Application: HMMs for Information Extraction (IE)

- IE: Text ➔ machine-understandable data

`Paris, the capital of France, …`

➔

(`Paris`, `France`) $\in$ `CapitalOf`, p=0.85

- Applied to Web: better search engines, semantic Web, step toward human-level AI

IE Automatically?

Intractable to get human labels for every concept expressed on the Web

Idea: extract from **semantically tractable** sentences

…**Edison invented the light bulb**…

(**Edison, light bulb**) ∈ **Invented**

$x\ V\ y \Rightarrow (x, y) \in V$

…**Bloomberg, mayor** of **New York City**…

⇒ (**Bloomberg, New York City**) ∈ **Mayor**

$x,\ C$ of $y \Rightarrow (x, y) \in C$

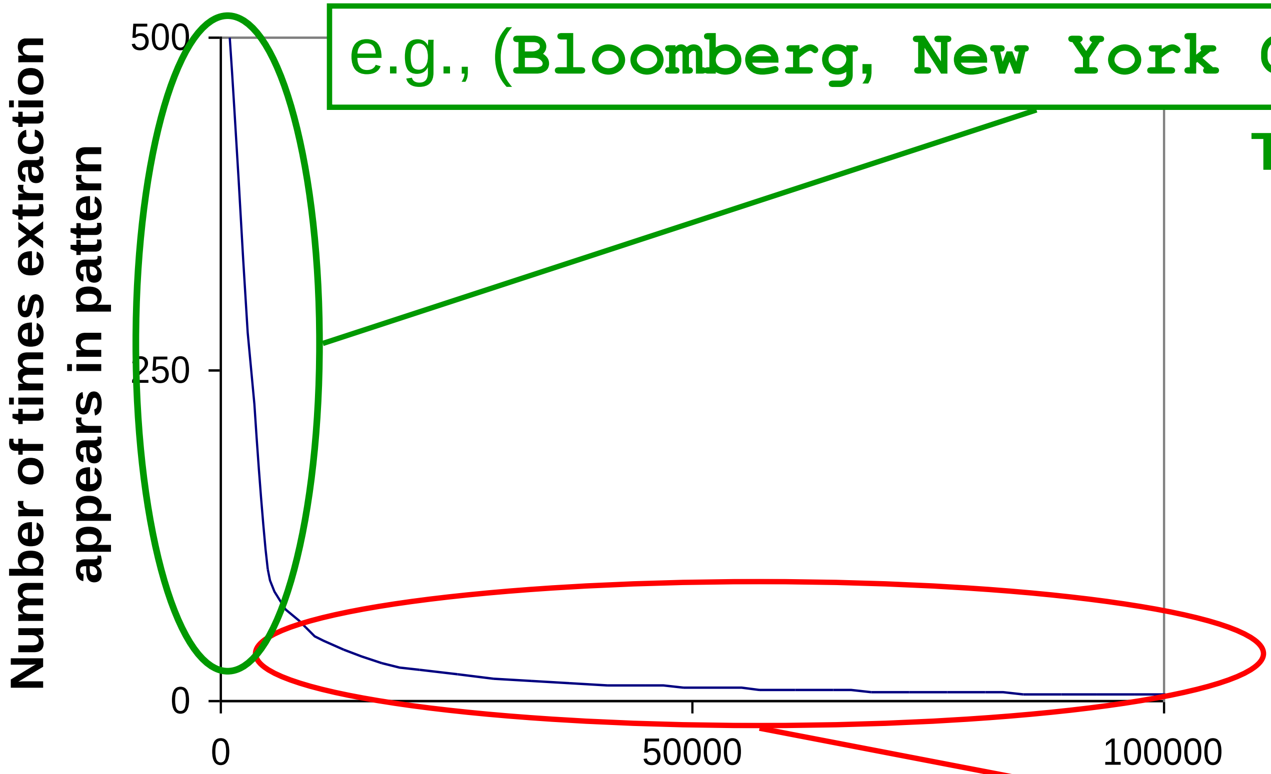
But…

Extraction patterns make errors:

"Erik Jonsson, CEO of **Texas Instruments**, **mayor** of **Dallas** from 1964-1971, and…"

- Empirical fact:
  - Extractions you see over and over tend to be correct
  - The problem is the "long tail"

Challenge: the “long tail”



e.g., (**Bloomberg, New York City**)

