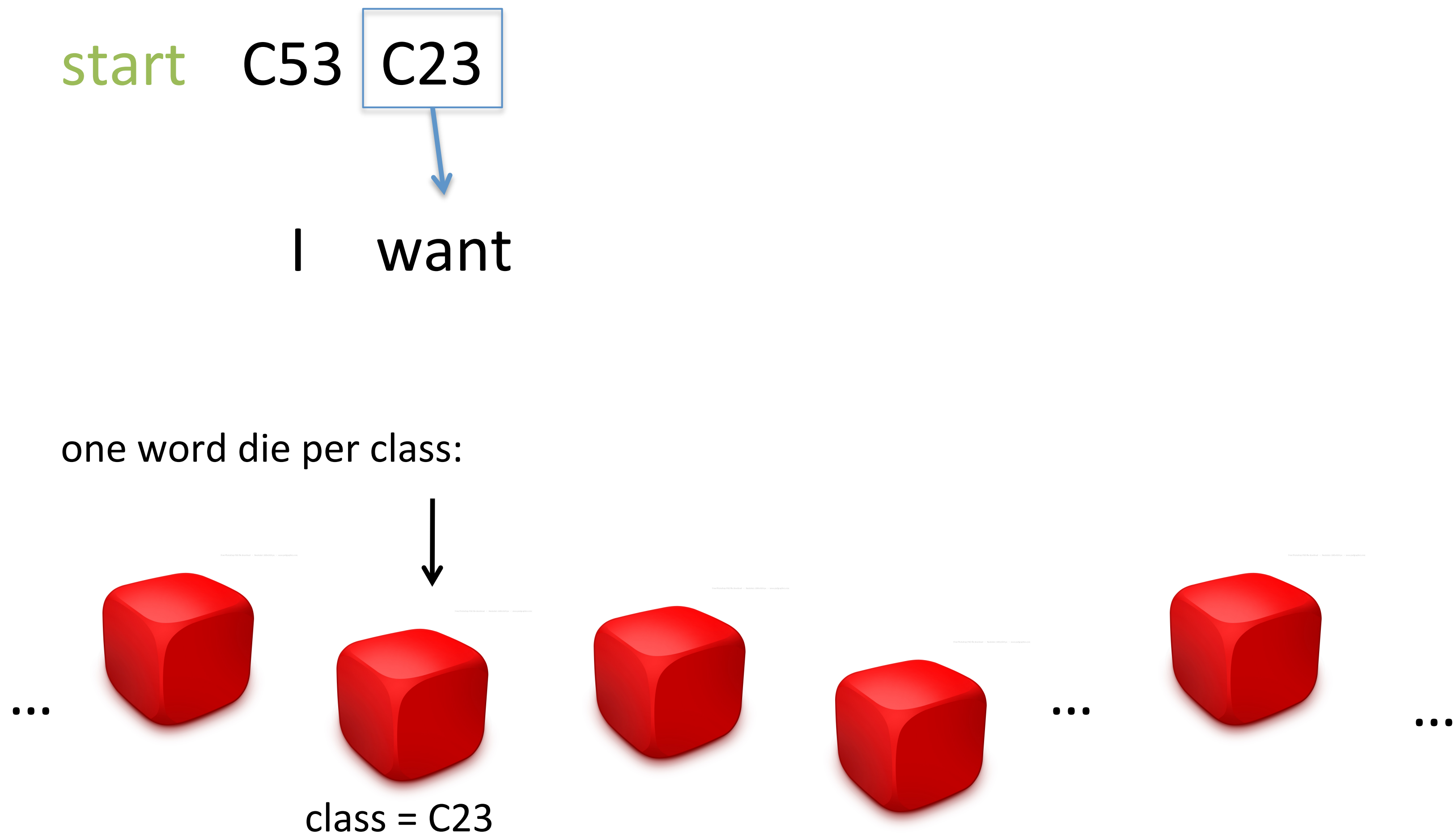
start C53 C23

one “next class” die per class:



... ...

