Natural Language Processing

AIMA Ch 23

[Some slides were adapted from slides by James H. Martin at U Colorado Boulder and by Albert Gatt at University of Malta]

Additional Reference

- [SLP] Speech and Language Processing, Daniel Jurafsky and James H. Martin
  - 2nd edition, 2008
  - 3rd edition, Oct. 2019

- Sequence labeling
  - [SLP 2nd ed.] Ch 5, 6
  - [SLP 3rd ed.] Ch 8, 9
- Parsing
  - [AIMA] Ch.23
  - [SLP 2nd ed.] Ch 12, 13, 14
  - [SLP 3rd ed.] Ch 12, 13, 14, 15

Natural Language Processing

- Get computers to perform useful and interesting tasks involving human languages.
  - Understanding
  - Generation
- Big applications
  - Question answering, conversational agents (ChatBot)
  - Financial document processing
  - Machine translation
  - News generation

Levels of NLP Research

Phonetics and phonology	knowledge about linguistic sounds
Morphology	knowledge of the meaningful components of words
Syntax	knowledge of the structural relationships between words
Lexical semantics	knowledge of word meaning
Compositional semantics	knowledge of the meaning of sentences
Pragmatics	knowledge of the relationship of meaning to the goals and intentions of the speaker
Discourse	knowledge about linguistic units larger than a single sentence

Sequence Labeling

Sequence Labeling

- Problem Definition
  - Known
    - A set of labels $C = \{c_1, c_2, \dots, c_J\}$
  - Input
    - Sentence $s = \{x^1, x^2, \dots, x^m\}$
  - Output
    - For each word x^i, predict a label $c^i \in C$

Examples

- Part-of-speech tagging
  - Input

Pierre	Vinken	,	61	years	old	,	will	join	…

  - Output

NNP	NNP	,	CD	NNS	JJ	,	MD	VB

NNP = Proper noun, singular

CD = Cardinal number

NNS = Noun, plural

JJ = Adjective

…

Examples

- Chinese word segmentation
  - Input

    瓦 里 西 斯 的 船 只 中 …
  - Output

B	I	I	E	S	B	E	S	
(瓦	里	西	斯)	(的)	(船	只)	(中)	…

B = beginning of a word

I = inside of a word

E = end of a word

S = single character word

Examples

- Named entity recognition
  - Input

Michael	Jeffrey	Jordan	was	born	in	Brooklyn	…

  - Output

B-PER	I-PER	E-PER	O	O	O	S-LOC

Michael Jeffrey Jordan — Person

Brooklyn — Location

B = beginning of an entity

I = inside of an entity

E = end of an entity

S = single word entity

O = outside of any entity

-PER = person

-LOC = location

-ORG = organization

…

Examples

- Semantic role labeling
  - Input

    The cat loves hats …

  - Output

    B-ARG0 E-ARG0 S-PRED S-ARG1

    The cat $\xleftarrow{\text{arg0}}$ loves $\xrightarrow{\text{arg1}}$ hats

B = beginning of an entity	-PRED = predicate
I = inside of an entity	-ARG0 = agent
E = end of an entity	-ARG1 = patient
S = single word entity	…
O = outside of any entity	

The simplest method

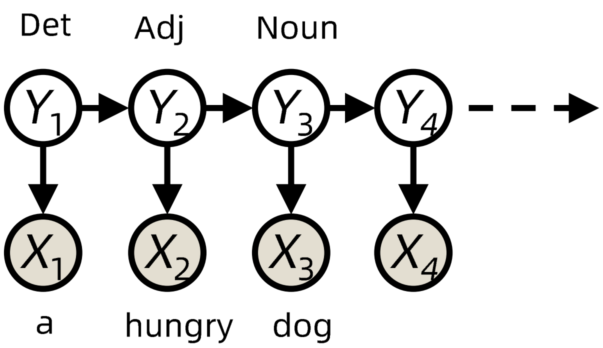
- For each word, predict its most frequent label
  - 90% accuracy on POS tagging!
  - Disadvantages:
    1. It does not consider the contextual info
       - “book a flight” vs. “read a book”
       - 我骑车差点摔倒，好在我一把把把把住了
    2. It does not consider relations between adjacent labels
       - In BIOES: "B-I" and "B-E" are OK, but "B-O" and "B-S" are not

Methods

- Hidden Markov Models (HMM)
- Max-Entropy Markov Models (MEMM)
- Conditional Random Fields (CRF)

Hidden Markov Model (HMM)

- Variables
  - X: word
  - Y: label (hidden state)
- Parameters
  - Transition model $P(y_t | y_{t-1})$
  - Emission model $P(x_t | y_t)$
  - Initial distribution $P(y_1)$
    - Can be seen as transition from Y_0=START to Y_1
  - Final distribution $P(y_n)$
    - Can be seen as transition from Y_n to Y_{n+1}=STOP



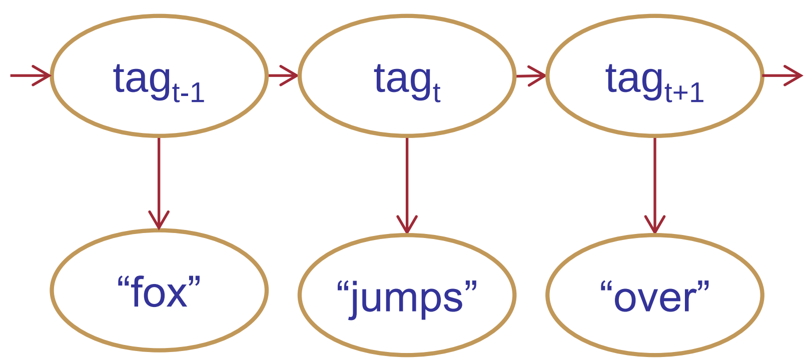
HMM Example

Transition

Y_{t-1}	$P(Y_t \mid Y_{t-1})$			
	N	V	P	…
START	0.5	0.1	0.1	…
N	0.4	0.3	0.1	…
V	0.5	0	0.3	…
P	0.3	0.1	0	…
…	…	…	…	…