Tend to be correct

e.g., (**Dave Shaver, Pickerington**)
(**Ronald McDonald, McDonaldland**)

A *mixture* of correct and incorrect

Mayor McCheese

Assessing Sparse Extractions

Strategy

1) Model how common extractions occur in text
2) Rank sparse extractions by fit to model

The Distributional Hypothesis

- *Terms in the same class tend to appear in similar contexts.*

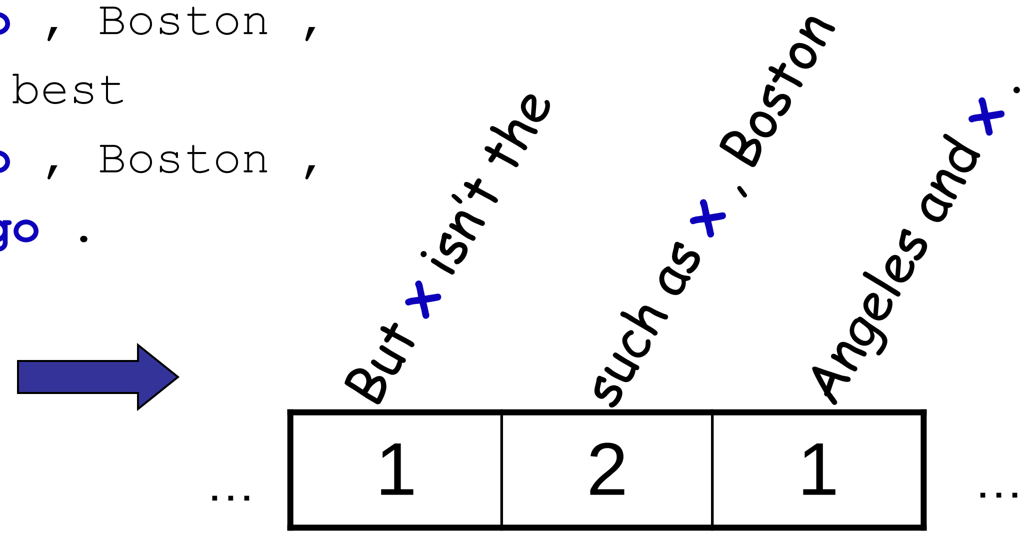
Context	Hits with **Chicago**	Hits with **Twisp**
"cities including __"	42,000	1
"__ and other cities"	37,900	0
"__ hotels"	2,000,000	1,670
"mayor of __"	657,000	82

