Natural Language Processing (CSE 517): Sequence Models (II)

Noah Smith

University of Washington
nasmith@cs.washington.edu

February 3, 2016

Full Viterbi Procedure

Input: $\boldsymbol{x}$, $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\pi}$

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

1. For $i \in \langle 1, \ldots, \ell \rangle$:
   - Solve for $s_i(*)$ and $b_i(*)$.
     - Special base case for $i = 1$ to handle $\boldsymbol{\pi}$ (base case)
     - General recurrence for $i \in \langle 2, \ldots, \ell - 1 \rangle$

$$s_i(y) = \theta_{x_i|y} \cdot \max_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot s_{i-1}(y')$$

$$b_i(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot s_{i-1}(y')$$

     - Special case for $i = \ell$ to handle stopping probability
2. $\hat{y}_\ell \leftarrow \operatorname*{argmax}_{y \in \mathcal{L}} s_\ell(y)$
3. For $i \in \langle \ell, \ldots, 1 \rangle$:
   - $\hat{y}_{i-1} \leftarrow b(y_i)$

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

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

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y	$s_1(y)$ $b_1(y)$			
y'	$s_1(y')$ $b_1(y')$			
$\vdots$				
y^{last}	$s_1(y^{last})$ $b_1(y^{last})$			

$$s_1(y) = \theta_{x_1|y} \cdot \max_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot \pi_{y'}$$

$$b_1(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot \pi_{y'}$$

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y	$s_1(y)$ $b_1(y)$	$s_2(y)$ $b_2(y)$		
y'	$s_1(y')$ $b_1(y')$	$s_2(y')$ $b_2(y')$		
$\vdots$				
y^{last}	$s_1(y^{last})$ $b_1(y^{last})$	$s_2(y^{last})$ $b_2(y^{last})$		

$$s_i(y) = \theta_{x_i|y} \cdot \max_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot \boxed{s_{i-1}(y')}$$

$$b_i(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot s_{i-1}(y')$$

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\dots$	x_ℓ
y	$s_1(y)$ $b_1(y)$	$s_2(y)$ $b_2(y)$		$s_\ell(y)$ $b_\ell(y)$
y'	$s_1(y')$ $b_1(y')$	$s_2(y')$ $b_2(y')$		$s_\ell(y')$ $b_\ell(y')$
$\vdots$				
y^{last}	$s_1(y^{last})$ $b_1(y^{last})$	$s_2(y^{last})$ $b_2(y^{last})$		$s_\ell(y^{last})$ $b_\ell(y^{last})$

$$s_\ell(y) = \gamma_{\;|y} \cdot \theta_{x_\ell|y} \cdot \max_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot \boxed{s_{\ell-1}(y')}$$

$$b_\ell(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} \gamma_{y|y'} \cdot s_{\ell-1}(y')$$

Viterbi Asymptotics

Space: $O(|\mathcal{L}|\ell)$

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

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

Generalizing Viterbi

- Instead of HMM parameters, we can use the featurized variant.

$$s_i(y) = \max_{y' \in \mathcal{L}} \exp\left(\mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y, y')\right) \cdot s_{i-1}(y')$$

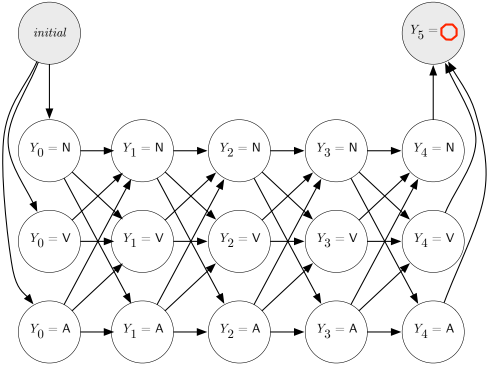
  More features may increase runtime, but asymptotic dependence on ℓ and $|\mathcal{L}|$ is the same.
  - For this case and for the HMM case, taking logarithms is a good idea.
  - Note that dependence on entirety of $\boldsymbol{x}$ doesn't affect asymptotics.