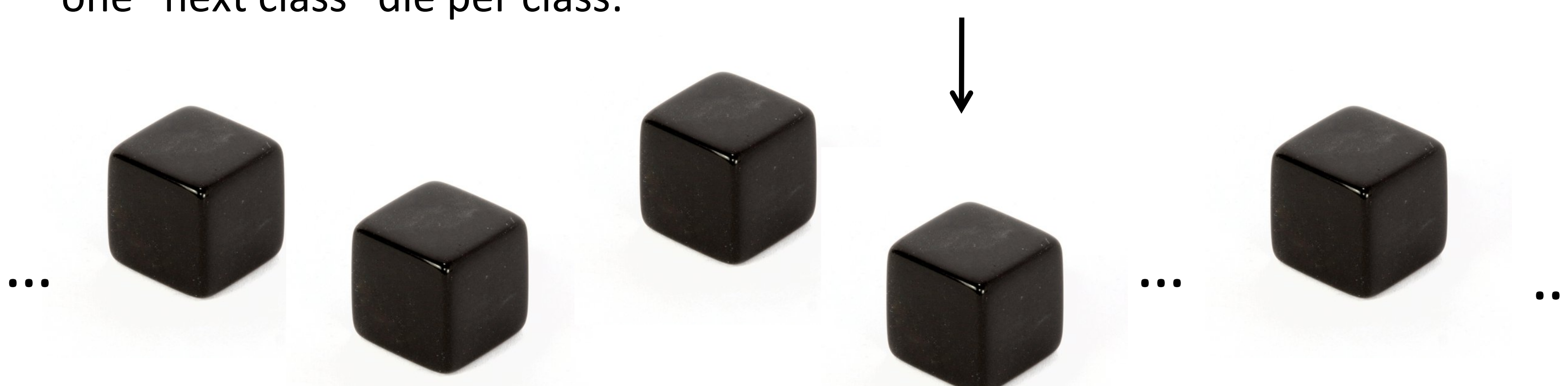
history = C53


start
C53
C23
I
want
one word die per class:
...
...
...
class = C23

start C53 $\boxed{\text{C23}}$ C2

I want

one “next class” die per class:

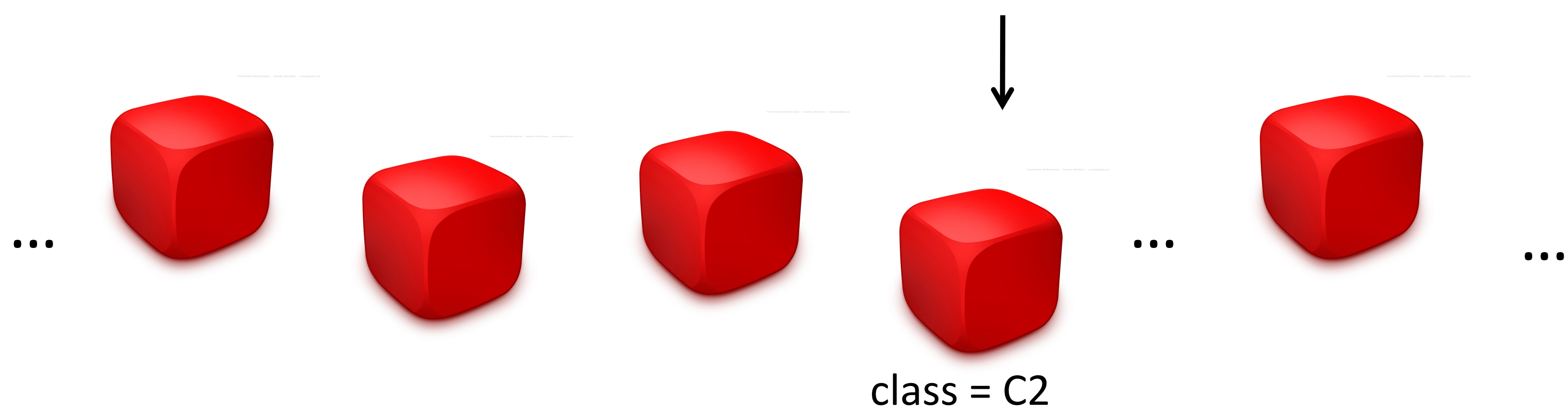


history = C23

start C53 C23 $\boxed{\text{C2}}$

I want a

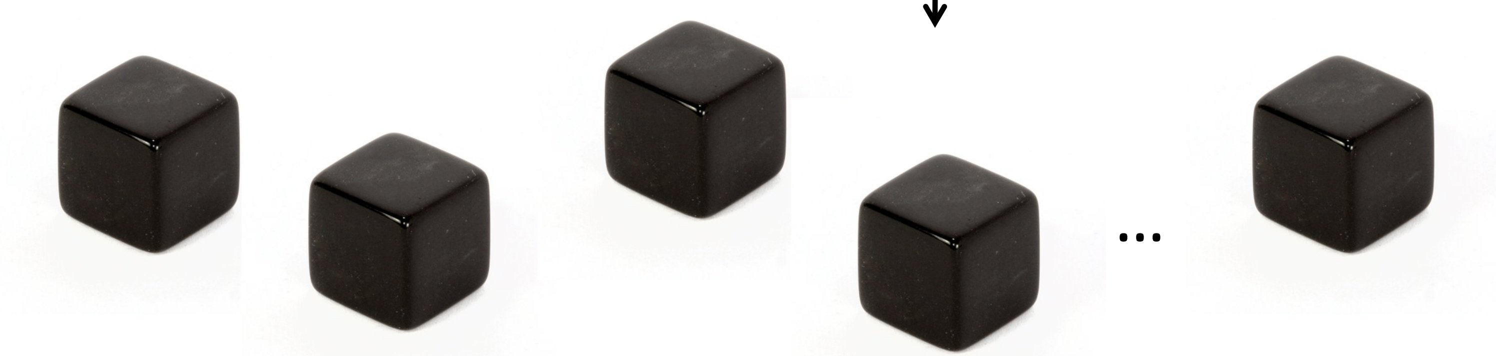
one word die per class:



start C53 C23 C2 C5

I want a

one “next class” die per class:

