Sequence Models for Information Extraction, POS tagging, Word Segmentation, Chunking, …

CS224N

2007

(Some slides are mine; many slides are borrowed from Andrew McCallum and William Cohen’s IE Tutorial)

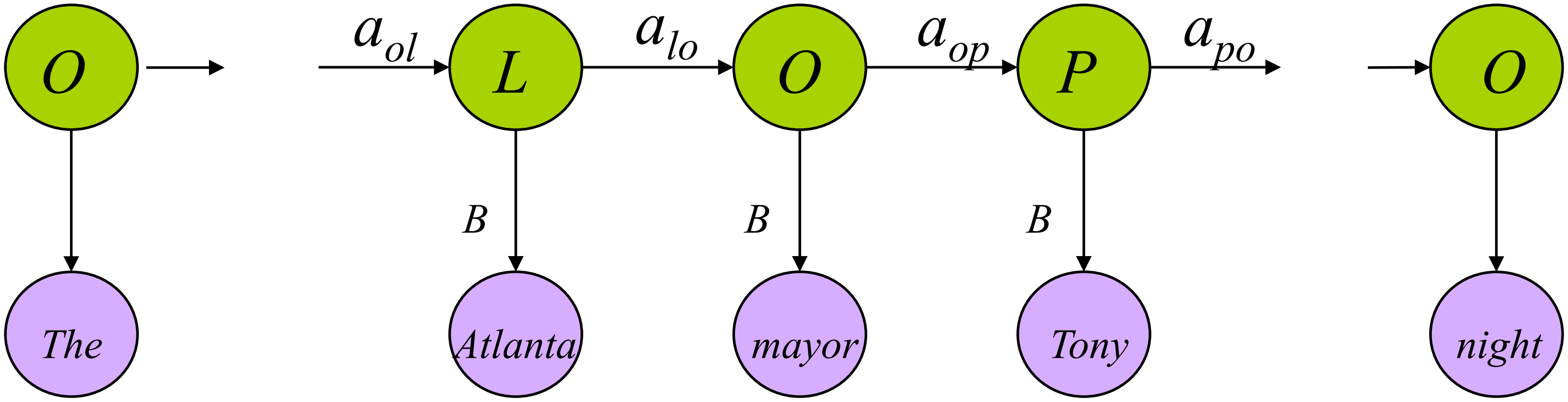
Statistical sequence models for Information Extraction

- There are several techniques for information extraction (template/wrapper learning, hand-coded rules) …
- But statistical sequence models (Hidden Markov Models, MaxEnt markov models, CRFs) are good methods for sequence-based information extraction
- Pros:
  - Well-understood underlying statistical model
  - Can do some form of optimal inference along sequence
  - Portable, broad coverage, robust, good recall
- Cons:
  - Not necessarily as good for complex or multi-slot patterns
  - Only doing the entity mention labeling task (in general)

Applying HMMs to IE

(Leek 1997, Freitag and McCallum 2000)

- **Multinomial HMMs** are sequential version of naïve Bayes/LM.
- **Document** ⇒ generated by a stochastic process
- **Observation** ⇒ word
- **State** ⇒ "reason/explanation" for a given token
  - '*Background'* state emits tokens like '*the*', '*said*', …
  - '*Money'* state emits tokens like '*million*', '*euro*', …
  - '*Person'* state emits tokens like '*Tony*', '*Prithi*', …
- **Extraction**: via the Viterbi (max likelihood parse) algorithm

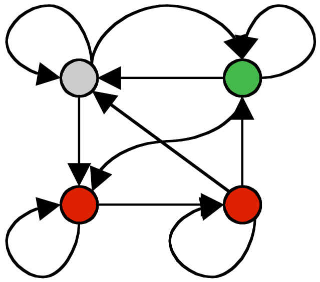


IE with Hidden Markov Models

Given a sequence of observations:

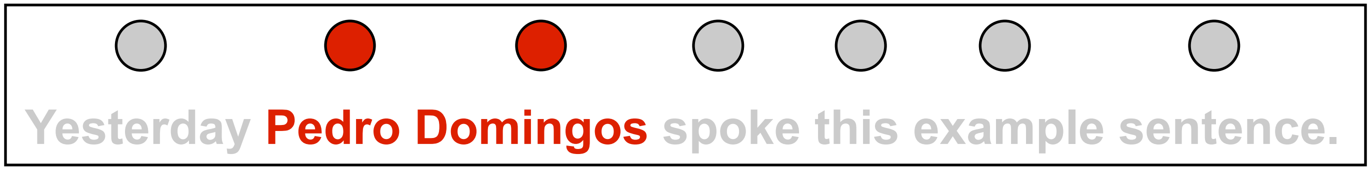
Yesterday Pedro Domingos spoke this example sentence.

and a trained HMM:



🟥 person name
🟩 location name
⬜ background

Find the most likely state sequence: (Viterbi)



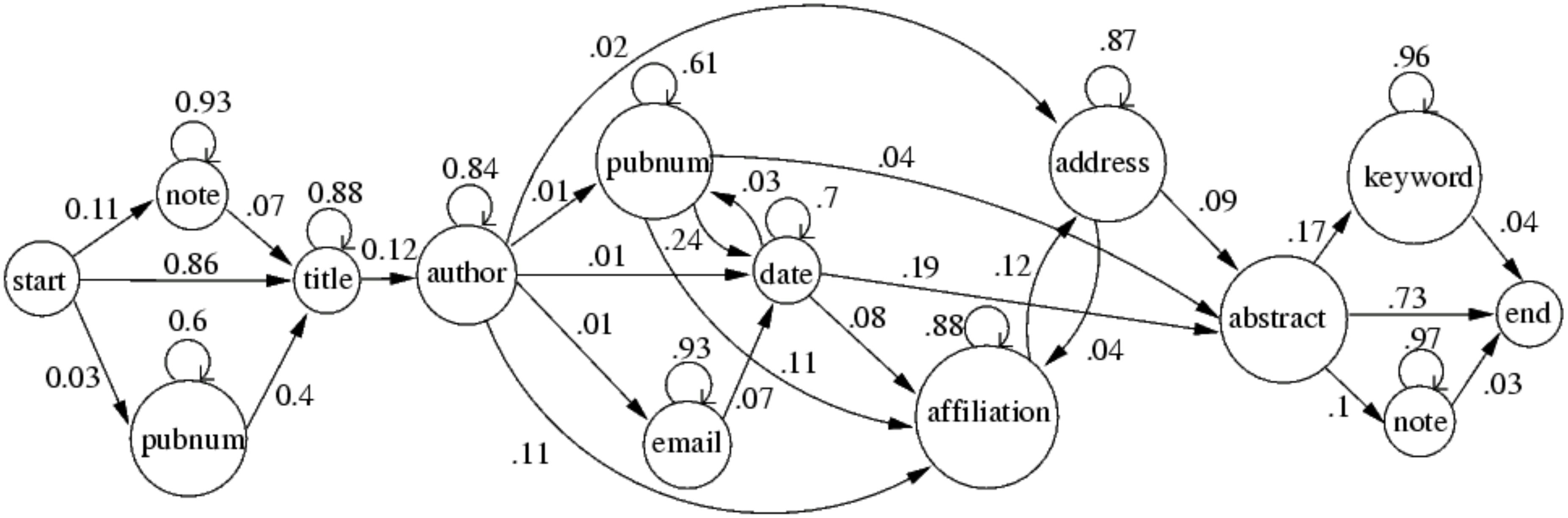
Yesterday **Pedro Domingos** spoke this example sentence.

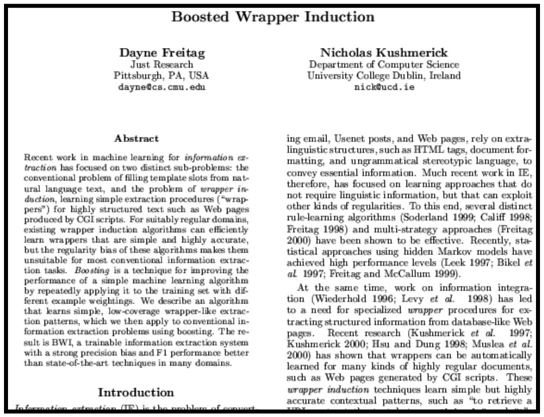
Any words said to be generated by the designated "person name" state extract as a person name:

Person name: Pedro Domingos

HMM for research papers: transitions A

[Seymore *et al.*, 99]





Boosted Wrapper Induction

Dayne Freitag
Just Research
Pittsburgh, PA, USA
dayne@cs.cmu.edu

Nicholas Kushmerick
Department of Computer Science
University College Dublin, Ireland
nick@ucd.ie

Abstract

Recent work in machine learning for *information extraction* has focused on two distinct sub-problems: the conventional problem of filling template slots from natural language text, and the problem of *wrapper induction*, learning simple extraction procedures (“wrappers”) for highly structured text such as Web pages produced by CGI scripts. For suitably regular domains, existing wrapper induction algorithms can efficiently learn wrappers that are simple and highly accurate, but the regularity bias of these algorithms makes them unsuitable for most conventional information extraction tasks. *Boosting* is a technique for improving the performance of a simple machine learning algorithm by repeatedly applying it to the training set with different example weightings. We describe an algorithm that learns simple, low-coverage wrapper-like extraction patterns, which we then apply to conventional information extraction problems using boosting. The result is BWI, a trainable information extraction system with a strong precision bias and F1 performance better than state-of-the-art techniques in many domains.

