Natural Language Processing (CSE 447/547M): Sequence Models, Continued

Noah Smith

University of Washington
nasmith@cs.washington.edu

February 11, 2019

Recap

- Version 0: $\hat{y}_i = \operatorname*{argmax}_{y \in \mathcal{L}} s(\boldsymbol{x}, i, y)$ ("simple sequence labeler")
- Version 2: $\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} = \langle y_1, \ldots, y_\ell \rangle \in \mathcal{L}^\ell} \sum_{i=0}^{\ell} s(\boldsymbol{x}, i, y_i, y_{i+1})$
  - HMM (version 1) is the special case where $s(\boldsymbol{x}, 0, \bigcirc, y_1) = \log \pi_{y_1}$ and for $i \geq 1$, $s(\boldsymbol{x}, i, y_i, y_{i+1}) = \log \theta_{x_i | y_i} + \log \gamma_{y_{i+1} | y_i}$

Part-of-Speech Tagging Example

	I	suspect	the	present	forecast	is	pessimistic	.
noun	•	•	•	•	•	•		
adj.		•		•	•		•	
adv.				•				
verb		•		•	•	•		
num.	•							
det.			•					
punc.								•

With this very simple tag set, $7^8 = 5.7$ million labelings.
(Even restricting to the possibilities above, 288 labelings.)

Two Obvious Solutions

Brute force: Enumerate all solutions, score them, pick the best.

Greedy: For each $i \in \{1, \ldots, \ell\}$, pick $\hat{y}_i$ according to:

$$\hat{y}_i = \operatorname*{argmax}_{y \in \mathcal{L}} s(\boldsymbol{x}, i-1, \hat{y}_{i-1}, y)$$

$$\overset{\text{HMM case}}{=} \operatorname*{argmax}_{y \in \mathcal{L}} \underbrace{\theta_{x_i|y} \cdot \gamma_{y|\hat{y}_{i-1}}}_{p(x_i|y) \cdot p(y|\hat{y}_{i-1})}$$

What's wrong with these?

Conditional Independence

We can get an exact solution in polynomial time!

$$Y_i \perp \boldsymbol{Y}_{1:i-2} \mid Y_{i-1}$$
$$Y_i \perp \boldsymbol{Y}_{i+2:\ell} \mid Y_{i+1}$$

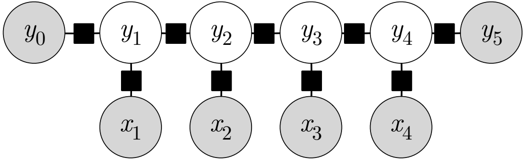
Given the adjacent labels to Y_i, others do not matter.

Let's start at the last position, ℓ ...

Order the labels in $\mathcal{L}$ as $\langle y, y', \ldots, y^{\diamondsuit} \rangle$

The End of the Sequence

	x_1	x_2	$\ldots$	x_ℓ
y				
y'				
$\vdots$				
$y^{\diamond}$				



$$\hat{y}_\ell = \operatorname*{argmax}_{y\in\mathcal{L}} s(\boldsymbol{x}, \ell - 1, {\color{blue}y_{\ell-1}}, y) + s(\boldsymbol{x}, \ell, y, {\color{red}\octagon})$$

The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!

High-Level View of the Viterbi Algorithm

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!

High-Level View of the Viterbi Algorithm

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!
- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ (and $y_{\ell-1}$) would be easy.

High-Level View of the Viterbi Algorithm

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!
- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ (and $y_{\ell-1}$) would be easy.
- Idea: for each position i, calculate the score of the best label prefix $\boldsymbol{y}_{1:i}$ ending in each possible value for Y_i.

High-Level View of the Viterbi Algorithm

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!
- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ (and $y_{\ell-1}$) would be easy.
- Idea: for each position i, calculate the score of the best label prefix $\boldsymbol{y}_{1:i}$ ending in each possible value for Y_i.
- With a little bookkeeping, we can then trace backwards and recover the best label sequence.

Recurrence

First, think about the *score* of the best sequence.

Let $\heartsuit_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$\heartsuit_\ell(y) = s(\boldsymbol{x}, \ell, y, \octagon) + \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \boxed{\heartsuit_{\ell-1}(y')}$$

Recurrence

First, think about the *score* of the best sequence.

Let $\heartsuit_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$\heartsuit_\ell(y) = s(\boldsymbol{x}, \ell, y, \bigcirc) + \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \boxed{\heartsuit_{\ell-1}(y')}$$

$$\heartsuit_{\ell-1}(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 2, y', y) + \boxed{\heartsuit_{\ell-2}(y')}$$

Recurrence

First, think about the *score* of the best sequence.

Let $\heartsuit_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$\heartsuit_\ell(y) = s(\boldsymbol{x}, \ell, y, \octagon) + \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \boxed{\heartsuit_{\ell-1}(y')}$$

$$\heartsuit_{\ell-1}(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 2, y', y) + \boxed{\heartsuit_{\ell-2}(y')}$$

$$\heartsuit_{\ell-2}(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 3, y', y) + \boxed{\heartsuit_{\ell-3}(y')}$$

Recurrence

First, think about the *score* of the best sequence.

