CS 294-5: Statistical Natural Language Processing

Dan Klein
MF 1-2:30pm
Soda Hall 310

Last Time

- Maximum entropy models
  - A technique for estimating multinomial distributions conditionally on many features

$$P(c \mid d, \lambda) = \frac{\exp \sum_i \lambda_i(c) f_i(d)}{\sum_{c'} \exp \sum_i \lambda_i(c') f_i(d)}$$

  - A building block of many NLP systems
  - Catch-up session on Wednesday!
    - (a) First part of my office hours (3-4)
    - (b) Right before my office hours (2-3)

Today

- Sequence modeling
  - Part-of-Speech Tagging
  - HMMs

Parts-of-Speech

- Syntactic classes of words
  - Useful distinctions vary from language to language
  - Tagsets vary from corpus to corpus [See M+S p. 142]
- Some tags from the Penn tagset

CD	numeral, cardinal	mid-1890 nine-thirty 0.5 one
DT	determiner	a all an every no that the
IN	preposition or conjunction, subordinating	among whether out on by if
JJ	adjective or numeral, ordinal	third ill-mannered regrettable
MD	modal auxiliary	can may might will would
NN	noun, common, singular or mass	cabbage thermostat investment subhumanity
NNP	noun, proper, singular	Motown Cougar Yvette Liverpool
PRP	pronoun, personal	hers himself it we them
RB	adverb	occasionally maddeningly adventurously
RP	particle	aboard away back by on open through
VB	verb, base form	ask bring fire see take
VBD	verb, past tense	pleaded swiped registered saw
VBN	verb, past participle	dilapidated imitated reunifed unsettled
VBP	verb, present tense, not 3rd person singular	twist appear comprise mold postpone

CC	conjunction, coordinating	and both but either or
CD	numeral, cardinal	mid-1890 nine-thirty 0.5 one
DT	determiner	a all an every no that the
EX	existential there	there
FW	foreign word	gemeinschaft hund ich jeux
IN	preposition or conjunction, subordinating	among whether out on by if
JJ	adjective or numeral, ordinal	third ill-mannered regrettable
JJR	adjective, comparative	braver cheaper taller
JJS	adjective, superlative	bravest cheapest tallest
MD	modal auxiliary	can may might will would
NN	noun, common, singular or mass	cabbage thermostat investment subhumanity
NNP	noun, proper, singular	Motown Cougar Yvette Liverpool
NNPS	noun, proper, plural	Americans Materials States
NNS	noun, common, plural	undergraduates bric-a-brac averages
POS	genitive marker	' 's
PRP	pronoun, personal	hers himself it we them
PRP$	pronoun, possessive	her his mine my our ours their thy your
RB	adverb	occasionally maddeningly adventurously
RBR	adverb, comparative	further gloomier heavier less-perfectly
RBS	adverb, superlative	best biggest nearest worst
RP	particle	aboard away back by on open through
TO	"to" as preposition or infinitive marker	to
UH	interjection	huh howdy uh whammo shucks heck
VB	verb, base form	ask bring fire see take
VBD	verb, past tense	pleaded swiped registered saw
VBG	verb, present participle or gerund	stirring focusing approaching erasing
VBN	verb, past participle	dilapidated imitated reunifed unsettled
VBP	verb, present tense, not 3rd person singular	twist appear comprise mold postpone
VBZ	verb, present tense, 3rd person singular	bases reconstructs marks uses
WDT	WH-determiner	that what whatever which whichever
WP	WH-pronoun	that what whatever which who whom
WP$	WH-pronoun, possessive	whose
WRB	Wh-adverb	however whenever where why

Part-of-Speech Ambiguity

- Example

VBD		VB			
VBN	VBZ	VBP	VBZ		
NNP	NNS	NN	NNS	CD	NN
Fed	raises	interest	rates	0.5	percent

- Two basic sources of constraint:
  - Grammatical environment
  - Identity of the current word
- Many more possible features:
  - … but we won't be able to use them until next class

Why POS Tagging?

- Useful in and of itself
  - Text-to-speech: record, lead
  - Lemmatization: saw[v] → see, saw[n] → saw
  - Quick-and-dirty NP-chunk detection: grep {JJ | NN}* {NN | NNS}
- Useful as a pre-processing step for parsing
  - Less tag ambiguity means fewer parses
  - However, some tag choices are better decided by parsers!