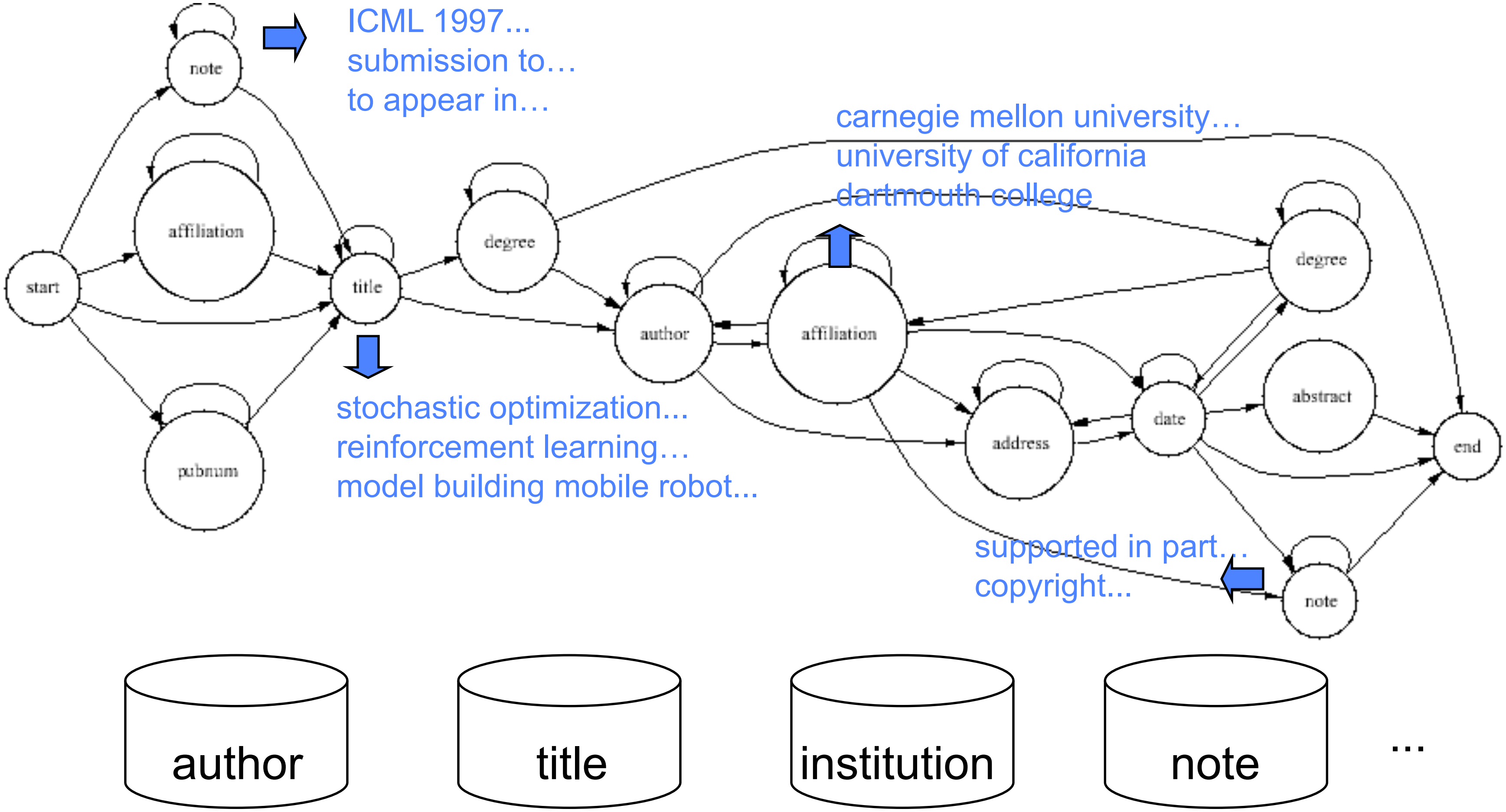
Introduction

Information extraction (IE) is the problem of converting email, Usenet posts, and Web pages, rely on extralinguistic structures, such as HTML tags, document formatting, and ungrammatical stereotypic language, to convey essential information. Much recent work in IE, therefore, has focused on learning approaches that do not require linguistic information, but that can exploit other kinds of regularities. To this end, several distinct rule-learning algorithms (Soderland 1999; Califf 1998; Freitag 1998) and multi-strategy approaches (Freitag 2000) have been shown to be effective. Recently, statistical approaches using hidden Markov models have achieved high performance levels (Leek 1997; Bikel *et al.* 1997; Freitag and McCallum 1999).

At the same time, work on information integration (Wiederhold 1996; Levy *et al.* 1998) has led to a need for specialized *wrapper* procedures for extracting structured information from database-like Web pages. Recent research (Kushmerick *et al.* 1997; Kushmerick 2000; Hsu and Dung 1998; Muslea *et al.* 2000) has shown that wrappers can be automatically learned for many kinds of highly regular documents, such as Web pages generated by CGI scripts. These *wrapper induction* techniques learn simple but highly accurate contextual patterns, such as “to retrieve a

HMM for research papers: emissions B

[Seymore *et al.*, 99]



ICML 1997...
submission to…
to appear in…

carnegie mellon university…
university of california
dartmouth college

stochastic optimization...
reinforcement learning…
model building mobile robot...

supported in part…
copyright...

author

title

institution

note

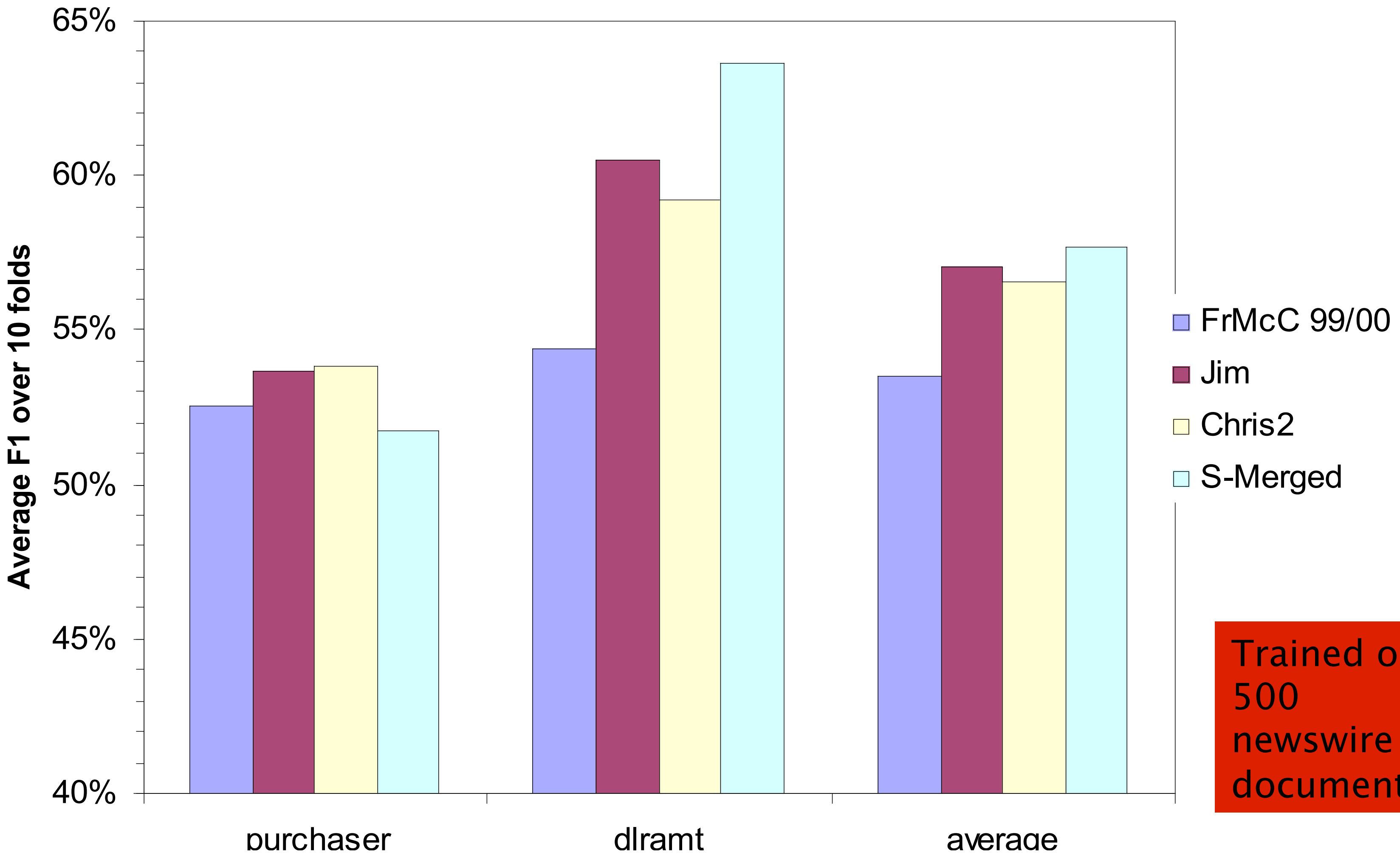
...