Let $\heartsuit_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$\heartsuit_\ell(y) = s(\boldsymbol{x}, \ell, y, \octagon) + \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \boxed{\heartsuit_{\ell-1}(y')}$$

$$\heartsuit_{\ell-1}(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 2, y', y) + \boxed{\heartsuit_{\ell-2}(y')}$$

$$\heartsuit_{\ell-2}(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 3, y', y) + \boxed{\heartsuit_{\ell-3}(y')}$$

$$\vdots$$

$$\heartsuit_i(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, i - 1, y', y) + \boxed{\heartsuit_{i-1}(y')}$$

Recurrence

First, think about the *score* of the best sequence.
Let $\heartsuit_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$
\begin{aligned}
\heartsuit_\ell(y) &= s(\boldsymbol{x}, \ell, y, \bigcirc) + \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \boxed{\heartsuit_{\ell-1}(y')} \\
\heartsuit_{\ell-1}(y) &= \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 2, y', y) + \boxed{\heartsuit_{\ell-2}(y')} \\
\heartsuit_{\ell-2}(y) &= \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 3, y', y) + \boxed{\heartsuit_{\ell-3}(y')} \\
&\vdots \\
\heartsuit_i(y) &= \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, i - 1, y', y) + \boxed{\heartsuit_{i-1}(y')} \\
&\vdots \\
\heartsuit_1(y) &= s(\boldsymbol{x}, 0, \bigcirc, y)
\end{aligned}
$$

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y				
y'				
$\vdots$				
$y^{\Diamond}$				

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y	$\heartsuit_1(y)$			
y'	$\heartsuit_1(y')$			
$\vdots$				
$y^{\diamondsuit}$	$\heartsuit_1(y^{\diamondsuit})$			

$$\heartsuit_1(y) = s(\boldsymbol{x}, 0, \bigcirc, y)$$

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y	$\heartsuit_1(y)$	$\heartsuit_2(y)$		
y'	$\heartsuit_1(y')$	$\heartsuit_2(y')$		
$\vdots$				
$y^{\diamondsuit}$	$\heartsuit_1(y^{\diamondsuit})$	$\heartsuit_2(y^{\diamondsuit})$		

$$\heartsuit_i(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, i-1, y', y) + \boxed{\heartsuit_{i-1}(y')}$$

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y	$\heartsuit_1(y)$	$\heartsuit_2(y)$		$\heartsuit_\ell(y)$
y'	$\heartsuit_1(y')$	$\heartsuit_2(y')$		$\heartsuit_\ell(y')$
$\vdots$				
$y^{\diamondsuit}$	$\heartsuit_1(y^{\diamondsuit})$	$\heartsuit_2(y^{\diamondsuit})$		$\heartsuit_\ell(y^{\diamondsuit})$

$$\heartsuit_\ell(y) = s(\boldsymbol{x}, \ell, y, \bigcirc) + \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \boxed{\heartsuit_{\ell-1}(y')}$$

High-Level View of the Viterbi Algorithm

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!
- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ (and $y_{\ell-1}$) would be easy.
- Idea: for each position i, calculate the score of the best label prefix $\boldsymbol{y}_{1:i}$ ending in each possible value for Y_i.
- With a little bookkeeping, we can then trace backwards and recover the best label sequence.

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y				
y'				
$\vdots$				
$y^\diamondsuit$				

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y	$\heartsuit_1(y)$ $b_1(y)$			
y'	$\heartsuit_1(y')$ $b_1(y')$			
$\vdots$				
$y^{\diamondsuit}$	$\heartsuit_1(y^{\diamondsuit})$ $b_1(y^{\diamondsuit})$			

$$\heartsuit_1(y) = s(\boldsymbol{x}, 0, \bigcirc, y)$$
$$b_1(y) = \bigcirc$$

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y	$\heartsuit_1(y)$ $b_1(y)$	$\heartsuit_2(y)$ $b_2(y)$		
y'	$\heartsuit_1(y')$ $b_1(y')$	$\heartsuit_2(y')$ $b_2(y')$		
$\vdots$				
$y^\diamondsuit$	$\heartsuit_1(y^\diamondsuit)$ $b_1(y^\diamondsuit)$	$\heartsuit_2(y^\diamondsuit)$ $b_2(y^\diamondsuit)$		

$$\heartsuit_i(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, i-1, y', y) + \boxed{\heartsuit_{i-1}(y')}$$

$$b_i(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} s(\boldsymbol{x}, i-1, y', y) + \heartsuit_{i-1}(y')$$

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y	$\heartsuit_1(y)$ $b_1(y)$	$\heartsuit_2(y)$ $b_2(y)$		$\heartsuit_\ell(y)$ $b_\ell(y)$
y'	$\heartsuit_1(y')$ $b_1(y')$	$\heartsuit_2(y')$ $b_2(y')$		$\heartsuit_\ell(y')$ $b_\ell(y')$
$\vdots$				
$y^{\diamondsuit}$	$\heartsuit_1(y^{\diamondsuit})$ $b_1(y^{\diamondsuit})$	$\heartsuit_2(y^{\diamondsuit})$ $b_2(y^{\diamondsuit})$		$\heartsuit_\ell(y^{\diamondsuit})$ $b_\ell(y^{\diamondsuit})$

$$\heartsuit_\ell(y) = s(\boldsymbol{x}, \ell, y, \bigcirc) + \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \boxed{\heartsuit_{\ell-1}(y')}$$

$$b_\ell(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \heartsuit_{\ell-1}(y')$$