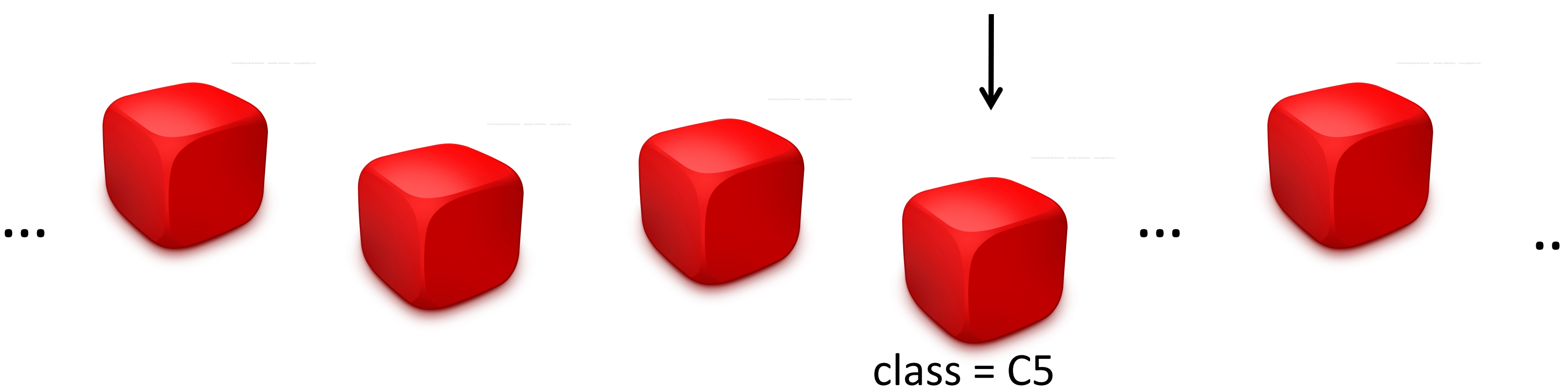
...  ... ...

history = C2

start C53 C23 C2 $\boxed{\text{C5}}$

I want a flight

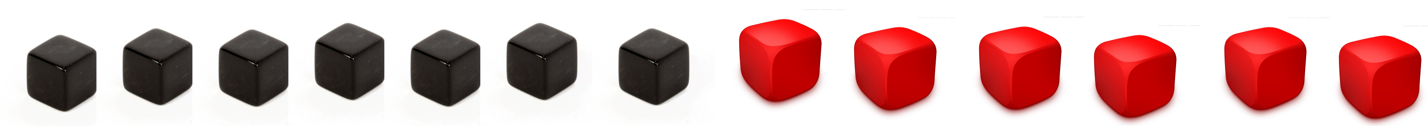
one word die per class:



Two Equivalent Stories

- First, as shown: transition, emit, transition, emit, transition, emit.

- Second:
  - Generate the sequence of transitions. Essentially, a Markov model on classes.
  - Stochastically replace each class with a word.



m^{th} Order Hidden Markov Models

- We can condition on a longer history of past states:

$$p(\text{start}, s_1, w_1, s_2, w_2, \ldots, s_n, w_n \text{stop}) \quad = \quad \prod_{i=1}^{n+1} \eta(w_i \mid s_i) \times \gamma(s_i \mid s_{i-m}, \ldots, s_{i-1})$$

- Number of parameters?
- Benefit: longer "memory."
- Today I will stick with first-order HMMs.

Uses of HMMs in NLP

- Part-of-speech tagging (Church, 1988; Brants, 2000)
- Named entity recognition (Bikel et al., 1999) and other information extraction tasks
- Text chunking and shallow parsing (Ramshaw and Marcus, 1995)
- Word alignment in parallel text (Vogel et al., 1996)

- Also popular in computational biology and central to speech recognition.

Part of Speech Tagging

After	paying	the	medical	bills	,	Frances	was	nearly	broke	.
RB	VBG	DT	JJ	NNS	,	NNP	VBZ	RB	JJ	.

- Adverb (RB)
- Verb (VBG, VBZ, and others)
- Determiner (DT)
- Adjective (JJ)
- Noun (NN, NNS, NNP, and others)
- Punctuation (., ,, and others)

Named Entity Recognition

With Commander Chris Ferguson at the helm ,

Atlantis touched down at Kennedy Space Center .