Trained on 2 million words of BibTeX data from the Web

Freitag and McCallum (2000) IE with HMMs details

- Partly fixed structure, partly hidden (constrained EM using remote supervision)
  - Class HMM (also used in comp. bio.)
- Parameter tying and shrinkage smoothing techniques
  - Better just to use a good unknown model?
- Structure learning of transition structure
  - Why not just plain EM?
- Results great on semi-structured data!
  - 92.9% token accuracy on paper/citations data
- Still rather modest on free form text

HMM IE results (F_1) on Freitag and McCallum Acquisitions data



Trained on 500 newswire documents

Other Sequence Modeling Tasks: Chinese Word Segmentation

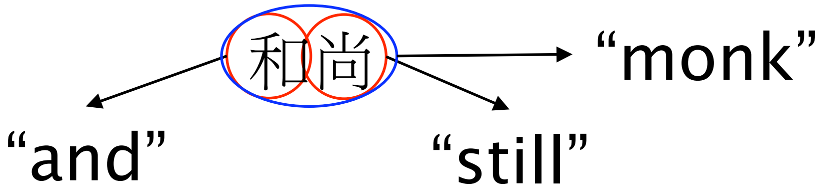
(also: Japanese, Thai, Ancient Greek, …)

- Basic units in written text are “characters”.

- A sentence is a sequence of “characters”, without explicit boundaries.

  已開發和尚在開發的資源

- Meaningful units in written texts are “words”
- Word meaning can differ greatly from characters



和尚 “monk”

“and”

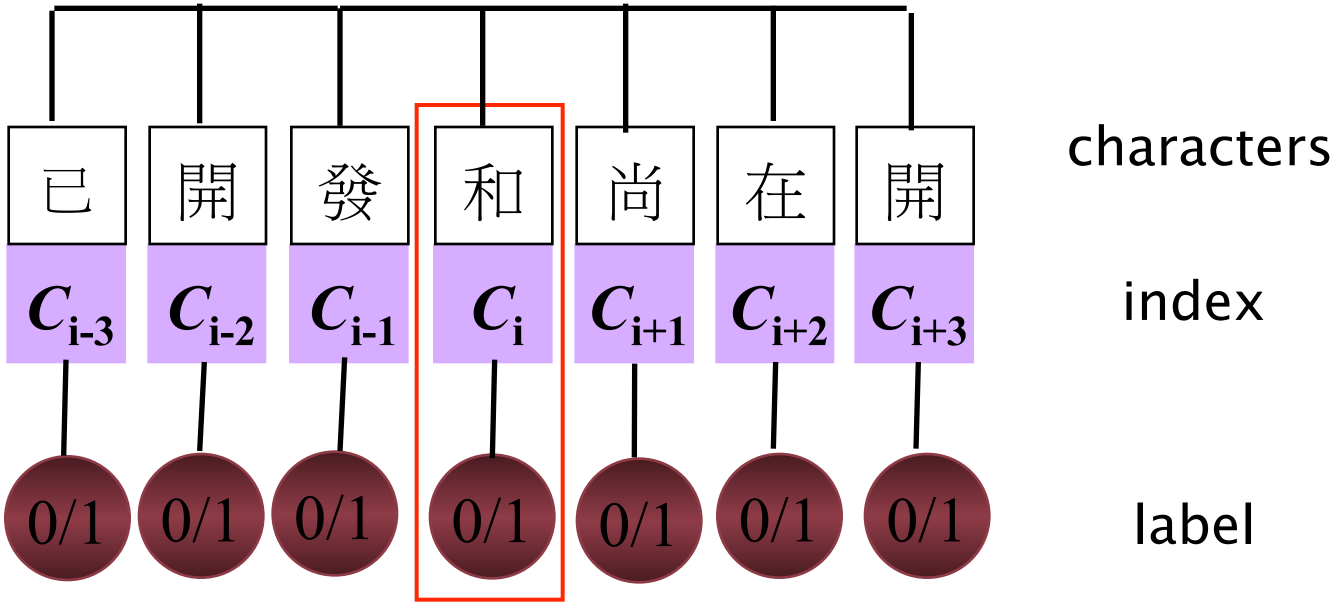
“still”

- But definition of “words” is debatable
  - Different segmentation standards defined by linguists
  - It’s like whether you segment compounds (cf. German)

Sequence Model Chinese Word Segmenter

0:NONSTART

1: START



Other sequence modeling tasks

- Base noun phrase chunking
  - Small noun phrases are a useful unit for many applications of terminology extraction, web search
  - Mitsubishi has just announced a new 21.3-inch flat panel monitor for the Japanese market, and even though it offers two DVI ports and a UXGA resolution of 1,600 x 1,200, we're not sure how many folks will be willing to part with close to 200,000 yen
  - Sequence model marks segment start, end

Other sequence modeling tasks

- Topic/FAQ segmentation (question, answer)
- Part-of-speech tagging
- Musical sequences
- DNA sequences
- …

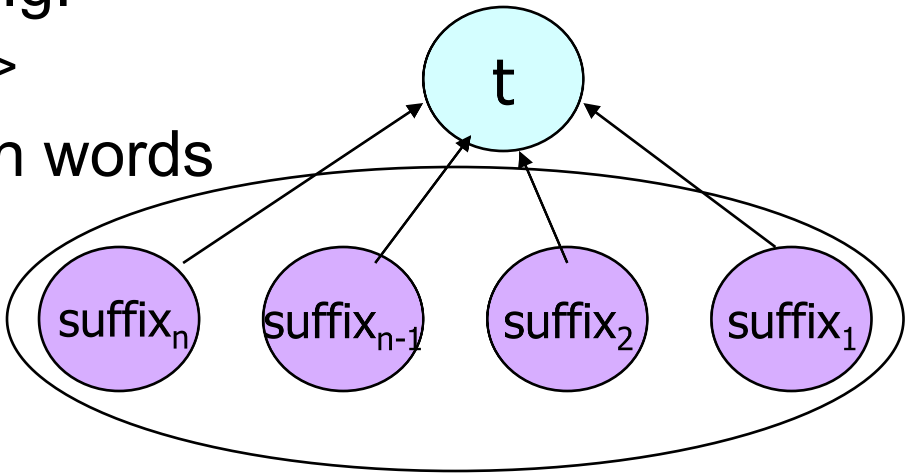
HMM Tagging Models - Brants 2000

- Highly competitive with other state-of-the art models
- Trigram HMM with smoothed transition probabilities
- Capitalization feature becomes part of the state – each tag state is split into two e.g.

  NN → <NN,cap>,<NN,not cap>
- Suffix features for unknown words

$$P(w \mid tag) = P(suffix \mid tag)(w \mid suffix)$$

$$\approx \hat{P}(suffix)\tilde{P}(tag \mid suffix) / \hat{P}(tag)$$



$$\tilde{P}(tag \mid suffix_n) = \lambda_1 \hat{P}(tag \mid suffix_n) + \lambda_2 \hat{P}(tag \mid suffix_{n-1}) + \ldots + \lambda_n \hat{P}(tag)$$

Named Entity Extraction

- The task: find and classify names in text, for example:

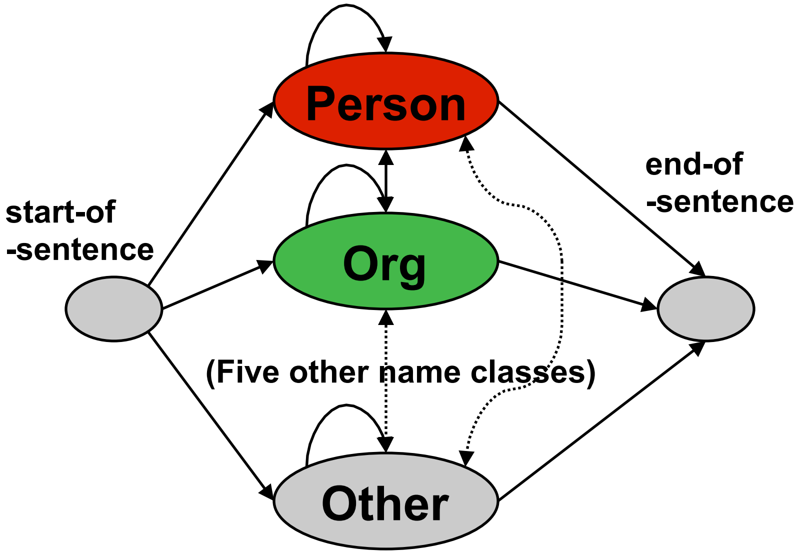
```
The European Commission [ORG] said on Thursday it disagreed
with German [MISC] advice.

Only France [LOC] and Britain [LOC] backed Fischler [PER]
's proposal .

“What we have to be extremely careful of is how other
countries are going to take Germany 's lead”, Welsh
National Farmers ' Union [ORG] ( NFU [ORG] ) chairman John
Lloyd Jones [PER] said on BBC [ORG] radio .
```

- The purpose:
  - … a lot of information is really associations between named entities.
  - … for question answering, answers are usually named entities.
  - … the same techniques apply to other slot-filling classifications.