Full Viterbi Procedure

Input: scores $s(\boldsymbol{x}, i, y', y), \forall i \in \langle 0, \ldots, \ell \rangle, \forall y' \in \mathcal{L}, \forall y \in \mathcal{L}$

Output: $\hat{\boldsymbol{y}}$

1. Base case: $\heartsuit_1(y) = s(\boldsymbol{x}, 0, \bigcirc, y)$
2. For $i \in \langle 2, \ldots, \ell - 1 \rangle$:
   - Solve for $\heartsuit_i(*)$ and $b_i(*)$.

   $$\heartsuit_i(y) = \max_{y' \in \mathcal{L}} s(\boldsymbol{x}, i-1, y', y) + \heartsuit_{i-1}(y'), \quad b_i(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} s(\boldsymbol{x}, i-1, y', y) + \heartsuit_{i-1}(y')$$

3. Special case for the end:

   $$\heartsuit_\ell(y) = s(\boldsymbol{x}, \ell, y, \octagon) + \underbrace{\max_{y' \in \mathcal{L}} s(\boldsymbol{x}, \ell - 1, y', y) + \heartsuit_{\ell-1}(y')}_{b_\ell(y) \text{ is the "argmax"}}$$

4. $\hat{y}_\ell \leftarrow \operatorname*{argmax}_{y \in \mathcal{L}} \heartsuit_\ell(y)$
5. For $i \in \langle \ell - 1, \ldots, 1 \rangle$:
   - $\hat{y}_i \leftarrow b_{i+1}(\hat{y}_{i+1})$

Viterbi Asymptotics

Space: $O(|\mathcal{L}|\ell)$ for the algorithm, but $O(|\mathcal{L}|^2\ell)$ for the scores

Runtime: $O(|\mathcal{L}|^2\ell)$

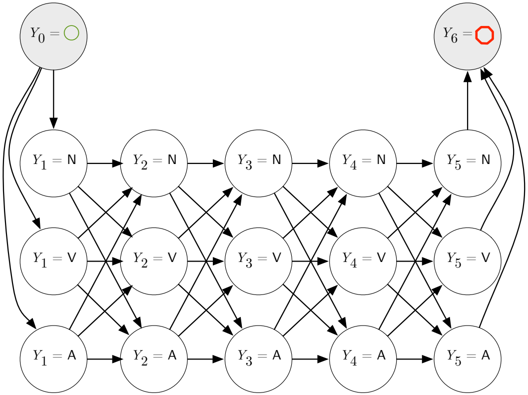
	x_1	x_2	$\ldots$	x_ℓ
y				
y'				
$\vdots$				
$y^\diamondsuit$				

Generalizing Viterbi

- Viterbi instantiates an general algorithm called **max-product variable elimination** for inference along a chain of variables with pairwise links.
  - Applicable to Bayesian networks and Markov networks.

Generalizing Viterbi

- Viterbi instantiates an general algorithm called **max-product variable elimination** for inference along a chain of variables with pairwise links.
- Viterbi solves a special case of the "best path" problem.



Generalizing Viterbi

- Viterbi instantiates an general algorithm called **max-product variable elimination** for inference along a chain of variables with pairwise links.
- Viterbi solves a special case of the "best path" problem.
- Dynamic programming algorithms.

Generalizing Viterbi

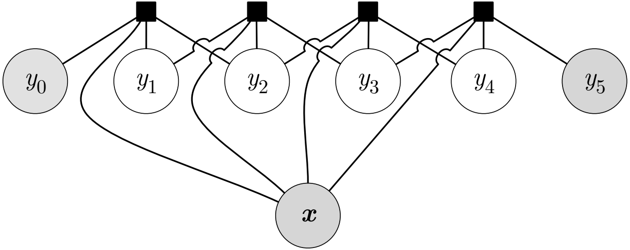
- Viterbi instantiates an general algorithm called **max-product variable elimination** for inference along a chain of variables with pairwise links.
- Viterbi solves a special case of the "best path" problem.
- Dynamic programming algorithms.
- Weighted finite-state analysis.

Version 3 (To Appear in Assignment 3)

Define scores of *triples* of adjacent word-labels in context: $s(\boldsymbol{x}, i, y'', y', y)$

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^\ell} \sum_{i=0}^{\ell-1} s(\boldsymbol{x}, i, y_i, y_{i+1}, y_{i+2})$$

This is known as a *second*-order model.



