Introducción a Natural Language Processing

Modelos de Secuencia

Ronald Cárdenas Acosta

Setiembre, 2016

Outline

Predicción Estructurada y Modelos de Secuencia

- Usados en escenarios de Predicción Estructurada, en el que el modelo inferirá una estructura determinada.
- Las muestras pueden presentar dependencia espacial o temporal.
- Un modelo de secuencia modela una estructura (una cadena, árbol, etc).
- Estos modelos se usan muchas aplicaciones de NLP:
  - Tagging
  - Named Entity Recognition
  - Part-of-Speech Tagging
  - Shallow Parsing (Chunking)
  - Syntactic and Dependency Parsing
  - Machine Translation

Part-of-Speech Tagging

INPUT:

Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT:

Profits/N soared/V at/P Boeing/N Co./N ,/, easily/ADV topping/V forecasts/N on/P Wall/N Street/N ,/, as/P their/POSS CEO/N Alan/N Mulally/N announced/V first/ADJ quarter/N results/N ./.

- N: sustantivo
- V: verbo
- P: preposición
- Adv: adverbio
- Adj: adjetivo

Named Entity Recognition

Según informó el (Departamento)$_{\text{ORG}}$ *que dirige el consejero* (Inaxio Oliveri)$_{\text{PER}}$ *, los representantes de la* (Consejería)$_{\text{ORG}}$ *y de las universidades del* (País Vasco)$_{\text{LOC}}$, (Deusto)$_{\text{ORG}}$ *y* (Mondragón)$_{\text{ORG}}$ *estudiaron los nuevos retos de estos centros educativos.*

- PER: persona
- ORG: organización/empresa
- LOC: lugar

Notación

Sea el par de entrenamiento: $x^i, y^i = (x_1, ..., x_L), (y_1, ..., y_L)$

$V = w_1, ..., w_{\|V\|}$	Vocabulario, o conjunto de variables observadas
$S = s_1, ..., s_{\|S\|}$	Conjunto de etiquetas a predecir, o estados
L	Longitud de secuencia
$x = x_1, ...x_L$	Secuencia de observaciones
$y = y_1, ...y_L$	Secuencia de estados

Outline

Hidden Markov Model

- Es un modelo generativo.
- Es uno de los modelos probabilísticos más comunes.
- Es un tipo especial de *Probabilistic Graphical Model*, con estructura lineal o secuencial.
- Distingue las variables entre:
  - *variables observadas* $(x = x_1 ... x_L)$ o observaciones
  - *variables no observadas* $(y = y_1 ... y_L)$ o estados
- Consideración importante: las observaciones son independientes dados los estados que las generaron.

Hidden Markov Model

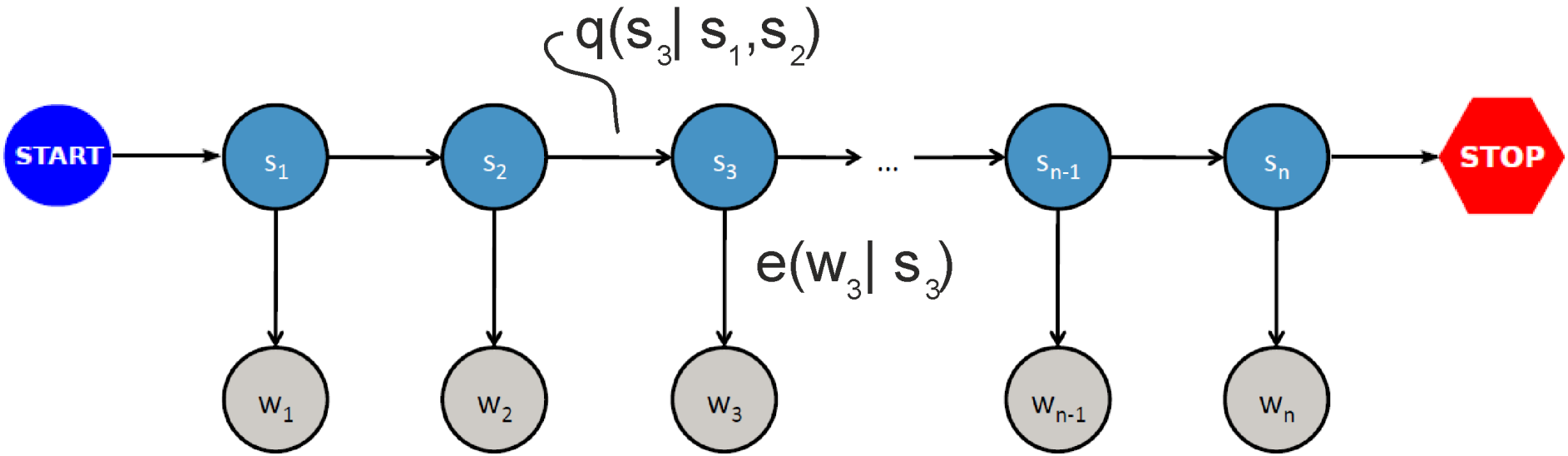

Figure: Modelo gráfico de un HMM.