HMM Example: "Nymble"

[Bikel, et al 1998], [BBN "IdentiFinder"]

Task: Named Entity Extraction



Transition probabilities

$$P(s_t \mid s_{t-1}, o_{t-1})$$

Observation probabilities

$$P(o_t \mid s_t, s_{t-1})$$

or $P(o_t \mid s_t, o_{t-1})$

Back-off to:

$$P(s_t \mid s_{t-1})$$

$$P(s_t)$$

Back-off to:

$$P(o_t \mid s_t)$$

$$P(o_t)$$

Train on ~500k words of news wire text.

Results:

Case	Language	F1 .
Mixed	English	93%
Upper	English	91%
Mixed	Spanish	90%

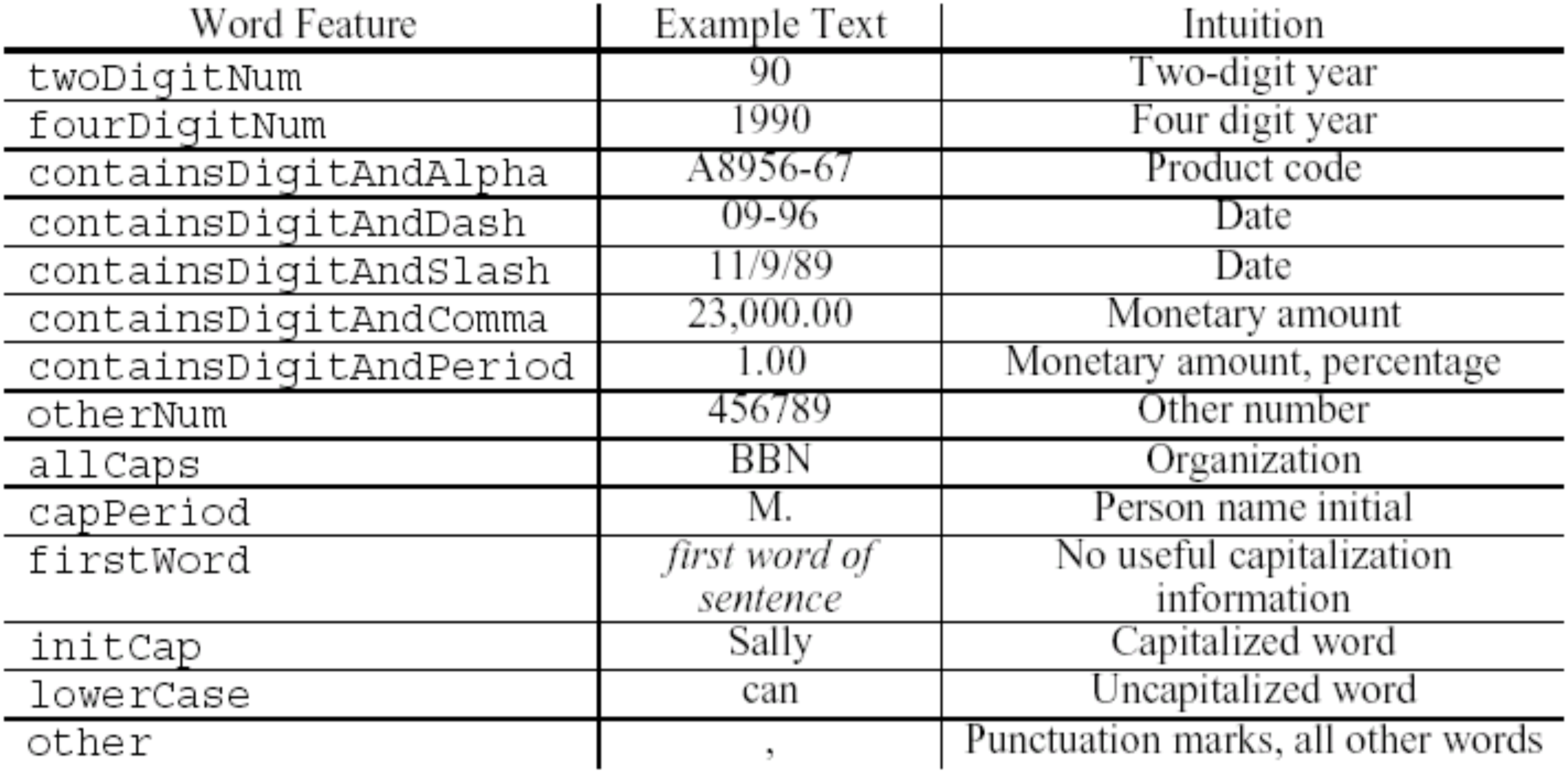
- Since 1997, probabilistic sequence approaches (BBN, NYU, then everyone) achieves state-of-the-art performance

Other examples of shrinkage for HMMs in IE: *[Freitag and McCallum '99]*

What is a symbol?

Bikel *et al* mix symbols from **two** abstraction levels

- A word token (for known words, seen more than *k* times

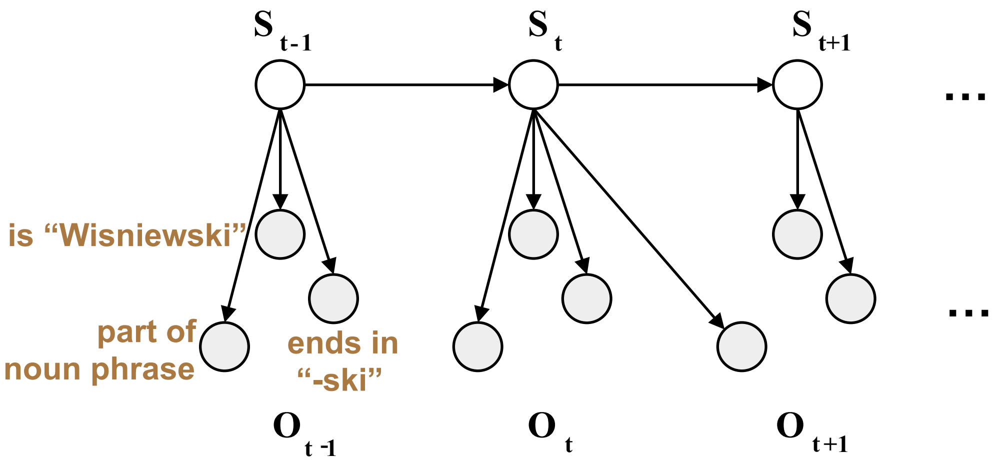


Word Feature	Example Text	Intuition
twoDigitNum	90	Two-digit year
fourDigitNum	1990	Four digit year
containsDigitAndAlpha	A8956-67	Product code
containsDigitAndDash	09-96	Date
containsDigitAndSlash	11/9/89	Date
containsDigitAndComma	23,000.00	Monetary amount
containsDigitAndPeriod	1.00	Monetary amount, percentage
otherNum	456789	Other number
allCaps	BBN	Organization
capPeriod	M.	Person name initial
firstWord	*first word of sentence*	No useful capitalization information
initCap	Sally	Capitalized word
lowerCase	can	Uncapitalized word
other	,	Punctuation marks, all other words

What is a symbol?

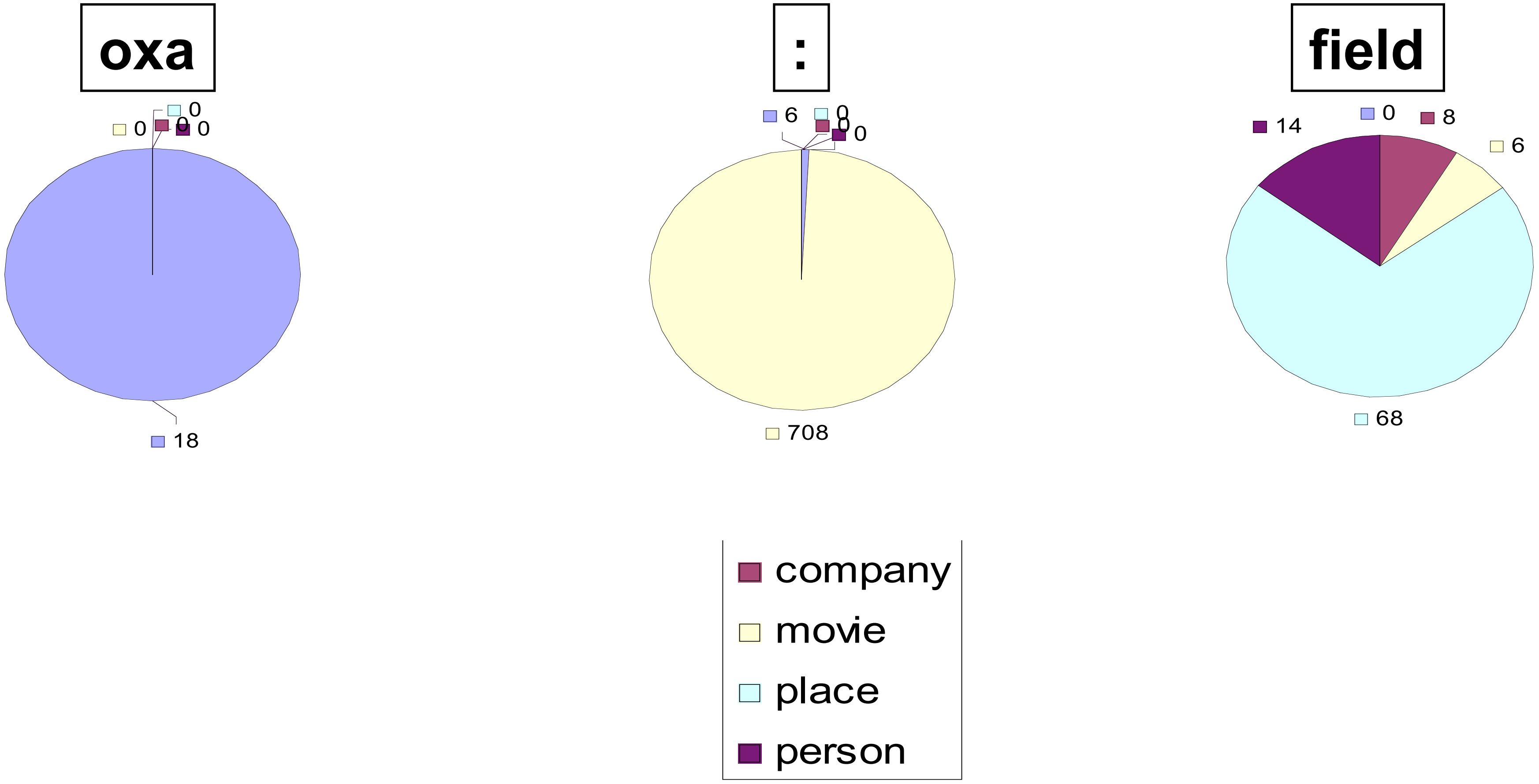
Ideally we would like to use many, arbitrary, overlapping features of words. Useful, but this is hard with HMMs