Emission

Y_t	$P(X_t \mid Y_t)$			
	"fox"	"dog"	"run"	…
N	0.02	0.03	0.01	…
V	0	0	0.05	…
P	0	0	0	…
…	…	…	…	…



HMM Inference

- Find the most likely label sequence of the input sentence
  - $\arg\max_{y_{0:t}} P(y_{0:t} \mid x_{1:t})$
- Algorithm?
  - Viterbi algorithm

$$\boldsymbol{m}_{1:t+1} = \textsc{Viterbi}(\boldsymbol{m}_{1:t}, e_{t+1})$$

$$= P(e_{t+1}|X_{t+1}) \max_{x_t} P(X_{t+1} \mid x_t)\, \boldsymbol{m}_{1:t}[x_t]$$

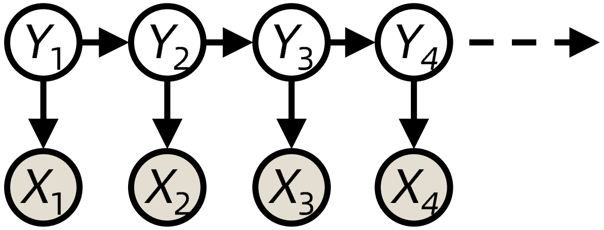
Beyond HMM

- The simplest method: for each word, predict its most frequent label
  - Problems:

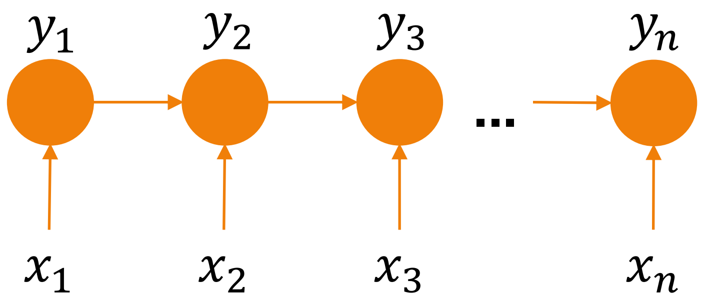
☹ 1. It does not consider the contextual info

☺ 2. It does not consider relations between adjacent labels

- HMM handles problem 2, but not 1



Max-Entropy Markov Models (MEMM)



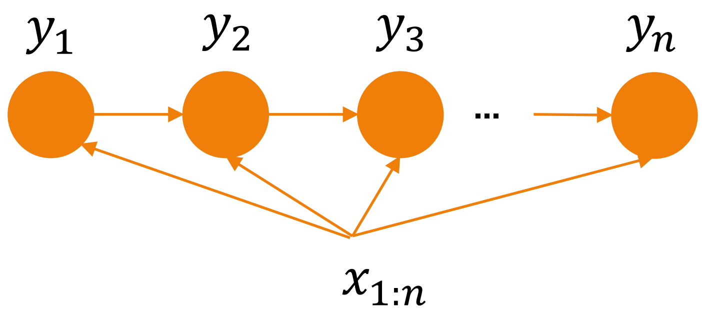
$$P(y_{1:n}|x_{1:n}) = P(y_1|x_1) \prod_{t=2}^{n} P(y_t|y_{t-1}, x_t)$$

$$P(y_t|y_{t-1}, x_t) = \frac{\exp(W^T \boxed{f(y_{t-1}, y_t, x_t)})}{Z(y_{t-1}, x_t)}$$

Possible features:

- y_{t-1} is B and y_t is E?
- y_{t-1} is B and y_t is O?
- x_t is a noun?
- x_t is capitalized?
- …

Max-Entropy Markov Models (MEMM)

