CS 6501 Natural Language Processing

Sequence Labeling (I)

Yangfeng Ji

September 12, 2018

Department of Computer Science
University of Virginia

Overview

1. Problem Formulation
2. Sequential Modeling
3. Viterbi Decoding
4. Parameter Estimation
5. Applications

Problem Formulation

Part of Speech (POS)

- A way to categorize words with similar *grammatical* properties
- Common English POS tags
  - NOUN: used to name persons, things, animals, places etc.

    e.g., `Tom Hanks`, `yesterday`, `Grounds`
  - VERB: show an action or state

    e.g., `fight`, `was`
  - PRONOUN: replacement of nouns

    e.g., `she`, `his`, `it`, `theirs`
  - ADJECTIVE: used to describe a noun or a pronoun

    e.g., `large`, `beautiful`

Part of Speech (II)

- Common English POS tags (cont.)
  - ADVERB: used to describe adjectives, verbs, or another adverb

    e.g., `gracefully`, `yesterday`, `very`
  - PREPOSITION: specify location or a location in time

    e.g., `above`, `near`, `since`
  - CONJUNCTION: join words, phrases, or clauses together

    e.g., `and`, `for`
  - INTERJECTION: convey strong emotions

    e.g., `Ouch`, `Hey`

POS Tagging

Example

Teacher Strikes Idle Children

- Teacher$_{\texttt{Noun}}$ Strikes$_{\texttt{Noun}}$ Idle$_{\texttt{Verb}}$ Children$_{\texttt{Noun}}$
- Teacher$_{\texttt{Noun}}$ Strikes$_{\texttt{Verb}}$ Idle$_{\texttt{Adj}}$ Children$_{\texttt{Noun}}$

[Eisenstein, 2018, Chap 8]

Goal

From a training set, to learn a mapping f,

$$f : x \rightarrow y \tag{1}$$

where

- x: a sentence
- y: the POS tag sequence of x

Sequence Labeling as Classification

$$f : \boldsymbol{x} \rightarrow \boldsymbol{y} \tag{2}$$

For example

- $\boldsymbol{x}$: entire sentence
- $\boldsymbol{y}$: entire sequence
- $P(\boldsymbol{y}|\boldsymbol{x})$

Example

Teacher$_{\texttt{Noun}}$ Strikes$_{\texttt{Verb}}$ Idle$_{\texttt{Adj}}$ Children$_{\texttt{Noun}}$

Sequence Labeling as Classification

$$f : x \to y \tag{2}$$

For example

- x: only one token
- y: only the corresponding tag
- $P(y_i|x_i)$

Example

Teacher$_{\texttt{Noun}}$ Strikes$_{\texttt{Verb}}$ Idle$_{\texttt{Adj}}$ Children$_{\texttt{Noun}}$

Sequence Labeling as Classification

Example

`The trash can is in the garage`

- f: logistic regression
- x: the target token (e.g., `can`) and its surroundings (e.g., `trash, is`)
  - MODAL VERB: can be
  - NOUN
- y: the corresponding tag

Sequential Decision

Example

`they can fish`

- $\texttt{they}_{\texttt{Pronoun}}$ $\texttt{can}_{\texttt{Modal\_Verb}}$ $\texttt{fish}_{\texttt{Verb}}$
- $\texttt{they}_{\texttt{Pronoun}}$ $\texttt{can}_{\texttt{Verb}}$ $\texttt{fish}_{\texttt{Noun}}$

[Eisenstein, 2018]

Sequential Decision

Example

`they can fish`

- they$_{\texttt{Pronoun}}$ can$_{\texttt{Modal_Verb}}$ fish$_{\texttt{Verb}}$
- they$_{\texttt{Pronoun}}$ can$_{\texttt{Verb}}$ fish$_{\texttt{Noun}}$

The dependency between $\{y_i\}$

[Eisenstein, 2018]

Sequential Modeling

Generative Models

- observation $\boldsymbol{x}$
- target variable $\boldsymbol{y}$

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y}) \tag{3}$$

Inference: Bayes rule

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y})}{P(\boldsymbol{x})} \tag{4}$$

$P(\boldsymbol{y})$

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y}) \tag{5}$$

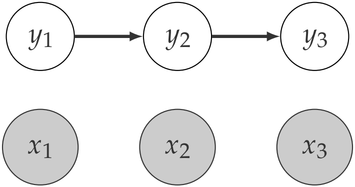
$P(\boldsymbol{y})$

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y}) \tag{5}$$

Factorization

$$P(\boldsymbol{y}) = \prod_{i=1} \underbrace{P(y_i|y_{i-1})}_{\text{Transition probability, Markov chain}} \tag{6}$$