identity of word
ends in “-ski”
is capitalized
is part of a noun phrase
is in a list of city names
is under node X in WordNet
is in bold font
is indented
is in hyperlink anchor
…



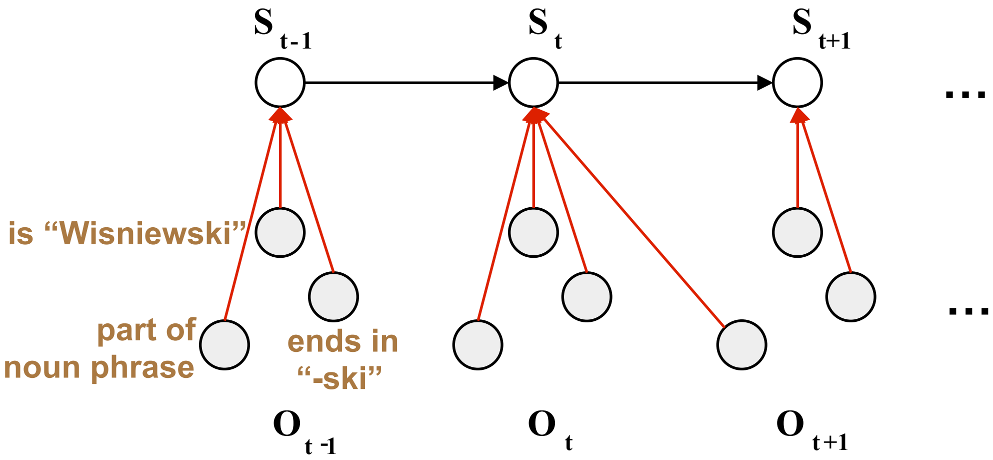
Lots of learning systems are **not** confounded by multiple, non-independent features: decision trees, maxent models, neural nets, SVMs, …

What’s in a Name?



What is a symbol?

identity of word
ends in "-ski"
is capitalized
is part of a noun phrase
is in a list of city names
is under node X in WordNet
is in bold font
is indented
is in hyperlink anchor
…

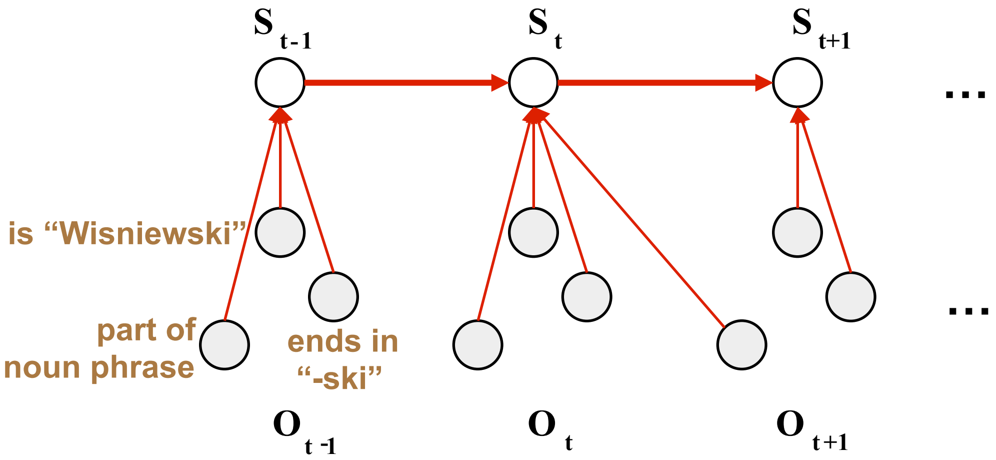


Idea: replace **generative** model in HMM with a **maxent** model, where **state** depends on **observations**

$$\Pr(s_t \mid x_t) = \ldots$$

What is a symbol?

identity of word
ends in “-ski”
is capitalized
is part of a noun phrase
is in a list of city names
is under node X in WordNet
is in bold font
is indented
is in hyperlink anchor
…

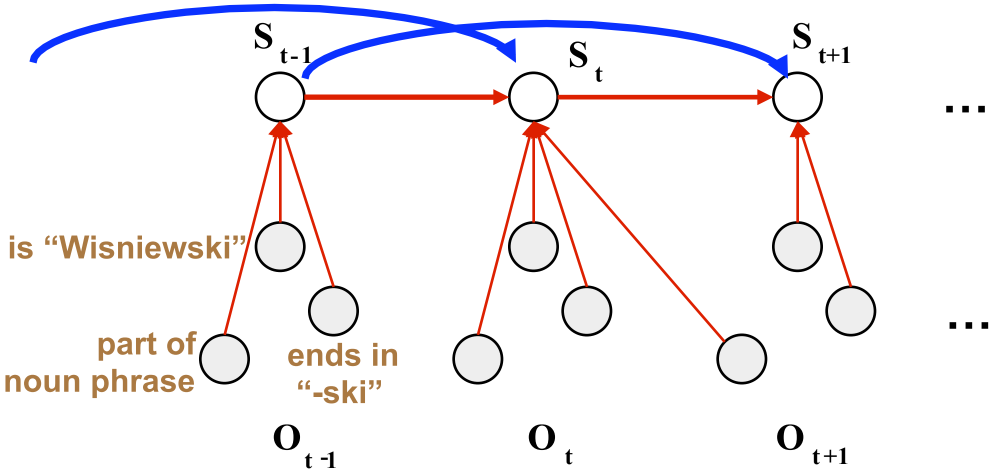


Idea: replace **generative** model in HMM with a **maxent** model, where **state** depends on **observations** and **previous state**

$$\Pr(s_t \mid x_t, s_{t-1,}) = \ldots$$

What is a symbol?

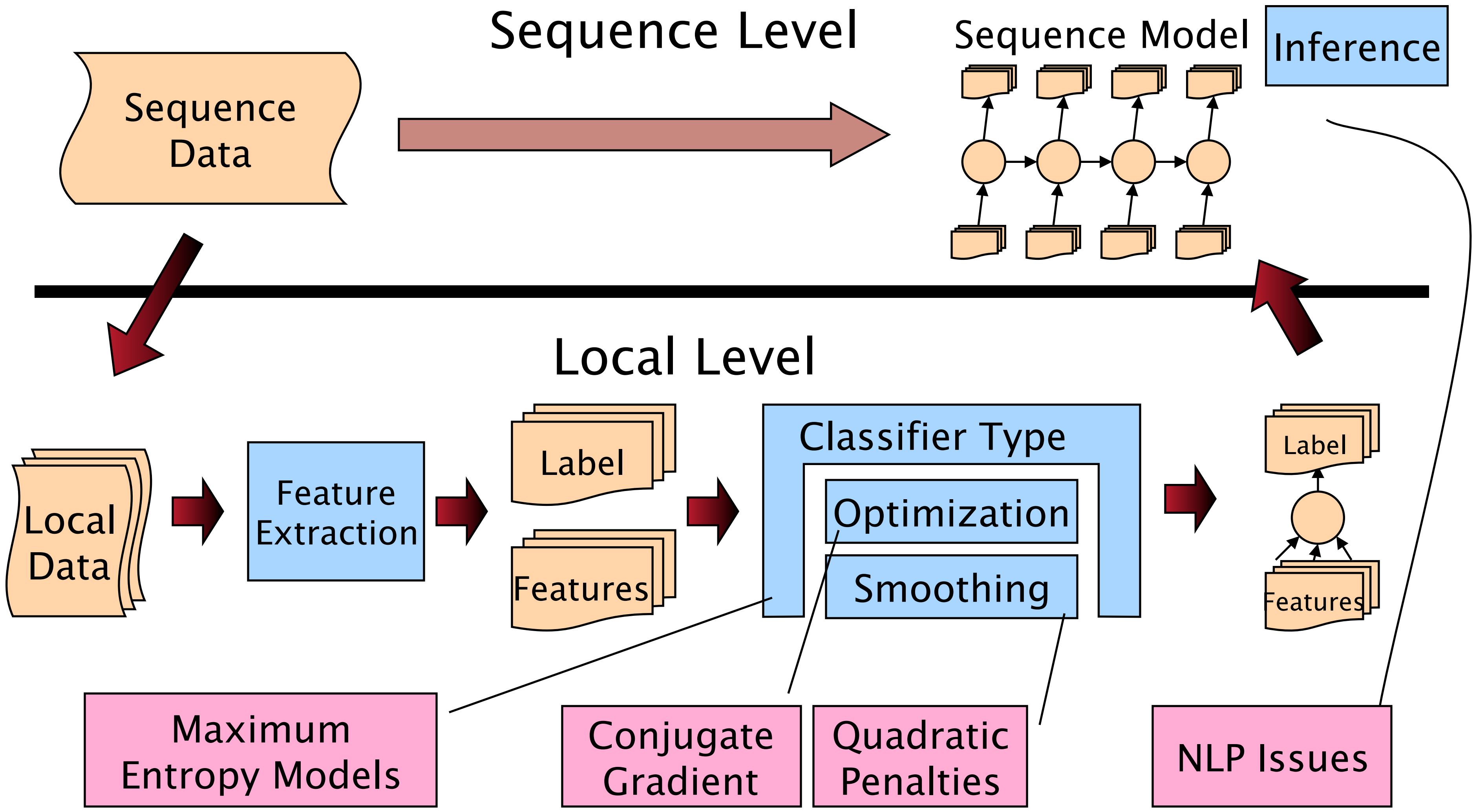
identity of word
ends in "-ski"
is capitalized
is part of a noun phrase
is in a list of city names
is under node X in WordNet
is in bold font
is indented
is in hyperlink anchor
…