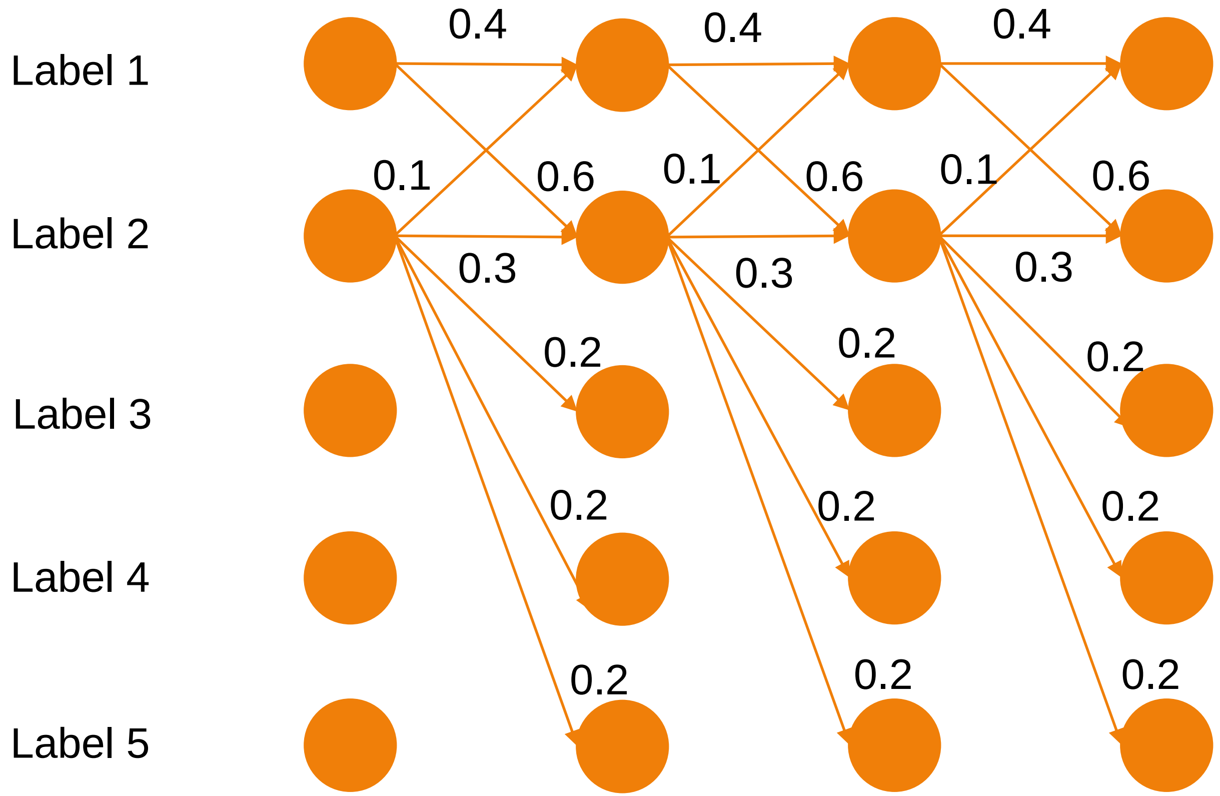


$$P(y_{1:n}|x_{1:n}) = P(y_1|x_{1:n}) \prod_{t=2}^{n} P(y_t|y_{t-1}, x_{1:n})$$

$$P(y_t|y_{t-1}, x_{1:n}) = \frac{\exp(W^T f(y_{t-1}, y_t, x_{1:n}))}{Z(y_{t-1}, x_{1:n})}$$

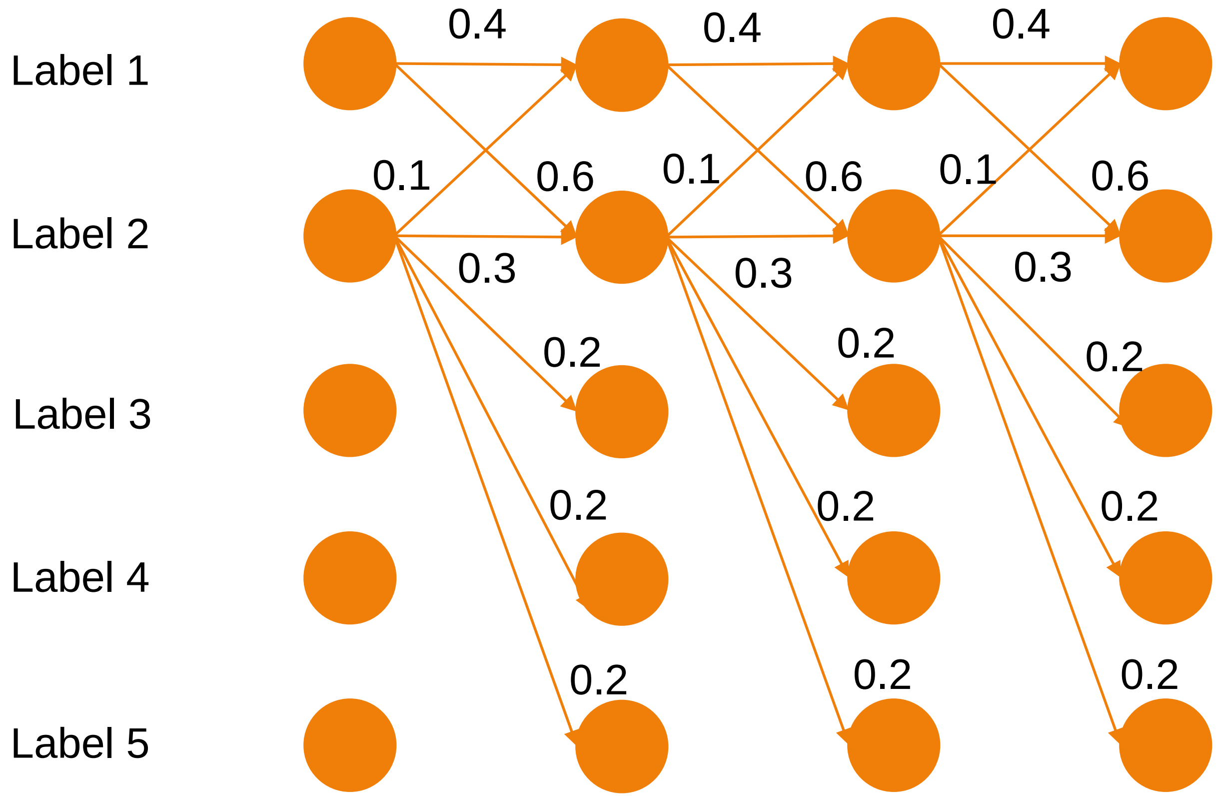
- MEMM considers both contextual info and relations between adjacent labels!
- But… MEMM suffers from label bias problem