Generalizing Viterbi

- Instead of HMM parameters, we can use the featurized variant.
- 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

- Instead of HMM parameters, we can use the featurized variant.
- 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

- Instead of HMM parameters, we can use the featurized variant.
- 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.
- Higher-order dependencies among $\boldsymbol{Y}$ are also possible.

$$s_i(y, y') = \max_{y'' \in \mathcal{L}} \exp\left(\mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y, y', y'')\right) \cdot s_{i-1}(y', y'')$$

Generalizing Viterbi

- Instead of HMM parameters, we can use the featurized variant.
- 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.
- Higher-order dependencies among $\boldsymbol{Y}$ are also possible.
- Dynamic programming algorithms.

Generalizing Viterbi

- Instead of HMM parameters, we can use the featurized variant.
- 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.
- Higher-order dependencies among $\boldsymbol{Y}$ are also possible.
- Dynamic programming algorithms.
- Weighted finite-state analysis.

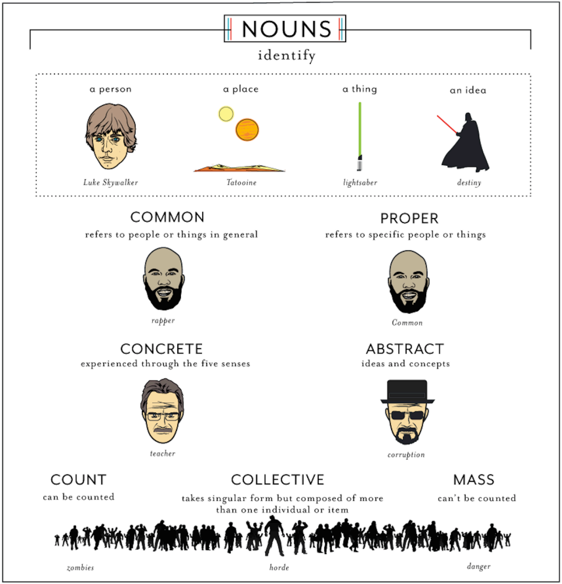
Applications of Sequence Models

- 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, $\sim$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 $\mathcal{V} \rightarrow \mathcal{L}$.

A Simple POS Tagger

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

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 $\langle\langle \boldsymbol{x}_1, \boldsymbol{y}_1, \rangle, \ldots, \langle \boldsymbol{x}_n, \boldsymbol{y}_n \rangle\rangle$

$$p(\boldsymbol{x}, \boldsymbol{y}) = \pi_{y_0} \prod_{i=1}^{\ell+1} \theta_{x_i|y_i} \cdot \gamma_{y_i|y_{i-1}}$$

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 $\langle\langle \boldsymbol{x}_1, \boldsymbol{y}_1, \rangle, \ldots, \langle \boldsymbol{x}_n, \boldsymbol{y}_n \rangle\rangle$

$$p(\boldsymbol{x}, \boldsymbol{y}) = \pi_{y_0} \prod_{i=1}^{\ell+1} \theta_{x_i|y_i} \cdot \gamma_{y_i|y_{i-1}}$$

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" ⇝ 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" ⇝ N.ARTIFACT
3. box/boxful: the quantity contained in a box. "he gave her a box of chocolates" ⇝ N.QUANTITY
4. corner/box: a predicament from which a skillful or graceful escape is impossible. "his lying got him into a tight corner" ⇝ N.STATE
5. box: a rectangular drawing. "the flowchart contained many boxes" ⇝ N.SHAPE
6. box/boxwood: evergreen shrubs or small trees ⇝ 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" ⇝ N.ARTIFACT
8. box/box seat: the driver's seat on a coach. "an armed guard sat in the box with the driver" ⇝ 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" ⇝ N.ARTIFACT
10. box: a blow with the hand (usually on the ear). "I gave him a good box on the ear" ⇝ N.ACT
11. box/package: put into a box. "box the gift, please" ⇝ V.CONTACT
12. box: hit with the fist. "I'll box your ears!" ⇝ V.CONTACT
13. box: engage in a boxing match. ⇝ V.COMPETITION

Readings and Reminders

- Jurafsky and Martin (2015)
- Submit a suggestion for an exam question by Friday at 5pm.
- Your project is due March 9.

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.

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.

Daniel Jurafsky and James H. Martin. Part-of-speech tagging (draft chapter), 2015. URL `https://web.stanford.edu/~jurafsky/slp3/9.pdf`.

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.

References II

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.

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

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

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.