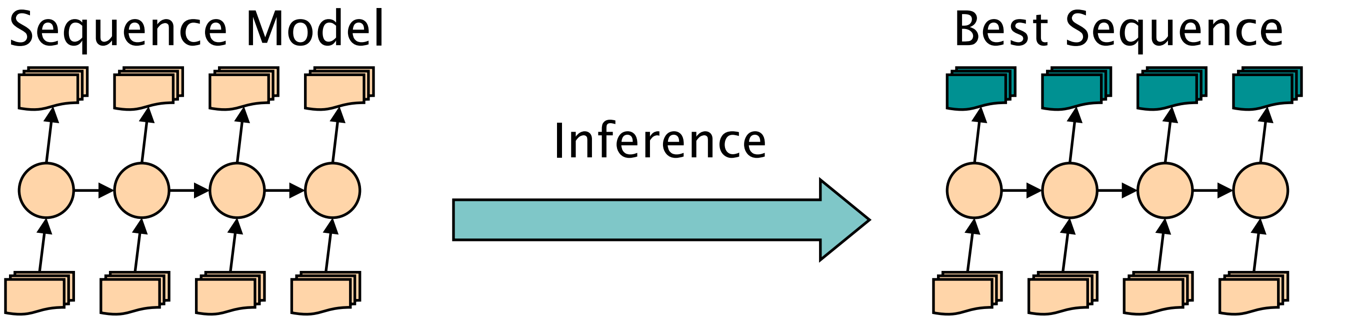
Idea: replace **generative** model in HMM with a **maxent** model, where **state** depends on **observations** and **previous state** history

$$\Pr(s_t \mid x_t, s_{t-1,} s_{t-2,} \ldots) = \ldots$$

Inference in Systems

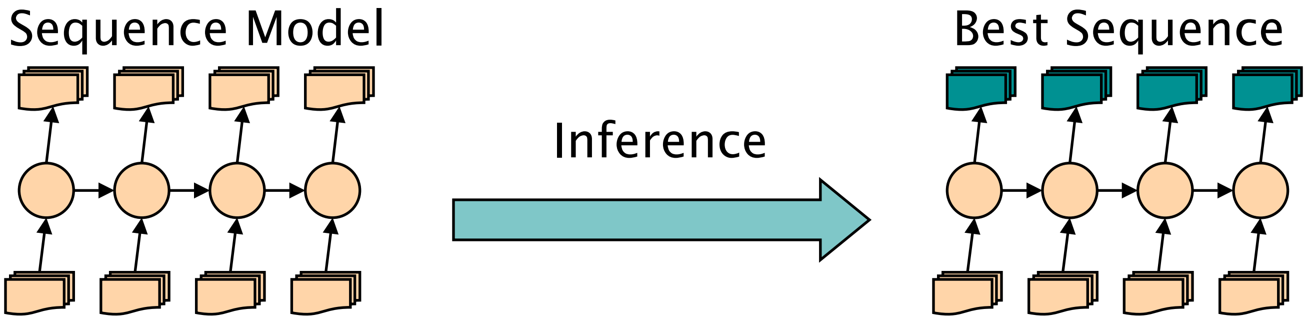


Beam Inference



- Beam inference:
  - At each position keep the top k complete sequences.
  - Extend each sequence in each local way.
  - The extensions compete for the k slots at the next position.
- Advantages:
  - Fast; and beam sizes of 3–5 are as good or almost as good as exact inference in many cases.
  - Easy to implement (no dynamic programming required).
- Disadvantage:
  - Inexact: the globally best sequence can fall off the beam.

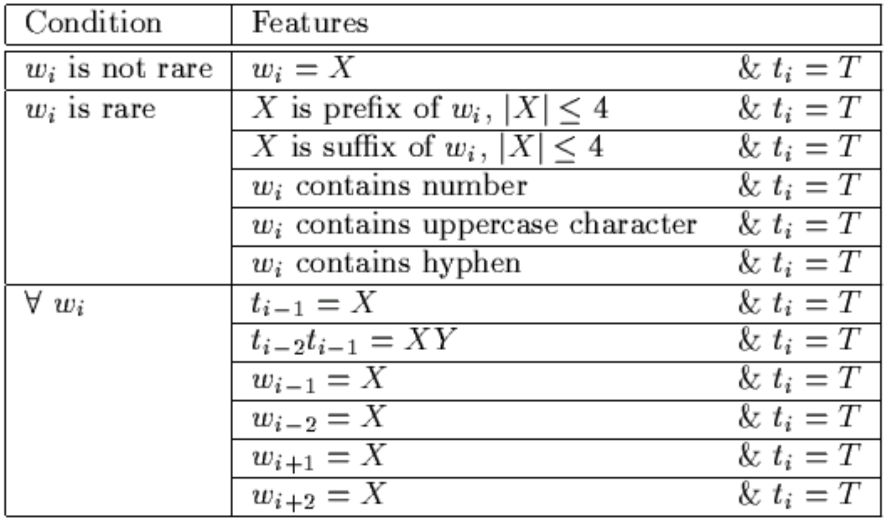
Viterbi Inference



- Viterbi inference:
  - Dynamic programming or memoization.
  - Requires small window of state influence (e.g., past two states are relevant).
- Advantage:
  - Exact: the global best sequence is returned.
- Disadvantage:
  - Harder to implement long-distance state-state interactions (but beam inference tends not to allow long-distance resurrection of sequences anyway).

POS tagging: Ratnaparkhi’s MXPOST

- Sequential learning problem: predict POS tags of words.
- Uses MaxEnt model described above.
- Rich feature set.
- To smooth, discard features occurring < 10 times.



Condition	Features	
w_i is not rare	$w_i = X$	$\& \; t_i = T$
w_i is rare	X is prefix of w_i, $\|X\| \leq 4$	$\& \; t_i = T$
	X is suffix of w_i, $\|X\| \leq 4$	$\& \; t_i = T$
	w_i contains number	$\& \; t_i = T$
	w_i contains uppercase character	$\& \; t_i = T$
	w_i contains hyphen	$\& \; t_i = T$
$\forall \; w_i$	$t_{i-1} = X$	$\& \; t_i = T$
	$t_{i-2}t_{i-1} = XY$	$\& \; t_i = T$
	$w_{i-1} = X$	$\& \; t_i = T$
	$w_{i-2} = X$	$\& \; t_i = T$
	$w_{i+1} = X$	$\& \; t_i = T$
	$w_{i+2} = X$	$\& \; t_i = T$

Table 1: Features on the current history h_i

CMM Tagging Models -II

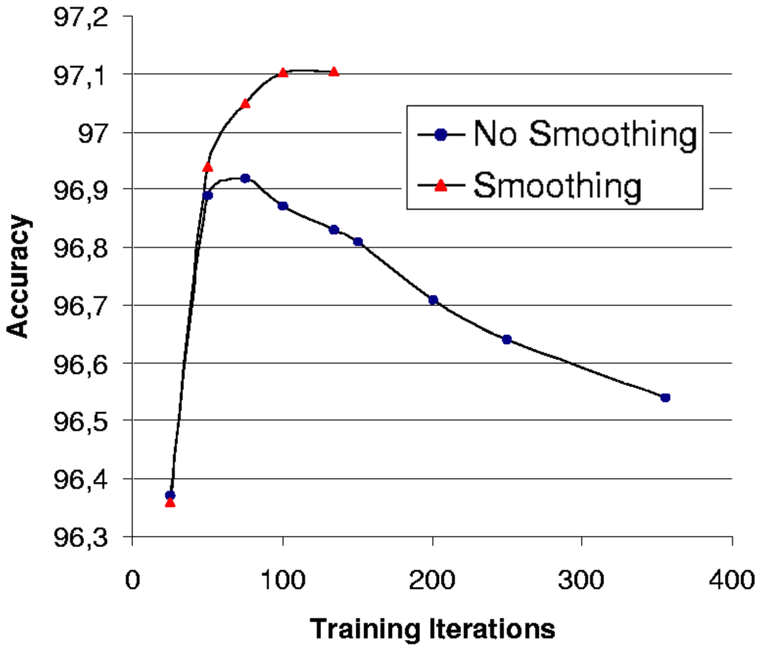
- Ratnaparkhi (1996): local distributions are estimated using maximum entropy models
  - Previous two tags, current word, previous two words, next two words, suffix, prefix, hyphenation, and capitalization features for unknown words
- Toutanova et al. (2003)
  - Richer features, bidirectional inference, better smoothing, better unknown word handling