Label Bias Problem



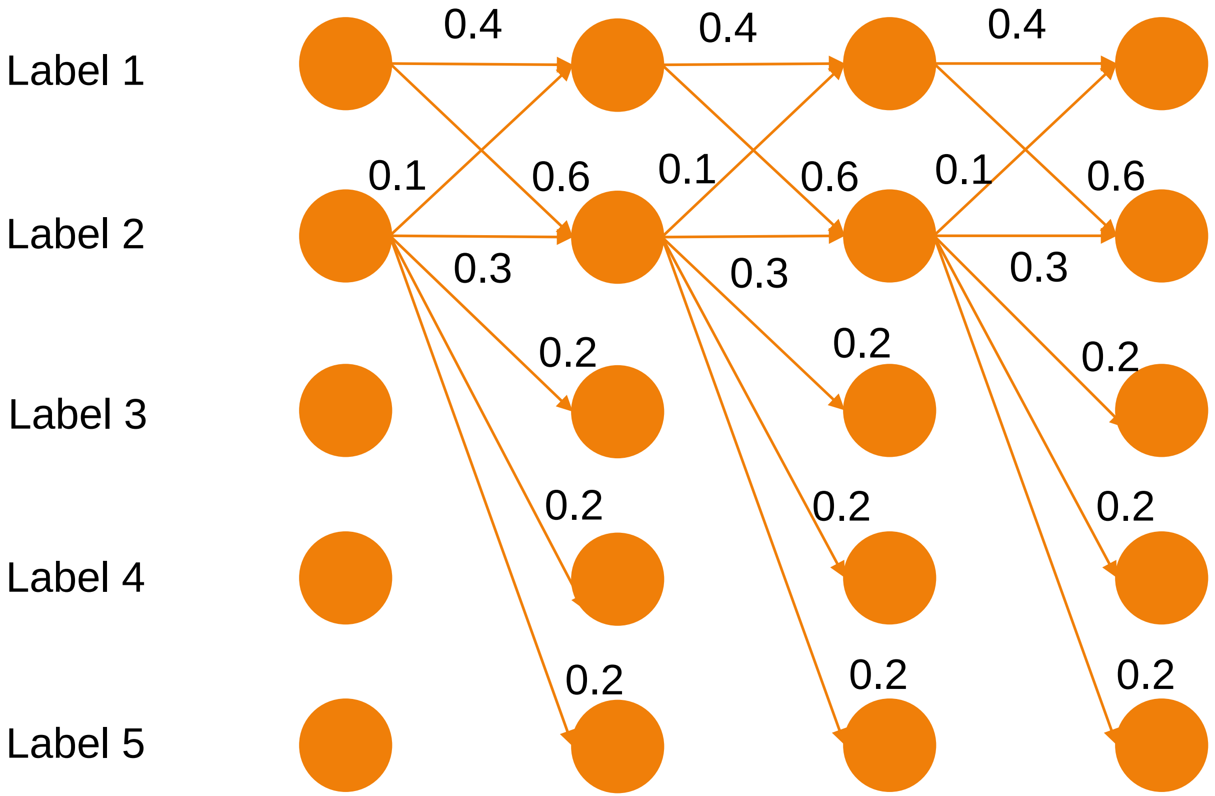
- What the local transition probabilities say:
  - Label 1 prefers to go to label 2
  - Label 2 prefers to stay at label 2

Label Bias Problem



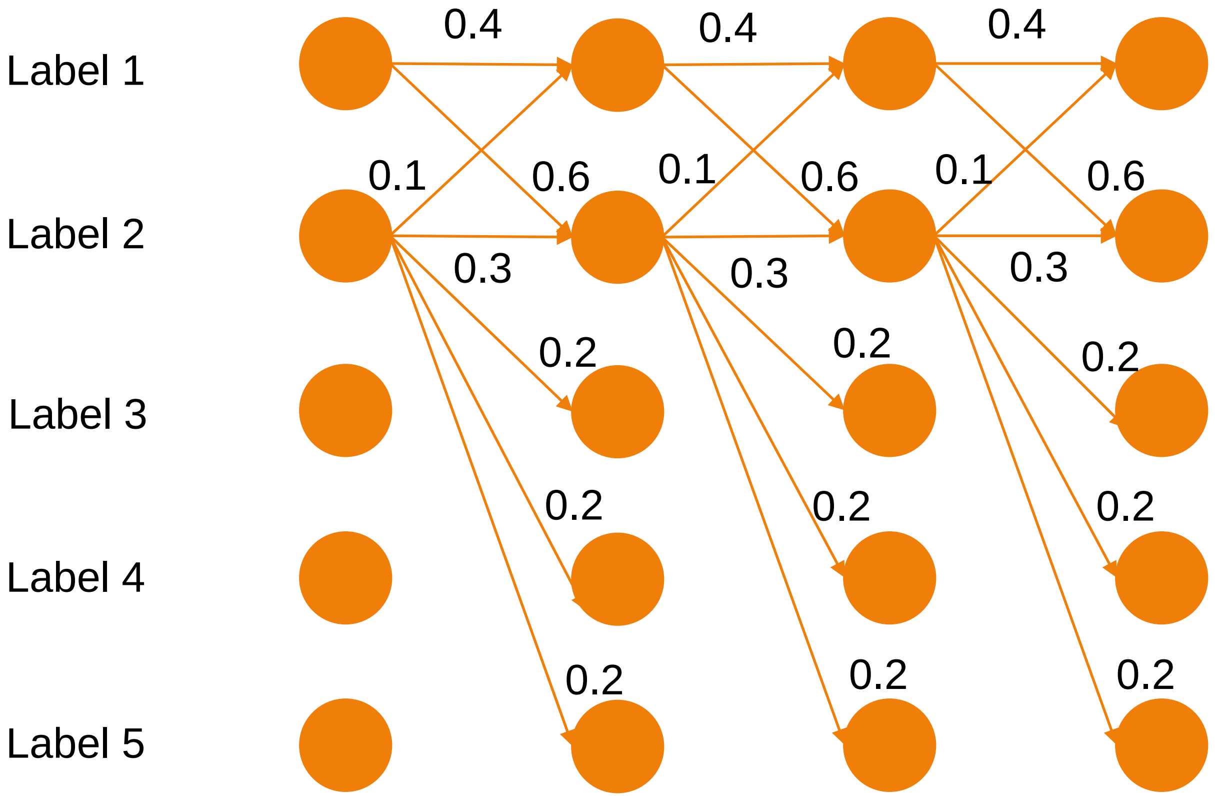
- $\boxed{P(1 \rightarrow 1 \rightarrow 1 \rightarrow 1) = 0.4^3 = 0.064}$
- $P(1 \rightarrow 2 \rightarrow 1 \rightarrow 2) = 0.6*0.1*0.6 = 0.036$
- $P(2 \rightarrow 2 \rightarrow 2 \rightarrow 2) = 0.3^3 = 0.027$
- $P(2 \rightarrow 1 \rightarrow 2 \rightarrow 1) = 0.1*0.6*0.1 = 0.006$