Version ∞

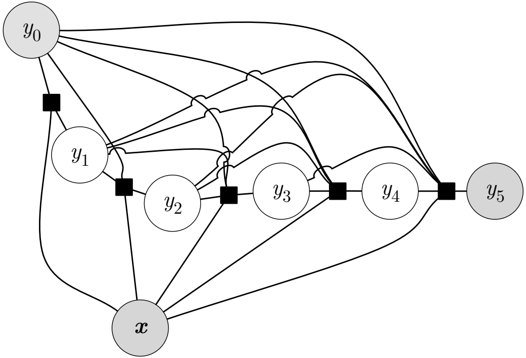
Define scores of word-labels that depend on all preceding labels: $s(\boldsymbol{x}, i, \boldsymbol{y}_{0:i})$

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^\ell} \sum_{i=0}^{\ell+1} s(\boldsymbol{x}, i, \boldsymbol{y}_{0:i})$$

Version ∞

Define scores of word-labels that depend on all preceding labels: $s(\boldsymbol{x}, i, \boldsymbol{y}_{0:i})$

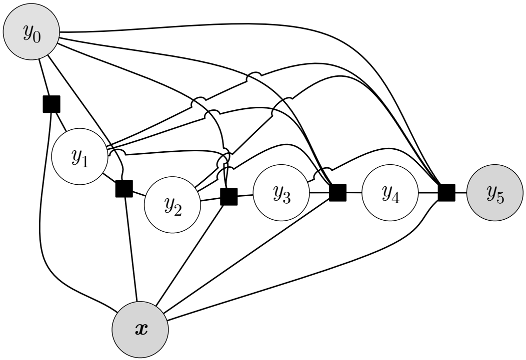
$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^\ell} \sum_{i=0}^{\ell+1} s(\boldsymbol{x}, i, \boldsymbol{y}_{0:i})$$



Version ∞

Define scores of word-labels that depend on all preceding labels: $s(\boldsymbol{x}, i, \boldsymbol{y}_{0:i})$

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^{\ell}} \sum_{i=0}^{\ell+1} s(\boldsymbol{x}, i, \boldsymbol{y}_{0:i})$$



Solving this problem exactly is hopeless; approximations required.

Natural Language Processing (CSE 447/547M): Sequence Model Applications

Noah Smith

University of Washington
nasmith@cs.washington.edu

February 11, 2019

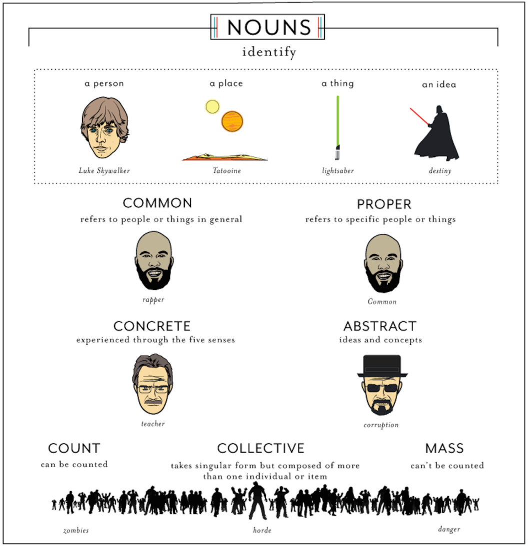
Applications of Sequence Models

- part-of-speech tagging (Church, 1988)
- supersense tagging (Ciaramita and Altun, 2006)
- named-entity recognition (Bikel et al., 1999)
- multiword expressions (Schneider and Smith, 2015)
- base noun phrase chunking (Sha and Pereira, 2003)

Along the way, we'll briefly mention two ways to *learn* sequence models.

Parts of Speech

http://mentalfloss.com/article/65608/master-particulars-grammar-pop-culture-primer



Parts of Speech

- "Open classes": Nouns, verbs, adjectives, adverbs, numbers
- "Closed classes":
  - Modal verbs
  - Prepositions (*on*, *to*)
  - Particles (*off*, *up*)
  - Determiners (*the*, *some*)
  - Pronouns (*she*, *they*)
  - Conjunctions (*and*, *or*)

Parts of Speech in English: Decisions

Granularity decisions regarding:

- verb tenses, participles
- plural/singular for verbs, nouns
- proper nouns
- comparative, superlative adjectives and adverbs

Some linguistic reasoning required:

- Existential *there*
- Infinitive marker *to*
- *wh* words (pronouns, adverbs, determiners, possessive *whose*)

Interactions with tokenization:

- Punctuation
- Compounds (*Mark'll*, *someone's*, *gonna*)

Penn Treebank: 45 tags, ∼40 pages of guidelines (Marcus et al., 1993)

Parts of Speech in English: Decisions

Granularity decisions regarding:

- verb tenses, participles
- plural/singular for verbs, nouns
- proper nouns
- comparative, superlative adjectives and adverbs

Some linguistic reasoning required:

- Existential *there*
- Infinitive marker *to*
- *wh* words (pronouns, adverbs, determiners, possessive *whose*)

Interactions with tokenization:

- Punctuation
- Compounds (*Mark'll*, *someone's*, *gonna*)
- Social media: hashtag, at-mention, discourse marker (*RT*), URL, emoticon, abbreviations, interjections, acronyms

Penn Treebank: 45 tags, ∼40 pages of guidelines (Marcus et al., 1993)