Graphical model



$P(\boldsymbol{x}|\boldsymbol{y})$

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y}) \tag{7}$$

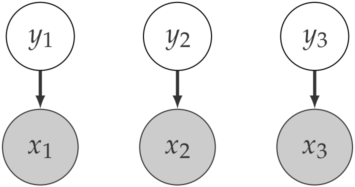
$P(\boldsymbol{x}|\boldsymbol{y})$

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y}) \tag{7}$$

Factorization

$$P(\boldsymbol{x}|\boldsymbol{y}) = \prod_{i=1} \underbrace{P(x_i|y_i)}_{\text{Emission probability}} \tag{8}$$

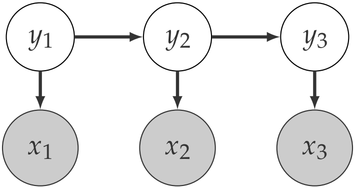
Graphical model



Hidden Markov Models

$$P(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1} \left\{ P(y_i | y_{i-1}) P(x_i | y_i) \right\} \tag{9}$$

Graphical model



- $\boldsymbol{x}$: observation (e.g., sentences)
- $\boldsymbol{y}$: hidden variables (e.g., POS sequences)

Viterbi Decoding

Formulation

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{x}, \boldsymbol{y}) \tag{10}$$

Formulation

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{x}, \boldsymbol{y}) \tag{10}$$

Dependency

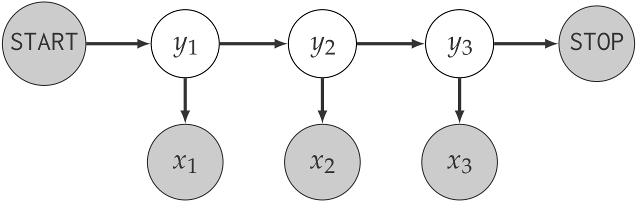
$$P(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1} \left\{ P(y_i | y_{i-1}) P(x_i | y_i) \right\} \tag{11}$$

The value of y_i depends on

- y_{i-1} via $P(y_i | y_{i-1})$
- y_{i+1} via $P(y_{i+1} | y_i)$
- x_i via $P(x_i | y_i)$

START and STOP

Graphical model



Factorization

Factorize $P(\boldsymbol{x}, \boldsymbol{y})$ with respect to (x_i, y_i)

$$
\begin{aligned}
P(\boldsymbol{x}, \boldsymbol{y}) =& P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}) \cdot P(x_i, y_i | \boldsymbol{y}_{\leq i-1}) \cdot P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1} | y_i) \\
=& P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}) \cdot P(x_i | y_i) \cdot P(y_i | y_{i-1}) \\
& \cdot P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1} | y_i)
\end{aligned}
$$

Factorization

Factorize $P(\boldsymbol{x}, \boldsymbol{y})$ with respect to (x_i, y_i)

$$
\begin{aligned}
P(\boldsymbol{x}, \boldsymbol{y}) =& P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}) \cdot P(x_i, y_i | \boldsymbol{y}_{\leq i-1}) \cdot P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1} | y_i) \\
=& P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}) \cdot P(x_i | y_i) \cdot P(y_i | y_{i-1}) \\
& \cdot P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1} | y_i)
\end{aligned}
$$

Three components

$$
\underbrace{P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1})}_{\text{past}} \cdot \underbrace{P(x_i | y_i) \cdot P(y_i | y_{i-1})}_{\text{present}} \cdot \underbrace{P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1} | y_i)}_{\text{future}} \tag{12}
$$

Basic Idea of Decoding

Three components

$$\underbrace{P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1})}_{\text{past}} \cdot \underbrace{P(x_i | y_i) \cdot P(y_i | y_{i-1})}_{\text{present}} \cdot \underbrace{P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1} | y_i)}_{\text{future}} \tag{13}$$

Basic Idea of Decoding

Three components

$$\underbrace{P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1})}_{\text{past}} \cdot \underbrace{P(x_i|y_i) \cdot P(y_i|y_{i-1})}_{\text{present}} \cdot \underbrace{P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1}|y_i)}_{\text{future}} \tag{13}$$

- Forward enumerating:
  - Start from y_1, for every possible value of y_i, from the best path from y_{i-1}
  - $\arg\max_{y_{i-1}} P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}) \cdot P(x_i|y_i) \cdot P(y_i|y_{i-1})$
  - Depends on past and present states $\{\boldsymbol{y}_{\leq i}\}$

Basic Idea of Decoding

Three components

$$\underbrace{P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1})}_{\text{past}} \cdot \underbrace{P(x_i|y_i) \cdot P(y_i|y_{i-1})}_{\text{present}} \cdot \underbrace{P(\boldsymbol{x}_{\geq i+1}, \boldsymbol{y}_{\geq i+1}|y_i)}_{\text{future}} \tag{13}$$