Label Bias Problem



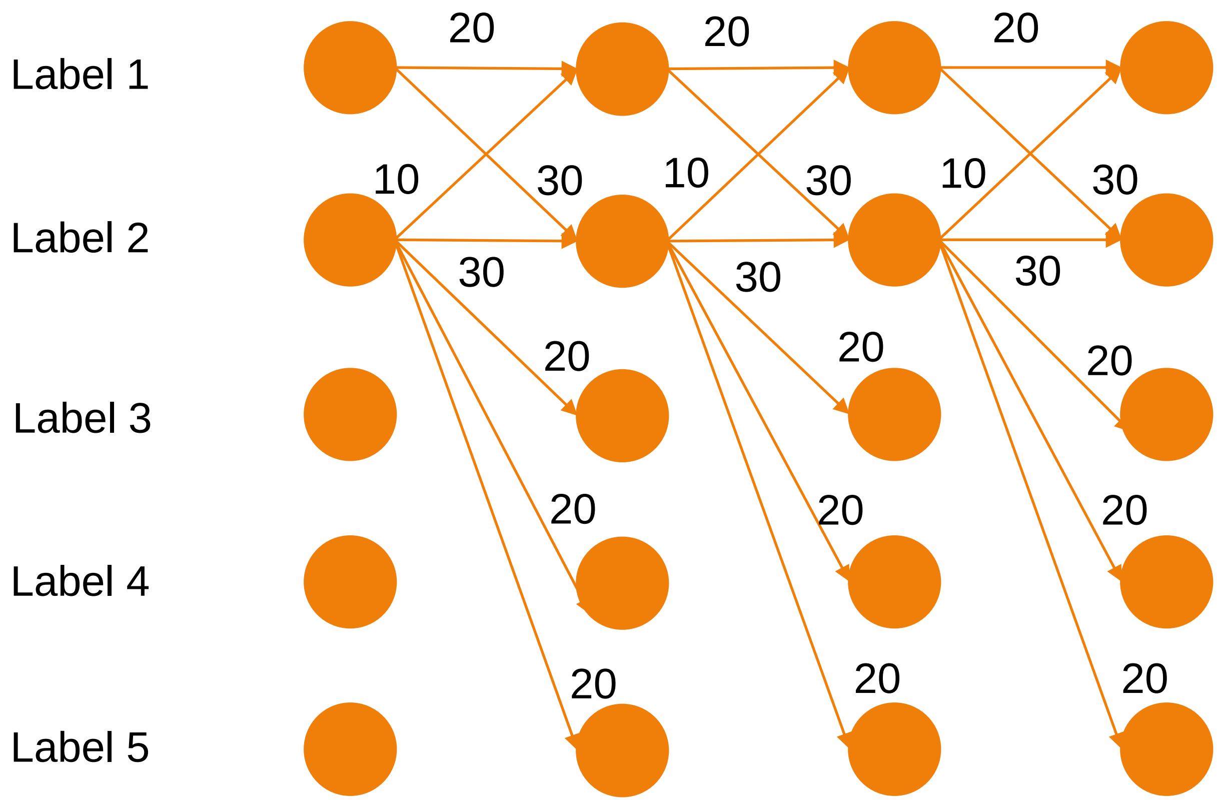
- Label 1 has only two transitions but label 2 has five
- Transition probabilities from label 2 are lower

Label Bias Problem



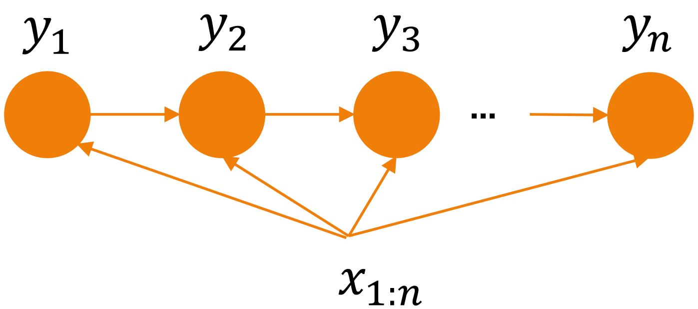
- Label bias in MEMM
  - Preference of states with lower number of transitions