TweetNLP: 20 tags, 7 pages of guidelines (Gimpel et al., 2011)

Example: Part-of-Speech Tagging

ikr smh he asked fir yo last name

so he can add u on fb lololol

Example: Part-of-Speech Tagging

I know, right	shake my head			for	your		
ikr	smh	he	asked	fir	yo	last	name

				you		Facebook	laugh out loud
so	he	can	add	u	on	fb	lololol

Example: Part-of-Speech Tagging

I know, right	shake my head			for	your		
ikr	smh	he	asked	fir	yo	last	name
!	G	O	V	P	D	A	N
interjection	acronym	pronoun	verb	prep.	det.	adj.	noun

				you		Facebook	laugh out loud
so	he	can	add	u	on	fb	lololol
P	O	V	V	O	P	^	!
preposition						proper noun	

Why POS?

- Text-to-speech: *record*, *lead*, *protest*
- Lemmatization: *saw*/V $\rightarrow$ *see*; *saw*/N $\rightarrow$ *saw*
- Quick-and-dirty multiword expressions: (Adjective | Noun)* Noun (Justeson and Katz, 1995)
- Preprocessing for harder disambiguation problems:
  - *The Georgia branch had taken* **on** *loan commitments* ...
  - *The average of interbank* **offered** *rates plummeted* ...

A Simple POS Tagger

Define a map $\mathcal{V} \to \mathcal{L}$.

A Simple POS Tagger

Define a map $\mathcal{V} \rightarrow \mathcal{L}$.

How to pick the single POS for each word? E.g., *raises*, *Fed*, . . .

A Simple POS Tagger

Define a map $\mathcal{V} \rightarrow \mathcal{L}$.

How to pick the single POS for each word? E.g., *raises*, *Fed*, . . .

Penn Treebank: most frequent tag rule gives 90.3%, 93.7% if you're clever about handling unknown words.

A Simple POS Tagger

Define a map $\mathcal{V} \rightarrow \mathcal{L}$.

How to pick the single POS for each word? E.g., *raises*, *Fed*, ...

Penn Treebank: most frequent tag rule gives 90.3%, 93.7% if you're clever about handling unknown words.

All datasets have some errors; estimated upper bound for Penn Treebank is 98%.

Supervised Training of Hidden Markov Models

Given: annotated sequences $\langle\langle \boldsymbol{x}_1, \boldsymbol{y}_1, \rangle, \ldots, \langle \boldsymbol{x}_n, \boldsymbol{y}_n \rangle\rangle$

$$p(\boldsymbol{x}, \boldsymbol{y}) = \pi_{y_1} \prod_{i=1}^{\ell} \theta_{x_i|y_i} \cdot \gamma_{y_{i+1}|y_i}$$

Parameters: for each state/label $y \in \mathcal{L}$:

- $\boldsymbol{\pi}$ is the "start" distribution
- $\boldsymbol{\theta}_{*|y}$ is the "emission" distribution
- $\boldsymbol{\gamma}_{*|y}$ is called the "transition" distribution

Supervised Training of Hidden Markov Models

Given: annotated sequences $\langle\langle \boldsymbol{x}_1, \boldsymbol{y}_1, \rangle, \ldots, \langle \boldsymbol{x}_n, \boldsymbol{y}_n \rangle\rangle$

$$p(\boldsymbol{x}, \boldsymbol{y}) = \pi_{y_1} \prod_{i=1}^{\ell} \theta_{x_i|y_i} \cdot \gamma_{y_{i+1}|y_i}$$

Parameters: for each state/label $y \in \mathcal{L}$:

- $\boldsymbol{\pi}$ is the "start" distribution
- $\boldsymbol{\theta}_{*|y}$ is the "emission" distribution
- $\boldsymbol{\gamma}_{*|y}$ is called the "transition" distribution

Maximum likelihood estimate: count and normalize!

Back to POS

TnT, a trigram HMM tagger with smoothing: 96.7% (Brants, 2000)

Back to POS

TnT, a trigram HMM tagger with smoothing: 96.7% (Brants, 2000)

State of the art: ∼97.5% (Toutanova et al., 2003); uses a feature-based model with:

- capitalization features
- spelling features
- name lists ("gazetteers")
- context words
- hand-crafted patterns

Other Labels

Parts of speech are a minimal *syntactic* representation.

Sequence labeling can get you a lightweight *semantic* representation, too.

Supersenses

A problem with a long history: word-sense disambiguation.

Supersenses

A problem with a long history: word-sense disambiguation.

Classical approaches assumed you had a list of ambiguous words and their senses.

- E.g., from a dictionary

Supersenses

A problem with a long history: word-sense disambiguation.

Classical approaches assumed you had a list of ambiguous words and their senses.

- E.g., from a dictionary

Ciaramita and Johnson (2003) and Ciaramita and Altun (2006) used a lexicon called WordNet to define 41 semantic classes for words.

- WordNet (Fellbaum, 1998) is a fascinating resource in its own right! See `http://wordnetweb.princeton.edu/perl/webwn` to get an idea.

Supersenses

A problem with a long history: word-sense disambiguation.