HMM Language Models

- Precomputed – scalable
- Handle sparsity

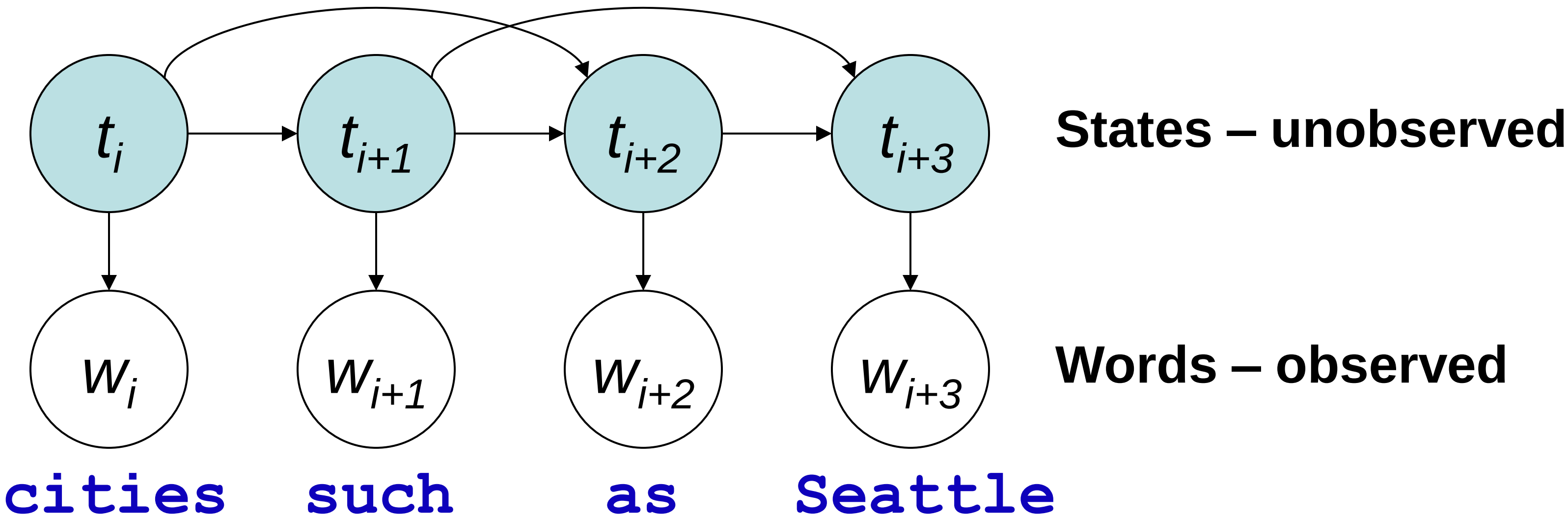
Baseline: context vectors

…
cities such as **Chicago** , Boston ,
But **Chicago** isn't the best
cities such as **Chicago** , Boston ,
Los Angeles and **Chicago** .
…



- Compute dot products between vectors of common and sparse extractions
  [*cf.* Ravichandran *et al.* 2005]

Hidden Markov Model (HMM)



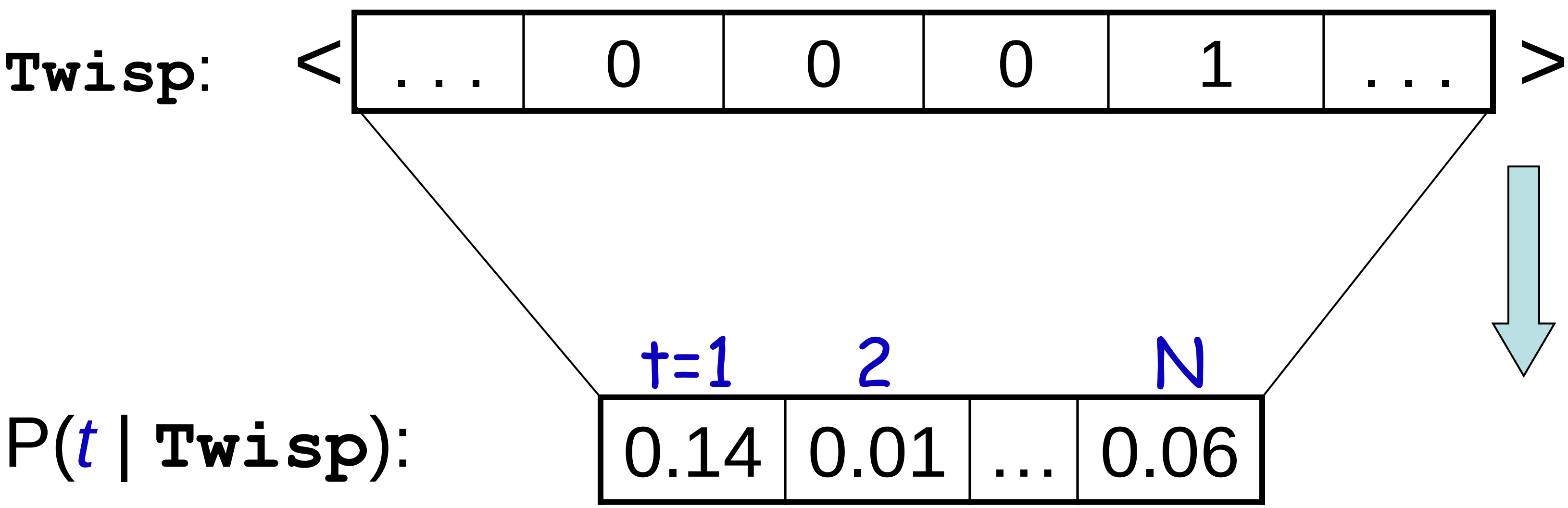
Hidden States $t_i \in \{1, \ldots, N\}$ (*N* fairly small)

Train on unlabeled data

– $P(t_i \mid w_i = w)$ is *N*-dim. distributional summary of *w*

– Compare extractions using KL divergence

HMM Compresses Context Vectors



Distributional Summary P(*t* | *w*)

- Compact (efficient – **10-50x** less data retrieved)
- Dense (accurate – **23-46%** error reduction)

Example

Is **Pickerington** of the same type as **Chicago**?