Hidden Markov Model: Formulación

Sea el par de entrenamiento $x = [x_1, ..., x_L], y = [y_1, .., y_L]$

$$P(x_1, ..., x_L, y_1, ..., y_L) = \prod_{j=1}^{L} q(y_j|y_1, ..., y_{j-1}) \cdot e(w_j|y_j)$$

Donde:

- $q(y_j|y_1, ...y_{j-1})$: Probabilidad de transición de estado
- $e(w_j|y_j)$: Probabilidad de emisión de observación

Hidden Markov Model: Formulación

Para una cadena de Markov de primer grado (Bigram HMM):

$$P(x_1, ..., x_L, y_1, ..., y_L) = \prod_{j=1}^{L} q(y_j|y_{j-1}) \cdot e(w_j|y_j)$$

Consideraciones:

- El estado actual y_j solo depende del anterior y_{j-1}.
- La prob. de transición de un estado s_a a un estado s_b es independiente de la posición en la secuencia.
  $q(y_2 = s_a|y_1 = s_b) == q(y_5 = s_a|y_4 = s_b)$

Outline

HMM: Estimación de Parámetros

- Sea el conjunto de todos los parámetros θ
- El modelo HMM es entrenado para maximizar la probabilidad logaritmica de la data:

$$\arg\max_{\theta} \sum_{i=1}^{N} logP_{\theta}(X = x^i, Y = y^i) \tag{1}$$

HMM: Estimación de Parámetros

- Las probabilidades se aproximan por *Maximum Likelihood Estimate*, es decir, por cuentas.

$$q(s_j|s_{j-1}) = \frac{count(s_{j-1}, s_j)}{\sum_{k=1}^{|S|} count(s_k, s_j)}$$

$$e(w_j|s_j) = \frac{count(w_j, s_j)}{\sum_{v=1}^{|V|} count(w_v, s_j)}$$

Outline

Inferencia de una secuencia

- Dada la secuencia de observación $x = x_1, ..., x_L$, se busca la secuencia $\hat{y}$ óptima.
- La inferencia de una secuencia se conoce como **decoding**.
- Existen dos enfoques principales, ambos basados en programación dinámica.

Inferencia de una secuencia

Posterior Decoding o **Minimum risk decoding**
Para cada posición j de la secuencia

$$\hat{y}_j = \arg\max_{s \in S} P(y_j = s | x_1, ..., x_L)$$

- Minimiza la probabilidad de error para cada estado y_j, uno a la vez.
- No se garantiza que la secuencia final $\hat{y} = \hat{y}_1, ..., \hat{y}_L$ sea válida, es decir, puede haber una prob. de transición q igual a cero.
- Algoritmo conocido como *Forward-Backward*

Algoritmo Forward-backward

Algorithm 7 Forward-Backward algorithm

input: sequence $x_1, \ldots, x_N$, scores $P_{\text{init}}, P_{\text{trans}}, P_{\text{final}}, P_{\text{emiss}}$
Forward pass: Compute the forward probabilities
Initialization
for $c_k \in \Lambda$ **do**
  $\text{forward}(1, c_k) = P_{\text{init}}(c_k|\texttt{start}) \times P_{\text{emiss}}(x_1|c_k)$
end for
for $i = 2$ **to** N **do**
  for $c_k \in \Lambda$ **do**

$$\text{forward}(i, c_k) = \left(\sum_{c_l \in \Lambda} P_{\text{trans}}(c_k|c_l) \times \text{forward}(i-1, c_l) \right) \times P_{\text{emiss}}(x_i|c_k)$$

  end for
end for
Backward pass: Compute the backward probabilities
Initialization
for $c_l \in \Lambda$ **do**
  $\text{backward}(N, c_l) = P_{\text{final}}(\texttt{stop}|c_l)$
end for
for $i = N - 1$ **to** 1 **do**
  $\text{backward}(i, c_l) = \sum_{c_k \in \Lambda} P_{\text{trans}}(c_k|c_l) \times \text{backward}(i+1, c_k) \times P_{\text{emiss}}(x_{i+1}|c_k)$
end for
output: The forward and backward probabilities.

Inferencia de una secuencia

Viberbi Decoding

Consiste en hallar la secuencia de estados globalmente óptima.

$$\hat{y} = \underset{y=y_1 \cdots y_L}{\arg\max} P(x = x_1, ..., x_L; y = y_1, ..., y_L)$$

Utiliza explícitamente las consideraciones de independencia del HMM.