Classical approaches assumed you had a list of ambiguous words and their senses.

- E.g., from a dictionary

Ciaramita and Johnson (2003) and Ciaramita and Altun (2006) used a lexicon called WordNet to define 41 semantic classes for words.

- WordNet (Fellbaum, 1998) is a fascinating resource in its own right! See `http://wordnetweb.princeton.edu/perl/webwn` to get an idea.

This represents a coarsening of the annotations in the Semcor corpus (Miller et al., 1993).

Example: *box*'s Thirteen Synonym Sets, Eight Supersenses

1. box: a (usually rectangular) container; may have a lid. “he rummaged through a box of spare parts”
2. box/loge: private area in a theater or grandstand where a small group can watch the performance. “the royal box was empty”
3. box/boxful: the quantity contained in a box. “he gave her a box of chocolates”
4. corner/box: a predicament from which a skillful or graceful escape is impossible. “his lying got him into a tight corner”
5. box: a rectangular drawing. “the flowchart contained many boxes”
6. box/boxwood: evergreen shrubs or small trees
7. box: any one of several designated areas on a ball field where the batter or catcher or coaches are positioned. “the umpire warned the batter to stay in the batter's box”
8. box/box seat: the driver's seat on a coach. “an armed guard sat in the box with the driver”
9. box: separate partitioned area in a public place for a few people. “the sentry stayed in his box to avoid the cold”
10. box: a blow with the hand (usually on the ear). “I gave him a good box on the ear”
11. box/package: put into a box. “box the gift, please”
12. box: hit with the fist. “I'll box your ears!”
13. box: engage in a boxing match.

Example: *box*'s Thirteen Synonym Sets, Eight Supersenses

1. box: a (usually rectangular) container; may have a lid. "he rummaged through a box of spare parts" $\rightsquigarrow$ N.ARTIFACT
2. box/loge: private area in a theater or grandstand where a small group can watch the performance. "the royal box was empty" $\rightsquigarrow$ N.ARTIFACT
3. box/boxful: the quantity contained in a box. "he gave her a box of chocolates" $\rightsquigarrow$ N.QUANTITY
4. corner/box: a predicament from which a skillful or graceful escape is impossible. "his lying got him into a tight corner" $\rightsquigarrow$ N.STATE
5. box: a rectangular drawing. "the flowchart contained many boxes" $\rightsquigarrow$ N.SHAPE
6. box/boxwood: evergreen shrubs or small trees $\rightsquigarrow$ N.PLANT
7. box: any one of several designated areas on a ball field where the batter or catcher or coaches are positioned. "the umpire warned the batter to stay in the batter's box" $\rightsquigarrow$ N.ARTIFACT
8. box/box seat: the driver's seat on a coach. "an armed guard sat in the box with the driver" $\rightsquigarrow$ N.ARTIFACT
9. box: separate partitioned area in a public place for a few people. "the sentry stayed in his box to avoid the cold" $\rightsquigarrow$ N.ARTIFACT
10. box: a blow with the hand (usually on the ear). "I gave him a good box on the ear" $\rightsquigarrow$ N.ACT
11. box/package: put into a box. "box the gift, please" $\rightsquigarrow$ V.CONTACT
12. box: hit with the fist. "I'll box your ears!" $\rightsquigarrow$ V.CONTACT
13. box: engage in a boxing match. $\rightsquigarrow$ V.COMPETITION

Supersense Tagging Example

Clara	Harris	,	one	of	the	guests	in	the
	N.PERSON					N.PERSON		

box	,	stood	up	and	demanded
N.ARTIFACT			V.MOTION		V.COMMUNICATION

water	.
N.SUBSTANCE	

Ciaramita and Altun's Approach

Features at each position in the sentence:

- word
- "first sense" from WordNet (also conjoined with word)
- POS, coarse POS
- shape (case, punctuation symbols, etc.)
- previous label

All of these fit into "$\boldsymbol{\phi}(\boldsymbol{x}, i, y, y')$."

Supervised Training of Sequence Models (Discriminative)

Given: annotated sequences $\langle\langle \boldsymbol{x}_1, \boldsymbol{y}_1, \rangle, \ldots, \langle \boldsymbol{x}_n, \boldsymbol{y}_n \rangle\rangle$

Assume:

$$\text{predict}(\boldsymbol{x}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^\ell} \underbrace{\sum_{i=0}^{\ell} s(\boldsymbol{x}, i, y_i, y_{i+1})}_{S(\boldsymbol{x}, \boldsymbol{y})}$$

Estimate: parameters of S

Perceptron

Perceptron algorithm for **classification**: Let $\mathbf{w}$ denote a vector containing *all* parameters of S.

- For $t \in \{1, \dots, T\}$:
  - Pick i_t uniformly at random from $\{1, \dots, n\}$.
  - $\hat{y}_{i_t} \leftarrow \underset{y \in \mathcal{L}}{\operatorname{argmax}}\, s(\boldsymbol{x}_{i_t}, y)$
  - $\mathbf{w} \leftarrow \mathbf{w} - \alpha \left(\nabla s(\boldsymbol{x}_{i_t}, \hat{y}_{i_t}) - \nabla s(\boldsymbol{x}_{i_t}, y_{i_t}) \right)$

Structured Perceptron

Collins (2002)

Perceptron algorithm for ~~classification~~ **structured prediction**: Let $\mathbf{w}$ denote a vector containing *all* parameters of S.