Label Bias Problem



- Solution
  - From local probabilities to local potentials

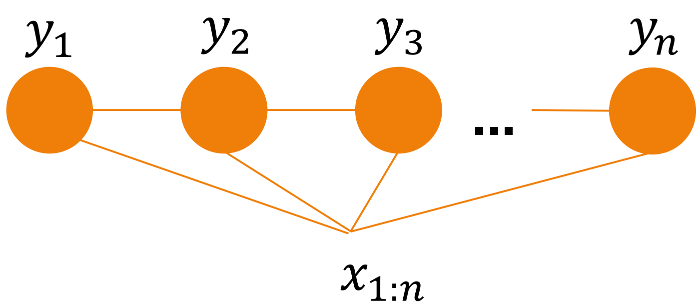
From MEMM to CRF



$$P(y_{1:n}|x_{1:n}) = P(y_1|x_{1:n}) \prod_{t=2}^{n} P(y_t|y_{t-1}, x_{1:n})$$

$$P(y_t|y_{t-1}, x_{1:n}) = \frac{\exp(W^T f(y_{t-1}, y_t, x_{1:n}))}{Z(y_{t-1}, x_{1:n})}$$

From MEMM to CRF



$$P(y_{1:n}|x_{1:n}) = \frac{1}{Z(x_{1:n})}\prod_{t=1}^{n}\exp(W^T f(y_{t-1}, y_t, x_{1:n}))$$

- Conditional Random Field (CRF) is an undirected graphical model
  - Global normalization instead of local normalization
  - Inference: Viterbi

Summary

- Sequence labeling
  - Predict a label for each word of a sentence
  - Many NLP tasks can be seen as sequence labeling
- Methods
  - HMM
  - MEMM
  - CRF

Parsing

Formal Grammars

Constituency

- Constituents
  - Groups of words within sentences can be shown to act as single units.
  - Ex: (The fox)(jumps (over (the dog)))
- These units form coherent classes
  - Units in the same class behave in similar ways
    - …with respect to their *internal* structure
    - …and with respect to other (*external*) units in the language
  - E.g., noun phrases

Constituency

- For example, it makes sense to say that the following are all ***noun phrases*** in English...

Harry the Horse	a high-class spot such as Mindy's
the Broadway coppers	the reason he comes into the Hot Box
they	three parties from Brooklyn

- Why?
  - Similar internal structures
    - e.g., determiner + modifier + noun + modifier
  - They can all precede verbs (external evidence)

Grammars and Constituency

- Grammar
  - the set of constituents and the rules that govern how they combine
- Lots of different theories of grammar
- Context-free grammars (CFGs)
  - Also known as: Phrase structure grammars
  - One of the simplest and most basic grammar formalisms

Context-Free Grammars

- A context-free grammar has four components
  - A set Σ of terminals (words)
  - A set N of nonterminals (phrases)
  - A start symbol $S \in N$
  - A set R of production rules
    - Specifies how a nonterminal can produce a string of terminals and/or nonterminals

Example Grammar

Grammar Rules			Examples
S	$\rightarrow$	$NP\ VP$	I + want a morning flight
NP	$\rightarrow$	$Pronoun$	I
	$\mid$	$Proper\text{-}Noun$	Los Angeles
	$\mid$	$Det\ Nominal$	a + flight
$Nominal$	$\rightarrow$	$Nominal\ Noun$	morning + flight
	$\mid$	$Noun$	flights
VP	$\rightarrow$	$Verb$	do
	$\mid$	$Verb\ NP$	want + a flight
	$\mid$	$Verb\ NP\ PP$	leave + Boston + in the morning
	$\mid$	$Verb\ PP$	leaving + on Thursday
PP	$\rightarrow$	$Preposition\ NP$	from + Los Angeles

Example Grammar

$$
\begin{aligned}
\textit{Noun} &\rightarrow \textit{flights} \mid \textit{breeze} \mid \textit{trip} \mid \textit{morning} \\
\textit{Verb} &\rightarrow \textit{is} \mid \textit{prefer} \mid \textit{like} \mid \textit{need} \mid \textit{want} \mid \textit{fly} \\
\textit{Adjective} &\rightarrow \textit{cheapest} \mid \textit{non-stop} \mid \textit{first} \mid \textit{latest} \\
&\quad \mid \textit{other} \mid \textit{direct} \\
\textit{Pronoun} &\rightarrow \textit{me} \mid \textit{I} \mid \textit{you} \mid \textit{it} \\
\textit{Proper-Noun} &\rightarrow \textit{Alaska} \mid \textit{Baltimore} \mid \textit{Los Angeles} \\
&\quad \mid \textit{Chicago} \mid \textit{United} \mid \textit{American} \\
\textit{Determiner} &\rightarrow \textit{the} \mid \textit{a} \mid \textit{an} \mid \textit{this} \mid \textit{these} \mid \textit{that} \\
\textit{Preposition} &\rightarrow \textit{from} \mid \textit{to} \mid \textit{on} \mid \textit{near} \\
\textit{Conjunction} &\rightarrow \textit{and} \mid \textit{or} \mid \textit{but}
\end{aligned}
$$