- Forward enumerating:
  - Start from y_1, for every possible value of y_i, from the best path from y_{i-1}
  - $\arg\max_{y_{i-1}} P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}) \cdot P(x_i|y_i) \cdot P(y_i|y_{i-1})$
  - Depends on past and present states $\{\boldsymbol{y}_{\leq i}\}$
- Backward tracing:
  - Start from $y_T = \texttt{STOP}$, for a given y_{i+1} find the best y_i
  - Depends on future states $\boldsymbol{y}_{\geq i+1}$

A Few More Notations

- v_{i-1}: score function associated with the past states
- s_i: score function associated with the present state

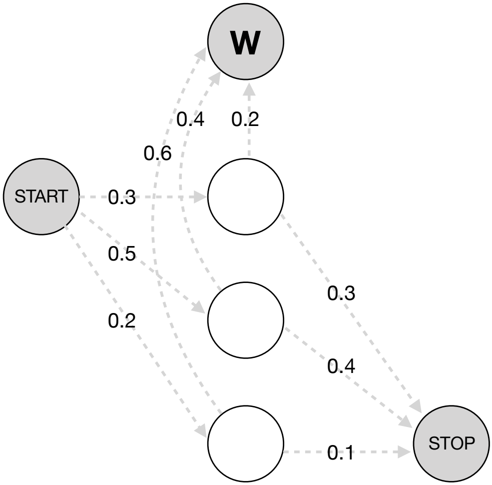
From

$$\arg\max_{y_{i-1}} P(\boldsymbol{x}_{\leq i-1}, \boldsymbol{y}_{\leq i-1}) \cdot P(x_i|y_i) \cdot P(y_i|y_{i-1})$$

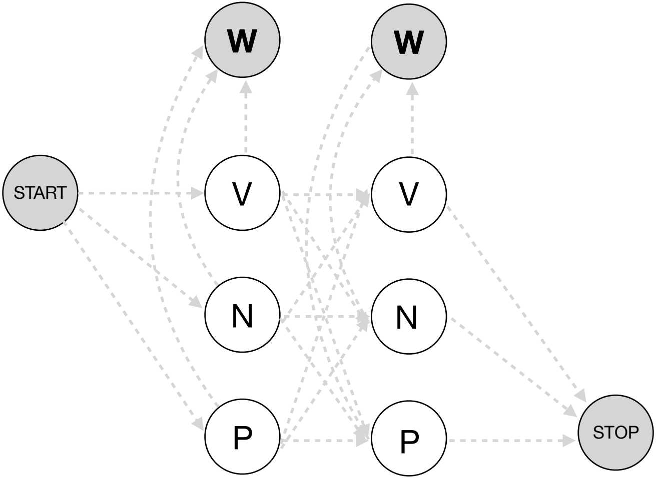
to

$$\arg\max_{y_{i-1}} s_i(y_i, y_{i-1}) + v_{i-1}(y_{i-1})$$

Example (I)



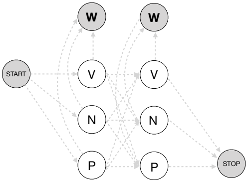
Example (II)



Viterbi Algorithm

Algorithm 11 The Viterbi algorithm. Each $s_m(k, k')$ is a local score for tag $y_m = k$ and $y_{m-1} = k'$.

> **for** $k \in \{0, \ldots K\}$ **do**
> $v_1(k) = s_1(k, \Diamond)$
> **for** $m \in \{2, \ldots, M\}$ **do**
>   **for** $k \in \{0, \ldots, K\}$ **do**
> $v_m(k) = \max_{k'} s_m(k, k') + v_{m-1}(k')$
> $b_m(k) = \mathrm{argmax}_{k'} s_m(k, k') + v_{m-1}(k')$
> $y_M = \mathrm{argmax}_k s_{M+1}(\blacklozenge, k) + v_M(k)$
> **for** $m \in \{M - 1, \ldots 1\}$ **do**
> $y_m = b_m(y_{m+1})$
> **return** $\boldsymbol{y}_{1:M}$



[Eisenstein, 2018]