Model	Overall Accuracy	Unknown Words
HMM (Brants 2000)	96.7	85.5
CMM (Ratn. 1996)	96.63	85.56
CMM (T. et al 2003)	97.24	89.04

Smoothing: POS Tagging

- From (Toutanova et al., 2003):

	Overall Accuracy	Unknown Word Acc
Without Smoothing	96.54	85.20
With Smoothing	97.10	88.20



- Smoothing helps:
  - Softens distributions.
  - Pushes weight onto more explanatory features.
  - Allows many features to be dumped (fairly) safely into the mix.
  - Speeds up convergence (if both are allowed to converge)!

Summary of POS Tagging

For tagging, the change from generative to discriminative model **does not by itself** result in great improvement

One profits from discriminative models for specifying dependence on **overlapping features of the observation** such as spelling, suffix analysis,etc

A CMM allows integration of rich features of the observations, but suffers strongly from assuming independence from following observations; this effect can be relieved by adding dependence on following words

This additional power (of the CMM ,CRF, Perceptron models) has been shown to result in improvements in accuracy

The **higher accuracy** of discriminative models comes at the price of **much slower training**

CoNLL (2003) Named Entity Recognition task

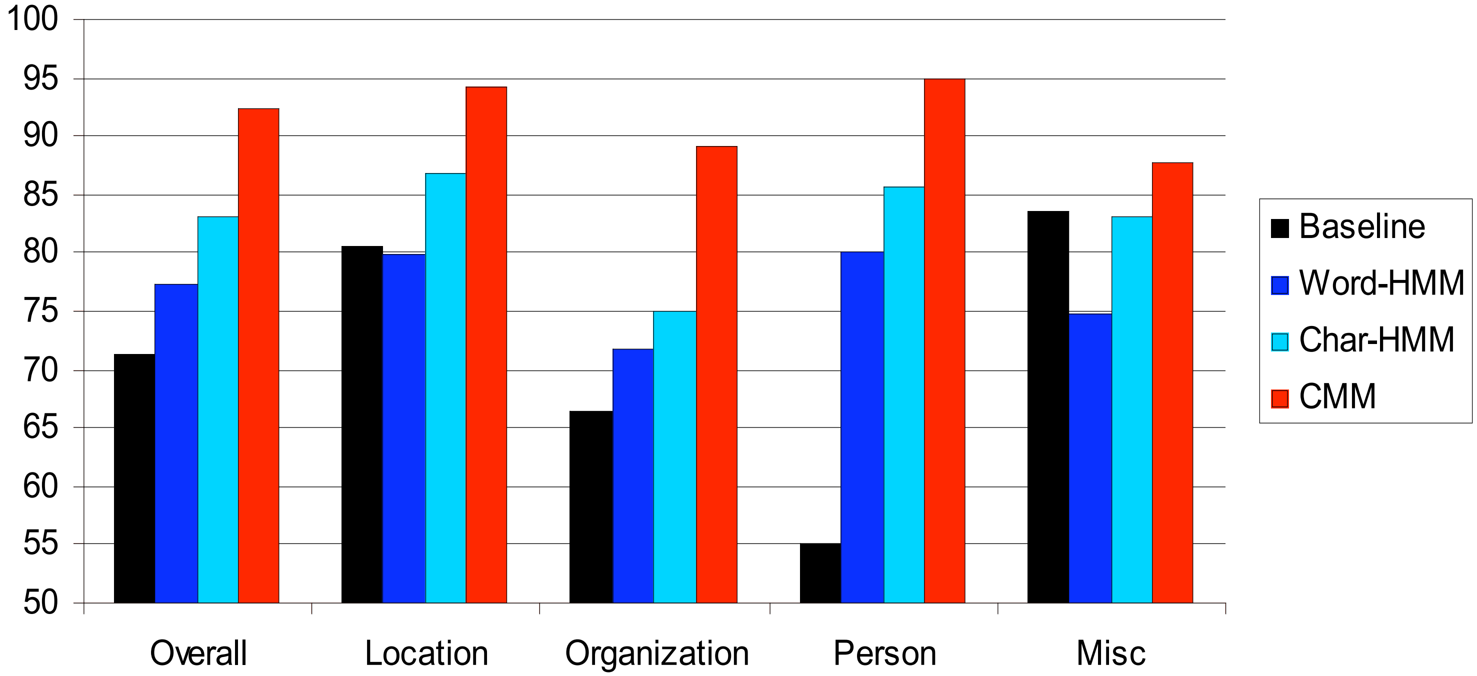
Task: Predict semantic label of each word in text

Foreign	NNP	I-NP	ORG
Ministry	NNP	I-NP	ORG
spokesman	NN	I-NP	O
Shen	NNP	I-NP	PER
Guofang	NNP	I-NP	PER
told	VBD	I-VP	O
Reuters	NNP	I-NP	ORG
:	:	:	:

} Standard evaluation is per entity, *not* per token

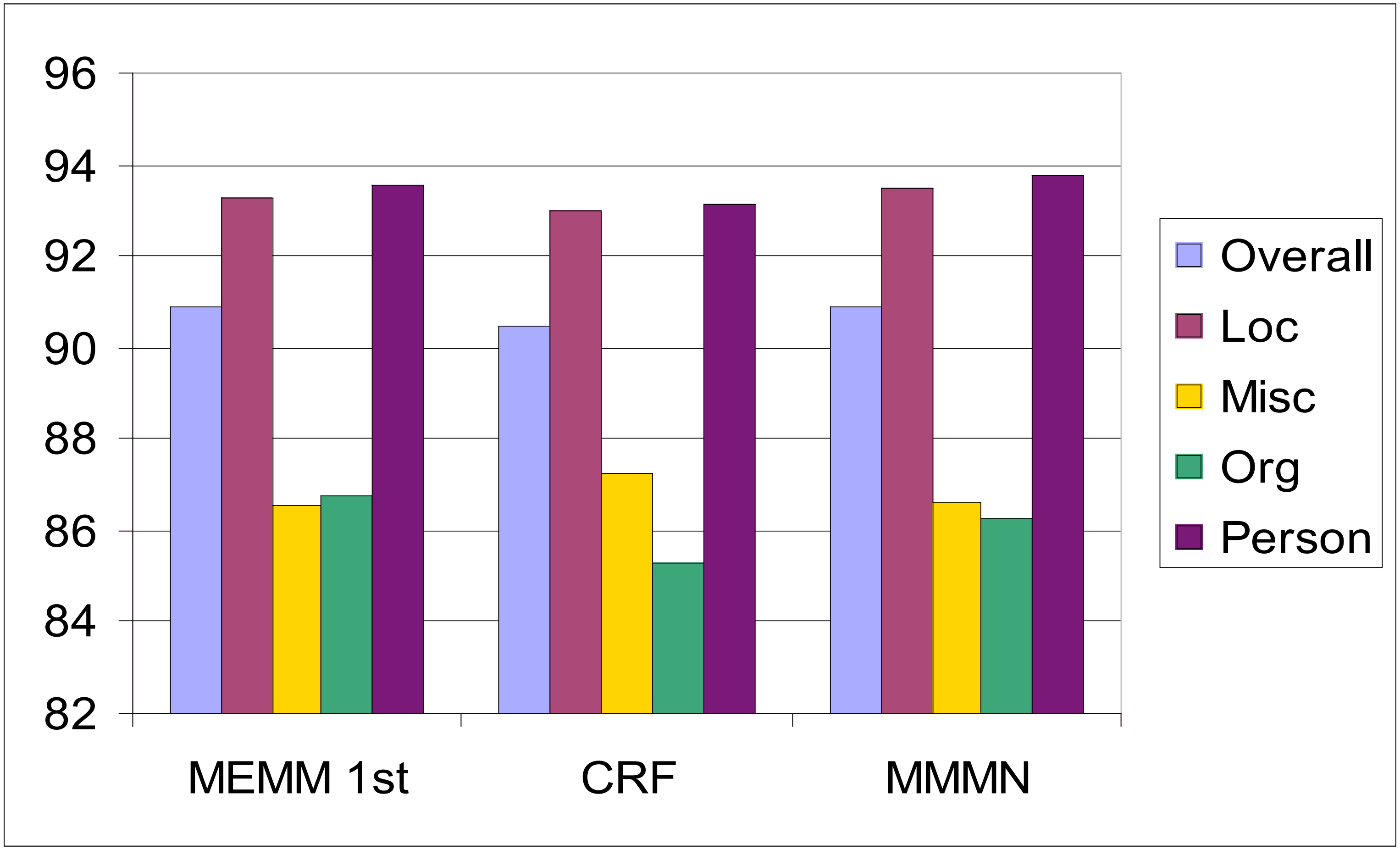
NER Results: Discriminative Model

- Increases from better features, a better classification model.