Chicago , Illinois

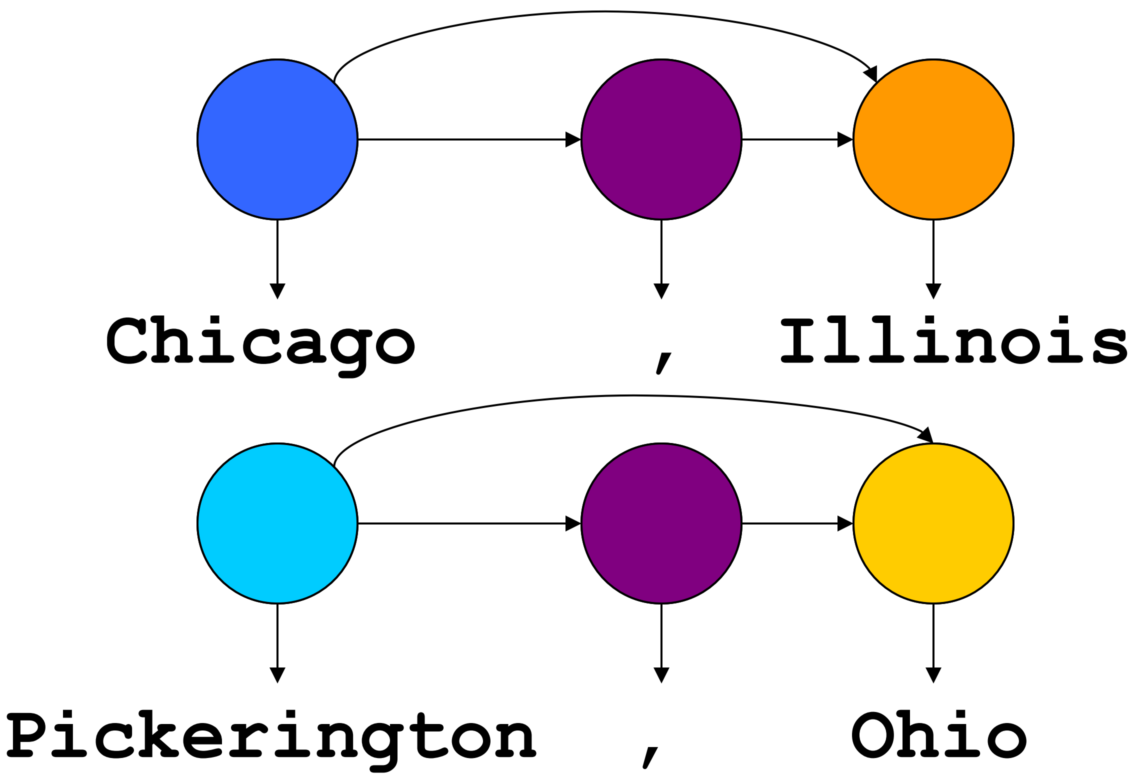
Pickerington , Ohio

	[x] , Illinois	[x] , Ohio	
Chicago:	291	0	…
Pickerington:	0	1	…

=> Context vectors say no, dot product is 0!

Example

HMM Generalizes:



Experimental Results

Task: Ranking sparse TextRunner extractions.

Metric: Area under precision-recall curve.

	`Headquartered`	`Merged`		**Average**
Frequency	0.710	0.784		0.713
PL	0.651	0.851	…	0.785
LM	**0.810**	**0.908**		**0.851**

Language models reduce missing area by **39%** over nearest competitor.

Example word distributions (1 of 2)

- P(word | state 3)
  - unk0 0.0244415
  - new 0.0235757
  - more 0.0123496
  - unk1 0.0119841
  - few 0.0114422
  - small 0.00858043
  - good 0.00806342
  - large 0.00736572
  - great 0.00728838
  - important 0.00710116
  - other 0.0067399
  - major 0.00628244
  - little 0.00545736
  - …
- P(word | state 24)
  - , 0.49014
  - . 0.433618
  - ; 0.0079789
  - -- 0.00365591
  - - 0.00302614
  - ! 0.00235752
  - : 0.001859

Example word distributions (2 of 2)