						IN			
DT	NNP	NN	VBD	VBN		RP	NN	NNS	
The	Georgia	branch	had	taken		on	loan	commitments	…

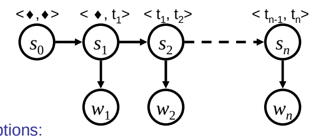
				VDN			
DT	NN	IN	NN	VBD	NNS	VBD	
The	average	of	interbank	offered	rates	plummeted	…

HMMs

- We want a generative model over sequences t and observations w using states s

$$P(T,W)=\prod_i P(t_i \mid t_{i-1},t_{i-2})P(w_i \mid t_i)$$

$$P(T,W)=\prod_i P(s_i \mid s_{i-1})P(w_i \mid s_i)$$



- Assumptions:
  - Tag sequence is generated by an order n markov model
  - This corresponds to a 1st order model over tag n-grams
  - Words are chosen independently, conditioned only on the tag
  - These are totally broken assumptions: why?

Parameter Estimation

- Need two multinomials
  - Transitions: $P(t_i \mid t_{i-1},t_{i-2})$
  - Emissions: $P(w_i \mid t_i)$
- Can get these off a collection of tagged sentences:
  - [examples]

Practical Issues with Estimation

- Use standard smoothing methods to estimate transition scores, e.g.:

$$P(t_i \mid t_{i-1},t_{i-2}) = \lambda_2 \hat{P}(t_i \mid t_{i-1},t_{i-2}) + \lambda_1 \hat{P}(t_i \mid t_{i-1})$$

- Emissions are tricker
  - Words we've never seen before
  - Words which occur with tags we've never seen
  - One option: break out the Good-Turning smoothing
  - Issue: words aren't black boxes:

    343,127.23 11-year Minteria reintroducible

  - Another option: decompose words into features and use a maxent model along with Bayes' rule.

$$P(w \mid t) = P_{MAXENT}(t \mid w)P(w)/P(t)$$

Disambiguation

- Given these two multinomials, we can score any word / tag sequence pair

<♦,♦>	<♦,NNP>	<NNP, VBZ>	<VBZ, NN>	<NN, NNS>	<NNS, CD>	<CD, NN>	<STOP>
	NNP	VBZ	NN	NNS	CD	NN	.
	Fed	raises	interest	rates	0.5	percent	.

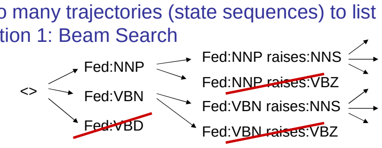
P(NNP|<♦,♦>) P(Fed|NNP) P(VBZ|<NNP,♦>) P(raises|VBZ) P(NN|VBZ,NNP)…..

- In principle, we're done – list all possible tag sequences, score each one, pick the best one (the Viterbi state sequence)

NNP VBZ NN NNS CD NN	⇨	logP = -23
NNP NNS NN NNS CD NN	⇨	logP = -29
NNP VBZ VB NNS CD NN	⇨	logP = -27

Finding the Best Trajectory

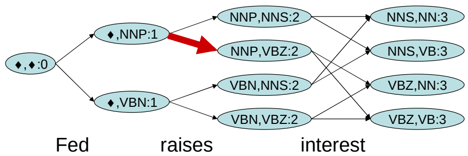
- Too many trajectories (state sequences) to list
- Option 1: Beam Search



  - A beam is a set of partial hypotheses
  - Start with just the single empty trajectory
  - At each derivation step:
    - Consider all continuations of previous hypotheses
    - Discard most, keep top k, or those within a factor of the best, (or some combination)
- Beam search works relatively well in practice
  - … but sometimes you want the optimal answer
  - … and you need optimal answers to validate your beam search

The Path Trellis

- Represent paths as a trellis over states



- Each arc (s_1:i → s_2:i+1) is weighted with the combined cost of:
  - Transitioning from s_1 to s_2 (which involves some unique tag t)
  - Emitting word i given t

P(VBZ | NNP, ♦) P(raises | VBZ)

- Each state path (trajectory):
  - Corresponds to a derivation of the word and tag sequence pair
  - Corresponds to a unique sequence of part-of-speech tags
  - Has a probability given by multiplying the arc weights in the path

The Viterbi Algorithm

- Dynamic program for computing

$$\delta_i(s) = \max_{s_0 \ldots s_{i-1} s} P(s_0 \ldots s_{i-1} s, w_1 \ldots w_i)$$

  - The score of a best path up to position i ending in state s

$$\delta_0(s) = \begin{cases} 1 & if\ s = <\bullet,\bullet> \\ 0 & otherwise \end{cases}$$

$$\delta_i(s) = \max_{s'} P(s \mid s')P(w \mid s)\delta_{i-1}(s')$$

  - Also store a backtrace

$$\psi_i(s) = \arg\max_{s'} P(s \mid s')P(w \mid s)\delta_{i-1}(s')$$

- Memoized solution
- Iterative solution

So How Does It Work?