- For $t \in \{1, \ldots, T\}$:
  - Pick i_t uniformly at random from $\{1, \ldots, n\}$.
  - $\hat{\boldsymbol{y}}_{i_t} \leftarrow \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^{\ell}} S(\boldsymbol{x}, \boldsymbol{y})$
  - $\mathbf{w} \leftarrow \mathbf{w} - \alpha \left(\nabla S(\boldsymbol{x}_{i_t}, \hat{\boldsymbol{y}}_{i_t}) - \nabla S(\boldsymbol{x}_{i_t}, \boldsymbol{y}_{i_t}) \right)$

This can be viewed as stochastic subgradient descent on the *structured* hinge loss:

$$\sum_{i=1}^{n} \underbrace{\max_{\boldsymbol{y} \in \mathcal{L}^{\ell_i}} S(\boldsymbol{x}_i, \boldsymbol{y})}_{\text{fear}} - \underbrace{S(\boldsymbol{x}_i, \boldsymbol{y}_i)}_{\text{hope}}$$

Back to Supersenses

Clara	Harris	,	one	of	the	guests	in	the
	N.PERSON					N.PERSON		

box	,	stood	up	and	demanded
N.ARTIFACT			V.MOTION		V.COMMUNICATION

water	.
N.SUBSTANCE	

Shouldn't *Clara Harris* and *stood up* be respectively "grouped"?

Segmentations

Segmentation:

- Input: $\boldsymbol{x} = \langle x_1, x_2, \ldots, x_\ell \rangle$
- Output: $\left\langle \boldsymbol{x}_{1:\ell_1}, \boldsymbol{x}_{(1+\ell_1):(\ell_1+\ell_2)}, \boldsymbol{x}_{(1+\ell_1+\ell_2):(\ell_1+\ell_2+\ell_3)}, \ldots, \boldsymbol{x}_{(1+\sum_{i=1}^{m-1} \ell_i):\sum_{i=1}^{m} \ell_i} \right\rangle$

where $\ell = \sum_{i=1}^{m} \ell_i$.

Application: word segmentation for writing systems without whitespace.

Segmentations

Segmentation:

- Input: $\boldsymbol{x} = \langle x_1, x_2, \ldots, x_\ell \rangle$
- Output: $\left\langle \boldsymbol{x}_{1:\ell_1}, \boldsymbol{x}_{(1+\ell_1):(\ell_1+\ell_2)}, \boldsymbol{x}_{(1+\ell_1+\ell_2):(\ell_1+\ell_2+\ell_3)}, \ldots, \boldsymbol{x}_{(1+\sum_{i=1}^{m-1} \ell_i):\sum_{i=1}^{m} \ell_i} \right\rangle$

where $\ell = \sum_{i=1}^{m} \ell_i$.

Application: word segmentation for writing systems without whitespace.

With arbitrarily long segments, this does not look like a job for $\boldsymbol{\phi}(\boldsymbol{x}, i, y, y')$!

Segmentation as Sequence Labeling

Ramshaw and Marcus (1995)

Two labels: B ("beginning of new segment"), I ("inside segment")

- $\ell_1 = 4, \ell_2 = 3, \ell_3 = 1, \ell_4 = 2 \longrightarrow \langle$B, I, I, I, B, I, I, B, B, I$\rangle$

Three labels: B, I, O ("outside segment")

Five labels: B, I, O, E ("end of segment"), S ("singleton")

Segmentation as Sequence Labeling

Ramshaw and Marcus (1995)

Two labels: B (“beginning of new segment”), I (“inside segment”)

- $\ell_1 = 4, \ell_2 = 3, \ell_3 = 1, \ell_4 = 2 \longrightarrow \langle$B, I, I, I, B, I, I, B, B, I$\rangle$

Three labels: B, I, O (“outside segment”)

Five labels: B, I, O, E (“end of segment”), S (“singleton”)

Bonus: combine these with a label to get *labeled* segmentation!

Named Entity Recognition as Segmentation and Labeling

An older and narrower subset of supersenses used in information extraction:

- person,
- location,
- organization,
- geopolitical entity,
- ... and perhaps domain-specific additions.

AllenNLP demo of two strong systems:
`https://demo.allennlp.org/named-entity-recognition`

Named Entity Recognition

With $\underbrace{\text{Commander Chris Ferguson}}_{\text{person}}$ at the helm ,

$\underbrace{\text{Atlantis}}_{\text{spacecraft}}$ touched down at $\underbrace{\text{Kennedy Space Center}}_{\text{location}}$.