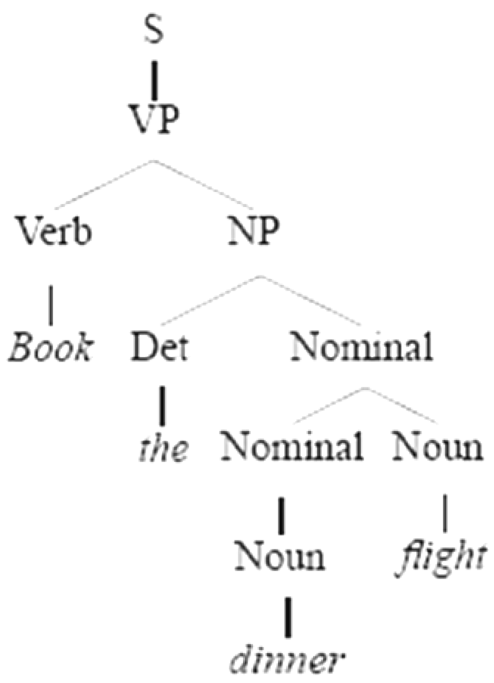
Sentence Generation

- A grammar can be used to generate a string
  - starting from a string containing only the start symbol S
  - recursively applying the rules to rewrite the string
  - until the string contains only terminals
- The generative process specifies the grammatical structure (parse tree) of the string

Example

$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$
$S \rightarrow VP$
$NP \rightarrow Pronoun$
$NP \rightarrow Proper\text{-}Noun$
$NP \rightarrow Det\ Nominal$
$NP \rightarrow Nominal$
$Nominal \rightarrow Noun$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$
$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow Verb\ NP\ NP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$

……



Book the dinner flight

Sentence Parsing

- Parsing is the process of taking a string and a grammar and returning one or more parse tree(s) for that string
  - If no parse tree can be found, then the string does not belong to the language
  - Parsing algorithms: CYK, Earley, etc.
    - To be introduced later

Probabilistic Grammars

- Also called stochastic grammars
- Each rule is associated with a probability

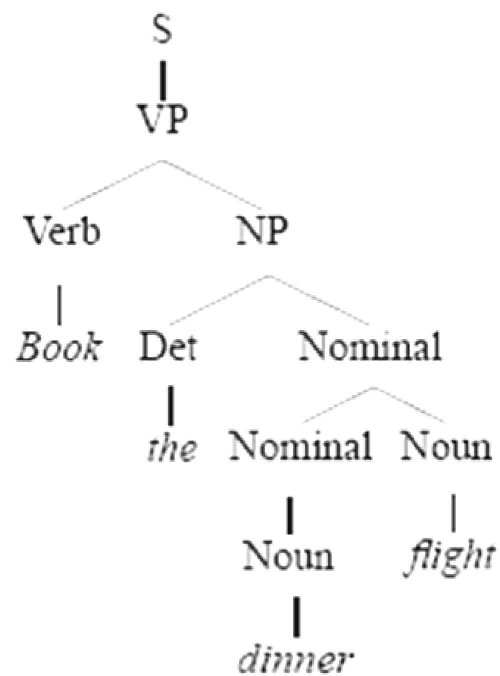
$$\alpha \rightarrow \beta : P(\alpha \rightarrow \beta | \alpha)$$

- The probability of a parse tree is the product of the probabilities of all the rules used in generating the parse tree

Example

S	$\rightarrow$	$NP\ VP$	[.80]
S	$\rightarrow$	$Aux\ NP\ VP$	[.15]
S	$\rightarrow$	VP	[.05]
NP	$\rightarrow$	$Pronoun$	[.35]
NP	$\rightarrow$	$Proper\text{-}Noun$	[.30]
NP	$\rightarrow$	$Det\ Nominal$	[.20]
NP	$\rightarrow$	$Nominal$	[.15]
$Nominal$	$\rightarrow$	$Noun$	[.75]
$Nominal$	$\rightarrow$	$Nominal\ Noun$	[.20]
$Nominal$	$\rightarrow$	$Nominal\ PP$	[.05]
VP	$\rightarrow$	$Verb$	[.35]
VP	$\rightarrow$	$Verb\ NP$	[.20]
VP	$\rightarrow$	$Verb\ NP\ PP$	[.10]
VP	$\rightarrow$	$Verb\ PP$	[.15]
VP	$\rightarrow$	$Verb\ NP\ NP$	[.05]
VP	$\rightarrow$	$VP\ PP$	[.15]
PP	$\rightarrow$	$Preposition\ NP$	[1.0]