Named Entity Recognition

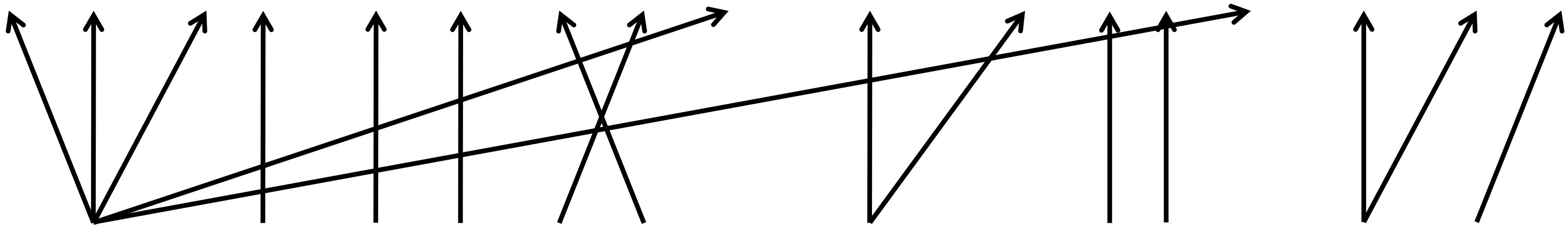
O	B-person	I-person	I-person	O	O	O	O
With	Commander	Chris	Ferguson	at	the	helm	,

B-space-shuttle	O	O	O	B-place	I-place	I-place	O
Atlantis	touched	down	at	Kennedy	Space	Center	.

- What makes this hard?

Word Alignment

Mr. President , Noah’s ark was filled not with production factors , but with living creatures.



NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah's ark was filled not with production factors , but with living creatures.

NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah's ark was filled not with production factors , but with living creatures.

NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah's ark was filled not with production factors , but with living creatures.

NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah’s ark was filled not with production factors , but with living creatures.

NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah’s ark was filled not with production factors , but with living creatures.

NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah's ark was filled not with production factors , but with living creatures.

NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah’s ark was filled not with production factors , but with living creatures.

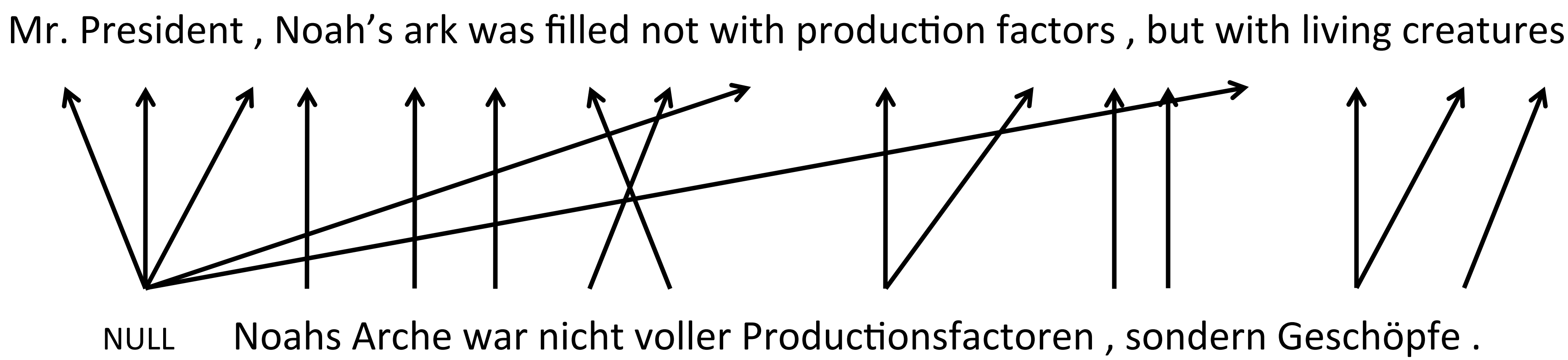
NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment

Mr. President , Noah's ark was filled not with production factors , but with living creatures.

NULL Noahs Arche war nicht voller Productionsfactoren , sondern Geschöpfe .

Word Alignment



Hidden Markov Model

- A model over sequences of symbols, but there is missing information associated with each symbol: its "state."
  - Assume a finite set of possible states, Λ.

$$p(\text{start}, s_1, w_1, s_2, w_2, \ldots, s_n, w_n \text{stop}) \quad = \quad \prod_{i=1}^{n+1} \eta(w_i \mid s_i) \times \gamma(s_i \mid s_{i-1})$$

- A *joint* model over the observable symbols and their hidden/latent/unknown classes.