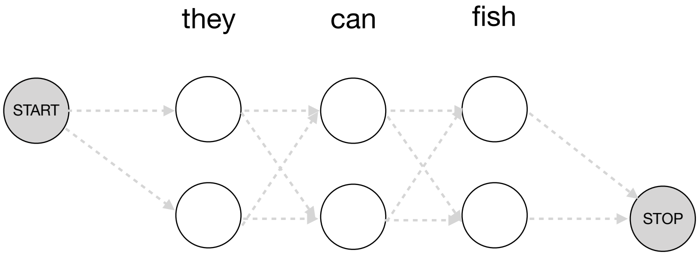
Example (III)

	they	*can*	*fish*
N	-2	-3	-3
V	-10	-1	-3

(a) Emission scores

	N	V	$\blacklozenge$
$\lozenge$	-1	-2	$-\infty$
N	-3	-1	-1
V	-1	-3	-1

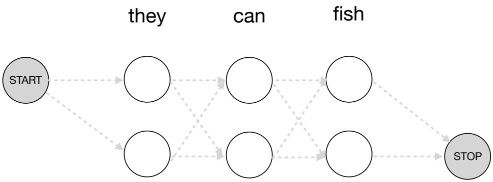
(b) Transition scores



(c) Trellis

Complexity

- T: sentence length
- K: possible tags



- $T \cdot K$ slots on trellis
- K computations for each slot

Therefore, total time complexity is $\mathcal{O}(TK^2)$

Parameter Estimation

Training Corpus

$$
\begin{aligned}
P(x_i|y_i) &=? \\
P(y_i|y_{i-1}) &=?
\end{aligned}
\tag{14}
$$

Training corpus

- $\texttt{they}_{\text{PRON}}$ $\texttt{can}_{\text{VERB}}$ $\texttt{fish}_{\text{NOUN}}$
- $\texttt{teacher}_{\text{NOUN}}$ $\texttt{strikes}_{\text{VERB}}$ $\texttt{idle}_{\text{ADJ}}$ $\texttt{children}_{\text{NOUN}}$
- ...

MLE

Transition probability

$$P(y_i|y_{i-1}) \approx \frac{c(y_i, y_{i-1})}{c(y_{i-1})} \tag{15}$$

Emission probability

$$P(x_i|y_i) \approx \frac{c(x_i, y_i)}{c(y_i)} \tag{16}$$

Applications

Parts of Speech

- *"Open classes"*
  - Nouns
  - Verbs
  - Adjectives
  - Adverbs
  - Numbers
- *"Closed classes"*
  - Modal verbs (e.g., `can, should`)
  - Prepositions (e.g., `on, to`)
  - Particles (e.g., `off, up`)
  - Determiners (e.g., `the, some`)
  - Pronouns (e.g., `she, they`)
  - Conjunctions (e.g., `and, or`)

[Smith, 2018]

Penn Treebank Tagset

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	“to”	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential ‘there’	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*’s*	”	Right quote	(’ or ”)
PRP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PRP$	Possessive pronoun	*your, one’s*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

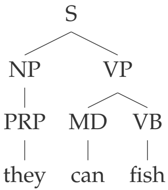
45 taggs, about 40 pages of guidelines [Marcus et al., 1993]

Why We Need POS?

- Disambiguation
  - they$_{\text{PRP}}$ can$_{\text{MD}}$ fish$_{\text{VB}}$

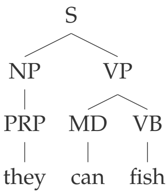
Why We Need POS?

- Disambiguation
  - they$_{PRP}$ can$_{MD}$ fish$_{VB}$
- Basic component for syntactic parsing



Why We Need POS?

- Disambiguation
  - $\texttt{they}_{\texttt{PRP}}\ \texttt{can}_{\texttt{MD}}\ \texttt{fish}_{\texttt{VB}}$
- Basic component for syntactic parsing



- Word prediction in speech recognition
  - Personal pronouns (`I`, `you`, `he`) are likely to be followed by verbs

Another Application: Named Entity Recognition

Example

`Atlantis touched down at Kennedy Space Center`

Another Application: Named Entity Recognition

Example

[Atlantis]$_{\text{MSIC}}$ touched down at [Kennedy Space Center]$_{\text{LOC}}$

Another Application: Named Entity Recognition

Example

`[Atlantis]`$_{\text{MSIC}}$ `touched down at` `[Kennedy Space Center]`$_{\text{LOC}}$

Tag set

- B: beginning
- I: inside
- O: outside

Category

- Person
- Location
- Organization
- Msic

Another Application: Named Entity Recognition

Example

`[Atlantis]`$_{\text{MSIC}}$ `touched down at [Kennedy Space Center]`$_{\text{LOC}}$

Tag set

- B: beginning
- I: inside
- O: outside

Category

- Person
- Location
- Organization
- Msic

Atlantis	touched	down	at	Kennedy	Space	Center	.
B_{MSIC}	O	O	O	B_{LOC}	I_{LOC}	I_{LOC}	O

Summary

- Sequence labeling problems
  - Part-of-Speech tagging
  - Named entity recognition
- Viterbi decoding
- Parameter estimation

Reference

Eisenstein, J. (2018).
Natural Language Processing.
MIT Press.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993).
Building a large annotated corpus of english: The penn treebank.
Computational linguistics, 19(2):313–330.

Smith, N. A. (2018).
Natural language processing: Lecture notes.