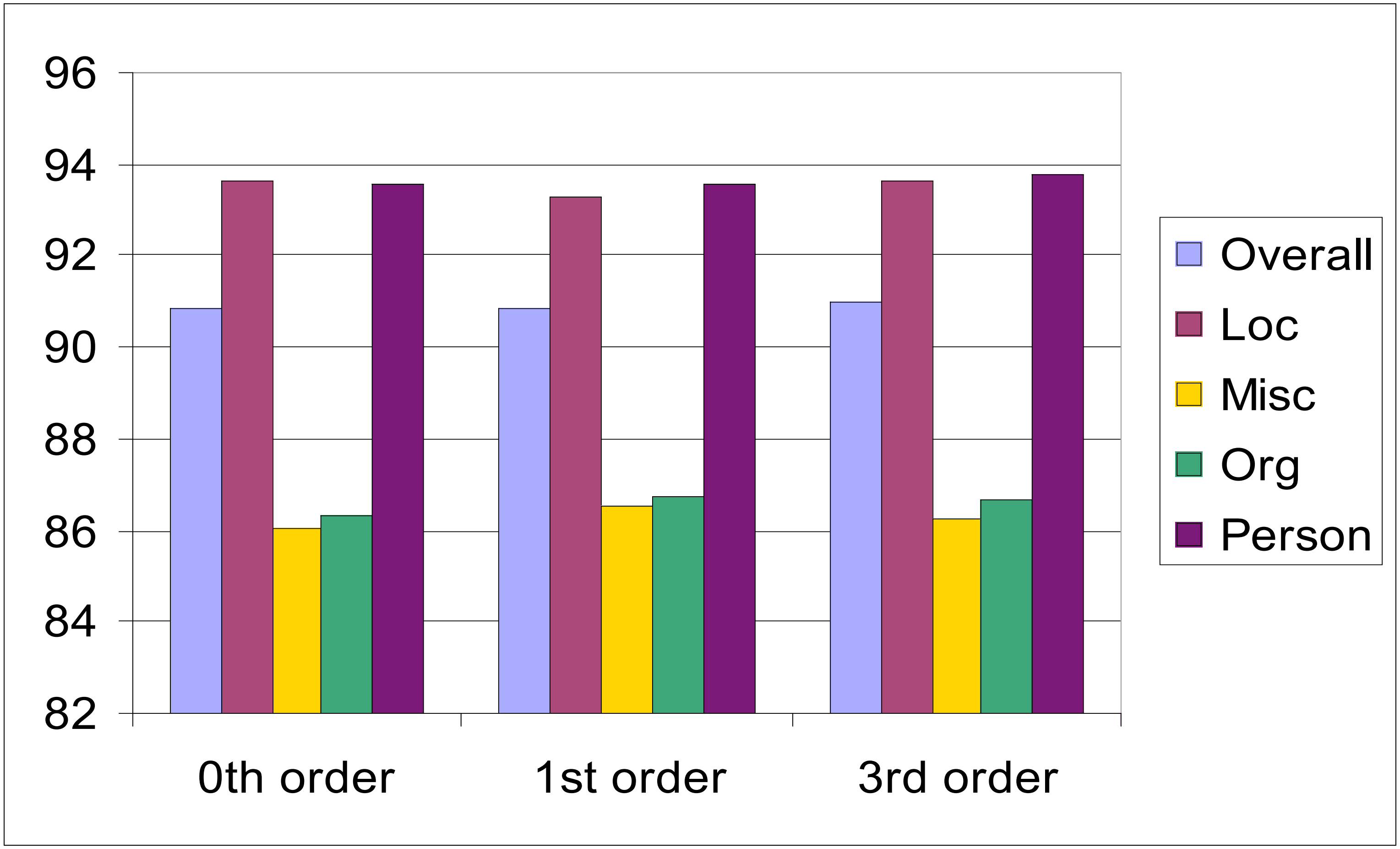


CoNLL 2003 Shared Task: English NER; entity precision/recall F1

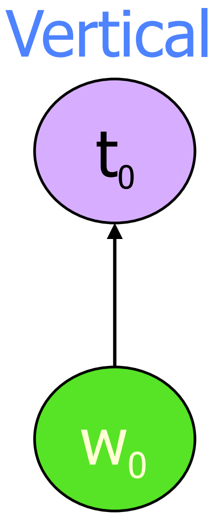
Sequence models? CoNLL 2003 NER shared task Results on English Devset

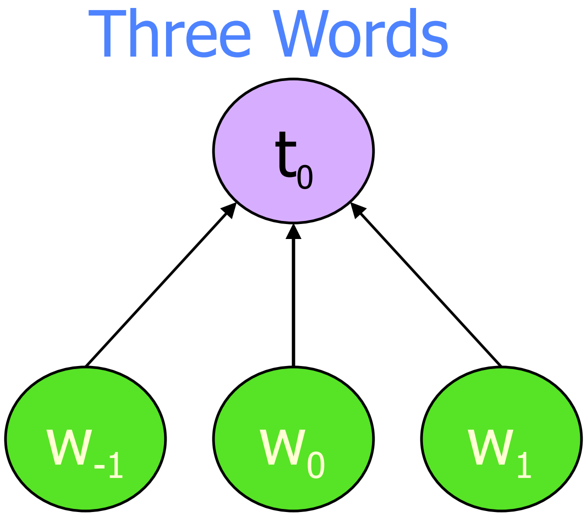


CoNLL NER Results: CMM Order



Sequence Tagging Without Sequence Information: POS tagging


Vertical


Three Words

Model	Features	Token	Unknown	Sentence
Vertical	56,805	**93.69%**	82.61%	26.74%
3Words	239,767	**96.57%**	86.78%	48.27%

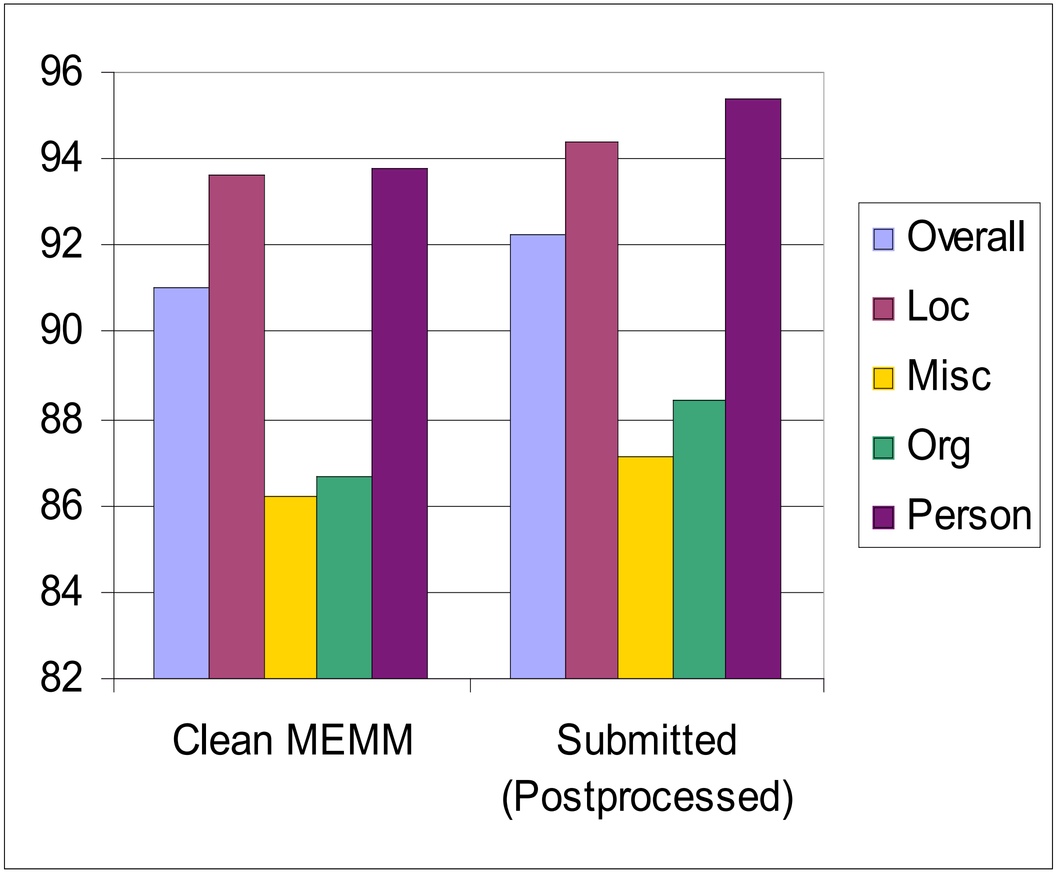
Using 3 words only works significantly better than using the previous two or three tags instead! (Toutanova et al. 2003)

CoNLL NER: A real difference

- A difference of about 0.7% gives significance among good CoNLL results

- Here we get one!

- It was done with some Perl regular expressions


Clean MEMM
Submitted (Postprocessed)
Overall
Loc
Misc
Org
Person