Named Entity Recognition

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

Atlantis	touched	down	at	Kennedy	Space	Center	.
spacecraft				location			
B	O	O	O	B	I	I	O

Named Entity Recognition: Evaluation

	1	2	3	4	5	6	7	8	9
$\boldsymbol{x}$ =	Britain	sent	warships	across	the	English	Channel	Monday	to
$\boldsymbol{y}$ =	B	O	O	O	O	B	I	B	O
$\boldsymbol{y}'$ =	O	O	O	O	O	B	I	B	O

10	11	12	13	14	15	16	17	18	19
rescue	Britons	stranded	by	Eyjafjallajökull	's	volcanic	ash	cloud	.
O	B	O	O	B	O	O	O	O	O
O	B	O	O	B	O	O	O	O	O

Segmentation Evaluation

Typically: precision, recall, and F_1.

Multiword Expressions

Schneider et al. (2014b)

- **MW compounds:** *red tape*, *motion picture*, *daddy longlegs*, *Bayes net*, *hot air balloon*, *skinny dip*, *trash talk*
- **verb-particle:** *pick up*, *dry out*, *take over*, *cut short*
- **verb-preposition:** *refer to*, *depend on*, *look for*, *prevent from*
- **verb-noun(-preposition):** *pay attention (to)*, *go bananas*, *lose it*, *break a leg*, *make the most of*
- **support verb:** *make decisions*, *take breaks*, *take pictures*, *have fun*, *perform surgery*
- **other phrasal verb:** *put up with*, *miss out (on)*, *get rid of*, *look forward to*, *run amok*, *cry foul*, *add insult to injury*, *make off with*
- **PP modifier:** *above board*, *beyond the pale*, *under the weather*, *at all*, *from time to time*, *in the nick of time*
- **coordinated phrase:** *cut and dry*, *more or less*, *up and leave*
- **conjunction/connective:** *as well as*, *let alone*, *in spite of*, *on the face of it/on its face*
- **semi-fixed VP:** *smack* <one>*'s lips*, *pick up where* <one> *left off*, *go over* <thing> *with a fine-tooth(ed) comb*, *take* <one>*'s time*, *draw* <oneself> *up to* <one>*'s full height*
- **fixed phrase:** *easy as pie*, *scared to death*, *go to hell in a handbasket*, *bring home the bacon*, *leave of absence*, *sense of humor*
- **phatic:** *You're welcome. Me neither!*
- **proverb:** *Beggars can't be choosers. The early bird gets the worm. To each his own. One man's* $<\text{thing}_1>$ *is another man's* $<\text{thing}_2>$.

Sequence Labeling with Nesting

Schneider et al. (2014a)

he	was	willing	to	budge$_1$	a$_2$	little$_2$	on$_1$	the	price
O	O	O	O	B	b	ī	Ī	O	O

which	means^4	a$_3^4$	lot$_3^4$	to^4	me^4	.
O	B	Ĩ	Ī	Ĩ	Ĩ	O

Strong (subscript) vs. weak (superscript) MWEs.

One level of nesting, plus strong/weak distinction, can be handled with an eight-tag scheme.

Back to Syntax

Base noun phrase chunking:

[He]$_{NP}$ reckons [the current account deficit]$_{NP}$ will narrow to [only \$ 1.8 billion]$_{NP}$ in [September]$_{NP}$

(What is a base noun phrase?)

"Chunking" used generically includes base verb and prepositional phrases, too.

Sequence labeling with BIO tags and features can be applied to this problem (Sha and Pereira, 2003).

Remarks

Sequence models are extremely useful:

- syntax: part-of-speech tags, base noun phrase chunking
- semantics: supersense tags, named entity recognition, multiword expressions

All of these are called "shallow" methods (why?).

Remarks

Sequence models are extremely useful:

- syntax: part-of-speech tags, base noun phrase chunking
- semantics: supersense tags, named entity recognition, multiword expressions

All of these are called "shallow" methods (why?).

Issues to be aware of:

- Supervised data for these problems is not cheap.
- Performance always suffers when you test on a different style, genre, dialect, etc. than you trained on.
- Runtime depends on the size of $\mathcal{L}$ and the number of consecutive labels that features can depend on.

References I

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34(1–3):211–231, 1999.

Thorsten Brants. TnT – a statistical part-of-speech tagger. In *Proc. of ANLP*, 2000.

Kenneth W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ANLP*, 1988.

Massimiliano Ciaramita and Yasemin Altun. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proc. of EMNLP*, 2006.

Massimiliano Ciaramita and Mark Johnson. Supersense tagging of unknown nouns in WordNet. In *Proc. of EMNLP*, 2003.

Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP*, 2002.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proc. of ACL*, 2011.

John S. Justeson and Slava M. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27, 1995.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

References II

G. A. Miller, C. Leacock, T. Randee, and R. Bunker. A semantic concordance. In *Proc. of HLT*, 1993.

Lance A Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning, 1995. URL `http://arxiv.org/pdf/cmp-lg/9505040.pdf`.

Nathan Schneider and Noah A. Smith. A corpus and model integrating multiword expressions and supersenses. In *Proc. of NAACL*, 2015.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. Discriminative lexical semantic segmentation with gaps: Running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206, April 2014a.

Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta Conrad, and Noah A. Smith. Comprehensive annotation of multiword expressions in a social web corpus. In *Proc. of LREC*, 2014b.

Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proc. of NAACL*, 2003.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of NAACL*, 2003.