……



Book the dinner flight

$P(T)$ = .05 × .20 × .20 × .20 × .75 × .30 × .60 × .10 × .40 = 2.2×10^{-6}

Ambiguity

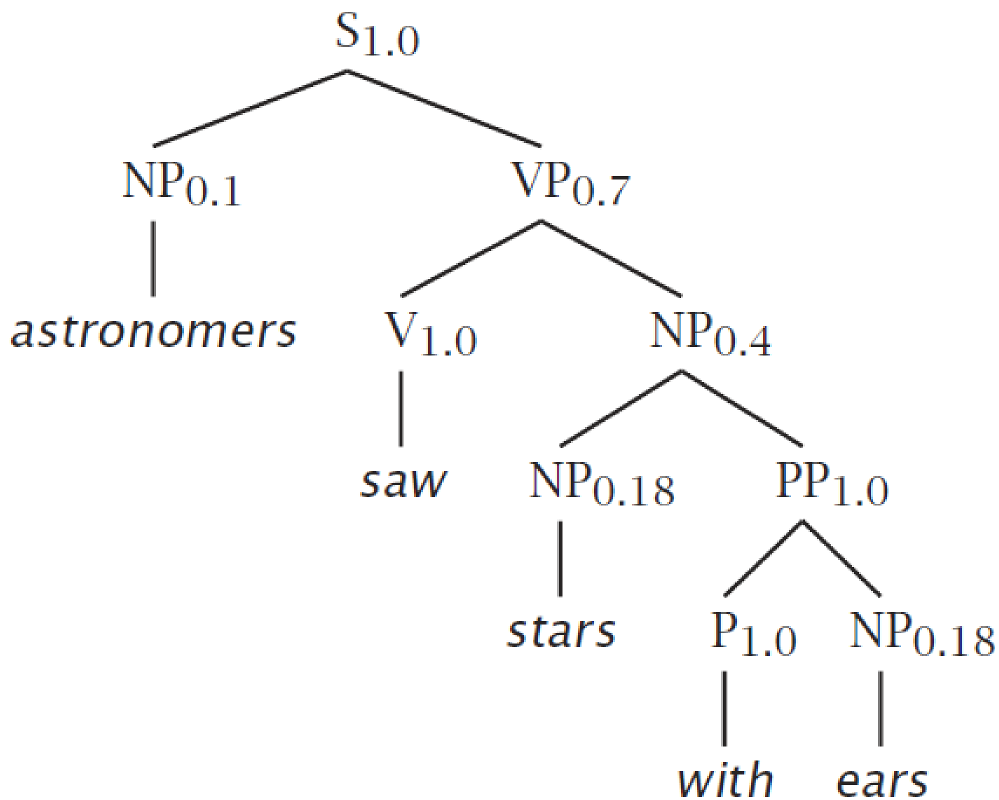
- A sentence is ambiguous if it has more than one possible parse tree
  - …and hence more than one interpretation
- Examples
  - Time flies like an arrow.
  - Astronomers saw stars with ears.

Example

$S \rightarrow NP\ VP$	1.0	$NP \rightarrow NP\ PP$	0.4
$PP \rightarrow P\ NP$	1.0	$NP \rightarrow$ *astronomers*	0.1
$VP \rightarrow V\ NP$	0.7	$NP \rightarrow$ *ears*	0.18
$VP \rightarrow VP\ PP$	0.3	$NP \rightarrow$ *saw*	0.04
$P \rightarrow$ *with*	1.0	$NP \rightarrow$ *stars*	0.18
$V \rightarrow$ *saw*	1.0	$NP \rightarrow$ *telescopes*	0.1

Example

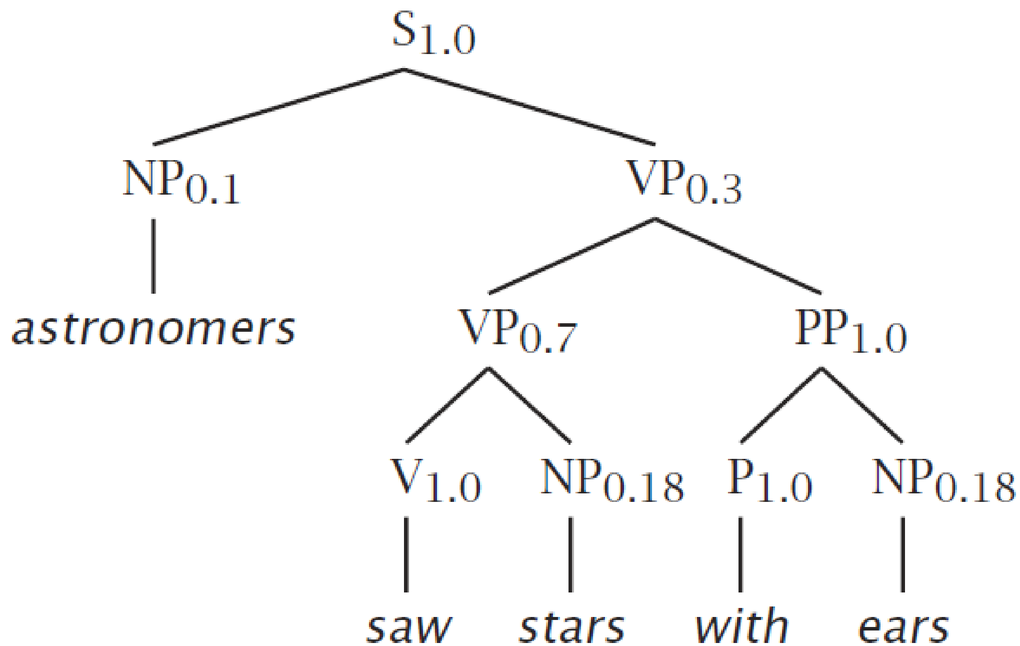
t_1:



$$
\begin{aligned}
P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0009072
\end{aligned}
$$

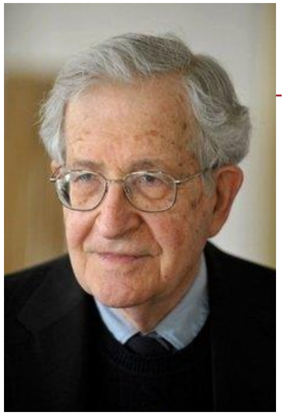
Example

t_2:



$$
\begin{aligned}
P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0006804
\end{aligned}
$$

Chomsky Normal Form (CNF)



Noam Chomsky

- Only two types of production rules in CNF

  $A \rightarrow B\ C$

  $A \rightarrow w$

- Any arbitrary CFG can be rewritten into CNF automatically
  - The resulting grammar accepts (and rejects) the same set of strings as the original grammar
  - But the resulting parse trees are different (i.e., binarized)