- P(word | state 1)
  - unk1 0.116254
  - United+States 0.012609
  - world 0.009212
  - U.S 0.007950
  - University 0.007243
  - Internet 0.007152
  - time 0.005167
  - end 0.004928
  - unk0 0.004818
  - war 0.004260
  - country 0.003774
  - way 0.003528
  - city 0.003429
  - US 0.003269
  - Sun 0.002982
  - Earth 0.002628
  -
- P(word | state 3)
  - the 0.863846
  - a 0.0131049
  - an 0.00960474
  - its 0.008541
  - our 0.00650477
  - this 0.00366675
  - unk1 0.00313899
  - your 0.00265876

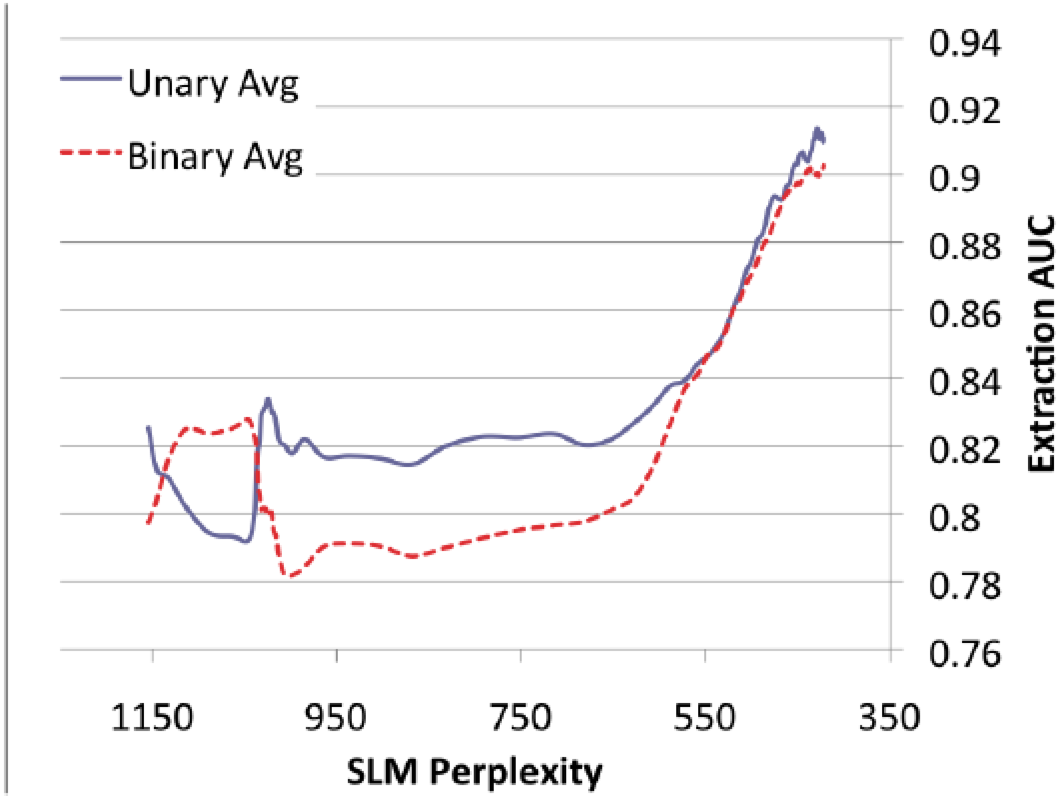
Correlation between LM and IE accuracy

Below: correlation coefficients

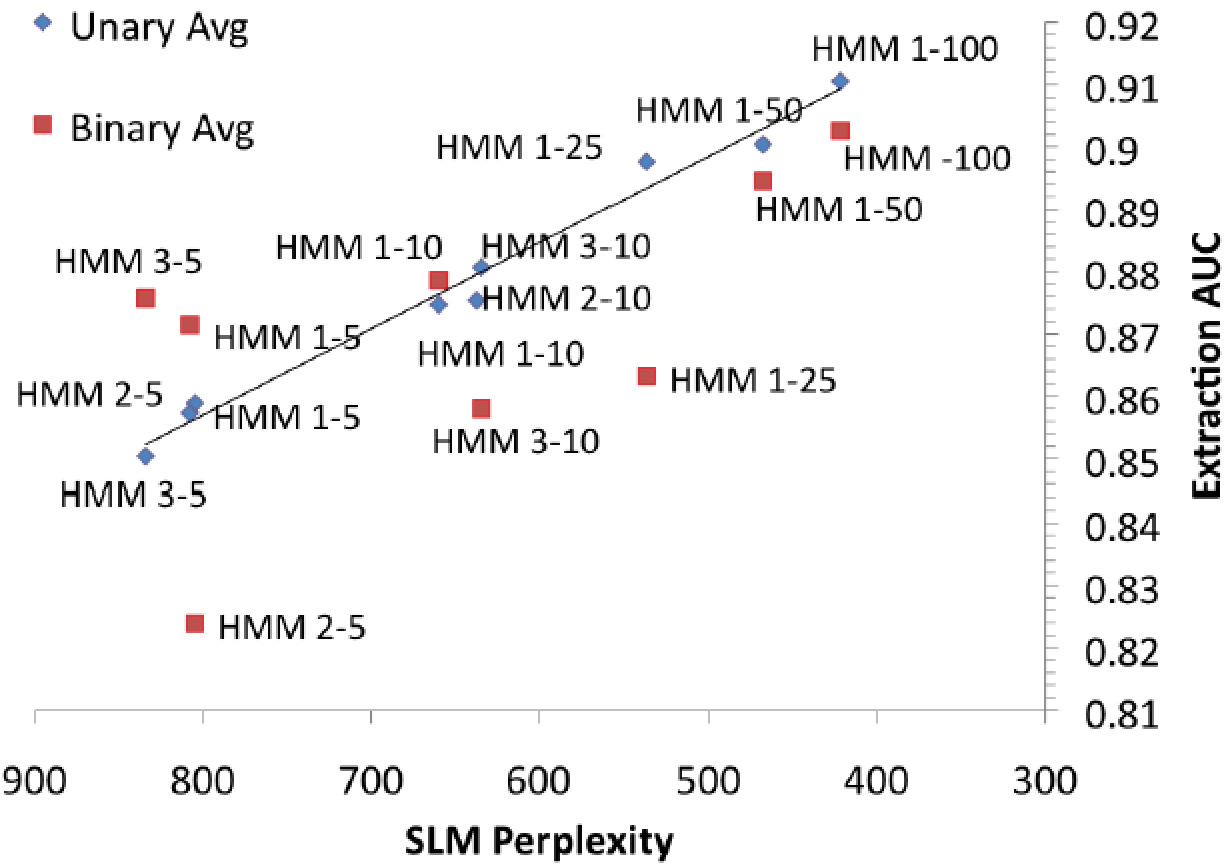
As LM error decreases, IE accuracy increases

LM	Unary	Binary	Wikipedia
HMM 1-5	-.911	-.361	-.994
HMM 2-5	-.856	.120	-.930
HMM 3-5	-.823	-.683	.922
HMM 1-10	-.916	-.967	-.905
HMM 2-10	-.877	-.797	-.963
HMM 3-10	-.957	-.669	-.924
HMM 1-25	-.933	-.850	-.959
HMM 1-50	-.942	-.942	-.947
HMM 1-100	-.896	-.877	-.942
N-Gram	-.512	-.999	-

Correlation between LM and IE accuracy



Correlation between LM and IE accuracy



What this suggests

- Better HMM language models => better information extraction

- Better HMM language models => … => human-level AI?
  - Consider: a good enough LM could do question answering, pass the Turing Test, etc.

- There are lots of paths to human-level AI, but LMs have:
  - Well-defined progress
  - Ridiculous amounts of training data

Also: active learning

- Today, people train language models by "taking what comes"
  - Larger corpora => better language models

- But corpus size limited by # of humans typing
  - What if we asked for the *most informative* sentences? (active learning)

What have we learned?

- In HMMs, general Bayes Net algorithms have simple & efficient form

1. **Evaluation**

   GIVEN a HMM M, and a sequence **x**,

   FIND Prob[**x** | M]

   Forward Algorithm and *Backward Algorithm* **(Variable Elimination)**

2. **Decoding**

   GIVEN a HMM M, and a sequence **x**,

   FIND the sequence π of states that maximizes P[**x**, π | M]

   Viterbi Algorithm **(MAP query)**

3. **Learning**

   GIVEN A sequence **x**,

   FIND HMM parameters $\theta = (e_i(.), a_{ij})$ that maximize P[**x** | θ]

   Baum-Welch/Forward-Backward algorithm **(EM)**

What have we learned?

- Unsupervised Learning of HMMs can power more scalable, accurate unsupervised IE