- Choose the most common tag
  - 90.3% with a bad unknown word model
  - 93.7% with a good one!
- TnT (Brants, 2000):
  - A carefully smoothed trigram tagger
  - 96.7% on WSJ text (SOA is ~97.2%)
- Noise in the data
  - Many errors in the training and test corpora

DT NN IN NN VBD NNS VBD
The average of interbank offered rates plummeted …

  - Probably about 2% guaranteed error from noise (on this data)

JJ JJ NN
chief executive officer

NN JJ NN
chief executive officer

JJ NN NN
chief executive officer

NN NN NN
chief executive officer

What's Next for POS Tagging

- Better features!

RB
PRP VBD IN RB IN PRP VBD .
They left as soon as he arrived .

  - We could fix this with a feature that looked at the next word

JJ
NNP NNS VBD VBN .
Intrinsic flaws remained undetected .

  - We could fix this by linking capitalized words to their lowercase versions
- Solution: maximum entropy sequence models (next class)
- Reality check:
  - Taggers are already pretty good on WSJ journal text…
  - What the world needs is taggers that work on other text!

HMMs as Language Models

- We have a generative model of tagged sentences:

$$P(T,W) = \prod_i P(t_i \mid t_{i-1}, t_{i-2}) P(w_i \mid t_i)$$

- We can turn this into a distribution over sentences by summing over the tag sequences:

$$P(W) = \sum_T \prod_i P(t_i \mid t_{i-1}, t_{i-2}) P(w_i \mid t_i)$$

- Problem: too many sequences!
- (And beam search isn't going to help this time)

Summing over Paths

- Just like Viterbi, but with sum instead of max

$$\delta_i(s) = \max_{s_0 \ldots s_{i-1} s} P(s_0 \ldots s_{i-1} s, w_1 \ldots w_i)$$

$$\alpha_i(s) = \sum_{s_0 \ldots s_{i-1} s} P(s_0 \ldots s_{i-1} s, w_1 \ldots w_i)$$

- Recursive decomposition

$$\alpha_0(s) = \begin{cases} 1 & if\ s = <\bullet,\bullet> \\ 0 & otherwise \end{cases}$$

$$\alpha_i(s) = \sum_{s'} P(s \mid s')P(w \mid s)\alpha_{i-1}(s')$$

The Forward-Backward Algorithm

$$\alpha_i(s) = \sum_{s_0 \ldots s_{i-1} s} P(s_0 \ldots s_{i-1} s, w_1 \ldots w_i)$$

$$\beta_i(s) = \sum_{s_{i+1} \ldots s_n} P(s_{i+1} \ldots s_n, w_{i+1} \ldots w_n \mid s)$$

What Does This Buy Us?

- Why do we want forward and backward probabilities?
  - Lets us ask more questions
  - Like: what fraction of sequences contain tag t at position i

$$\gamma_i(s,s') = \alpha_{i-1}(s) P(s' \mid s) P(w_i \mid s') \beta_i(s')$$

$$P(t_i = t \mid w_1 \ldots w_n) = \frac{\sum_{s \to s' : tag(s') = t_i} \gamma_i(s,s')}{\sum_{s \to s'} \gamma_i(s,s')}$$

- Max-tag decoding:
  - Pick the tag at each point which has highest expectation
  - Raises accuracy a tiny bit
  - Bad idea in practice (why?)
- Also: Unsupervised learning of HMMs
  - At least in theory, more later…

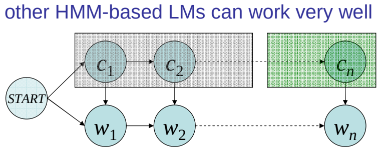
How's the HMM as a LM?

- POS tagging HMMs are terrible as LMs!

I bought an ice cream ___

The computer that I set up yesterday just ___

  - Don't capture long-distance effects like a parser could
  - Don't capture local collocational effects like n-grams
- But other HMM-based LMs can work very well



Next Time

- Better Tagging Features using Maxent
  - Dealing with unknown words
  - Adjacent words
  - Longer-distance features
- Named-Entity Recognition
- Reading: M+S 9-10, J+M 7.1-7.4