Algoritmo más utilizado en Predicción Estructurada en general.

Algoritmo de Viterbi

Algorithm 8 Viterbi algorithm

input: sequence $x_1, \ldots, x_N$, scores $P_{\text{init}}, P_{\text{trans}}, P_{\text{final}}, P_{\text{emiss}}$
Forward pass: Compute the best paths for every end state
Initialization
for $c_k \in \Lambda$ **do**
  $\text{viterbi}(1, c_k) = P_{\text{init}}(c_k|\texttt{start}) \times P_{\text{emiss}}(x_1|c_k)$
end for
for $i = 2$ **to** N **do**
  for $c_k \in \Lambda$ **do**
    $$\text{viterbi}(i, c_k) = \left(\max_{c_l \in \Lambda} P_{\text{trans}}(c_k|c_l) \times \text{viterbi}(i-1, c_l) \right) \times P_{\text{emiss}}(x_i|c_k)$$
    $$\text{backtrack}(i, c_k) = \left(\arg\max_{c_l \in \Lambda} P_{\text{trans}}(c_k|c_l) \times \text{viterbi}(i-1, c_l) \right)$$
  end for
end for
$\max_{y \in \Lambda^N} P(X = x, Y = y) := \max_{c_l \in \Lambda} P_{\text{final}}(\texttt{stop}|c_l) \times \text{viterbi}(N, c_l)$

Backward pass: backtrack to obtain the most likely path
$\widehat{y}_N = \arg\max_{c_l \in \Lambda} P_{\text{final}}(\texttt{stop}|c_l) \times \text{viterbi}(N, c_l)$
for $i = N - 1$ **to** 1 **do**
  $\widehat{y}_i = \text{backtrack}(i + 1, \widehat{y}_{i+1})$
end for
output: the viterbi path $\widehat{y}$.

Outline

Clasificadores Secuenciales Discriminativos

Se busca hallar la secuencia óptima $\hat{y}$

$$\begin{aligned}\hat{y} &= \arg\max_{y_1,\ldots,y_L} P(y|x) \\ &= \arg\max_{y_1,\ldots,y_L} w \cdot f(x,y) \\ &= \arg\max_{y_1,\ldots,y_L} \sum_{j=1}^{L} w \cdot f_{trans}(j, x_j, y_j, y_{j-1}) + w \cdot f_{emis}(x_j, y_j)\end{aligned}$$

Donde:

$$P(y|x) = \frac{1}{Z(x,y)} exp(\sum_{j=1}^{L} w \cdot f_{trans}(j, x_j, y_j, y_{j-1}) + w \cdot f_{emis}(x_j, y_j))$$

$Z(x,y)$ es la función partición

Outline

Features Básicas

Las features o características binarias se pueden dividir en dos grupos, de manera que imiten los parámetros de un HMM:
$f_{transicion}(j, x_j, y_j, y_{j-1})$ y $f_{emision}(x_j, y_j)$

Condición	Nombre
$y_j = s_k \wedge j = 1$	Feature de transición inicial
$y_j = s_k \wedge y_{j-1} = s_l$	Feature de transición
$y_j = s_k \wedge j = L$	Feature de transición final
$x_j = w_j \wedge y_j = s_k$	Feature de emisión

Features Extendidas

Se pueden definir features que:

- capturen morfología de la palabra: sufijos y prefijos
- dependan arbitrariamente de la secuencia entera de observación (p.e. x_{j-1}, x_j, x_{j+1})
- Información ortográfica: primera letra mayúscula, todo mayúscula, etc
- capturen la semántica de la palabra (p.e. id del cluster al que pertenece)
- Entre muchas más.

Features Extendidas

Condición	Nombre
$y_j = s_k \wedge j = 1$	Feature de trans. inicial
$y_j = s_k \wedge y_{j-1} = s_l$	Feature de transición
$y_j = s_k \wedge j = L$	Feature de trans. final
$x_j = w_j \wedge y_j = s_k$	Feature de emisión básica
$x_j = w_j \wedge w_j[0:p] \forall p \in [1,2,3] \wedge y_j = s_k$	Feature prefijos
$x_j = w_j \wedge w_j[L-p:L] \forall p \in [1,2,3] \wedge y_j = s_k$	Feature sufijos
$x_j = w_j \wedge w_j$ es todo mayuscula $\wedge y_j = s_k$	Feature ortografica: mayúscula
$x_j = w_j \wedge w_j$ tiene un digito $\wedge y_j = s_k$	Feature ortografica: digitos

Matriz de transición (Bigram HMM)

- Permite visualizar la distribución de probabilidad de transición
- Solo aplicable a HMM de primer grado (bigramas).
- Cada columna es el estado previo.
- Cada fila es el estado actual

Matriz de transición (Bigram HMM)

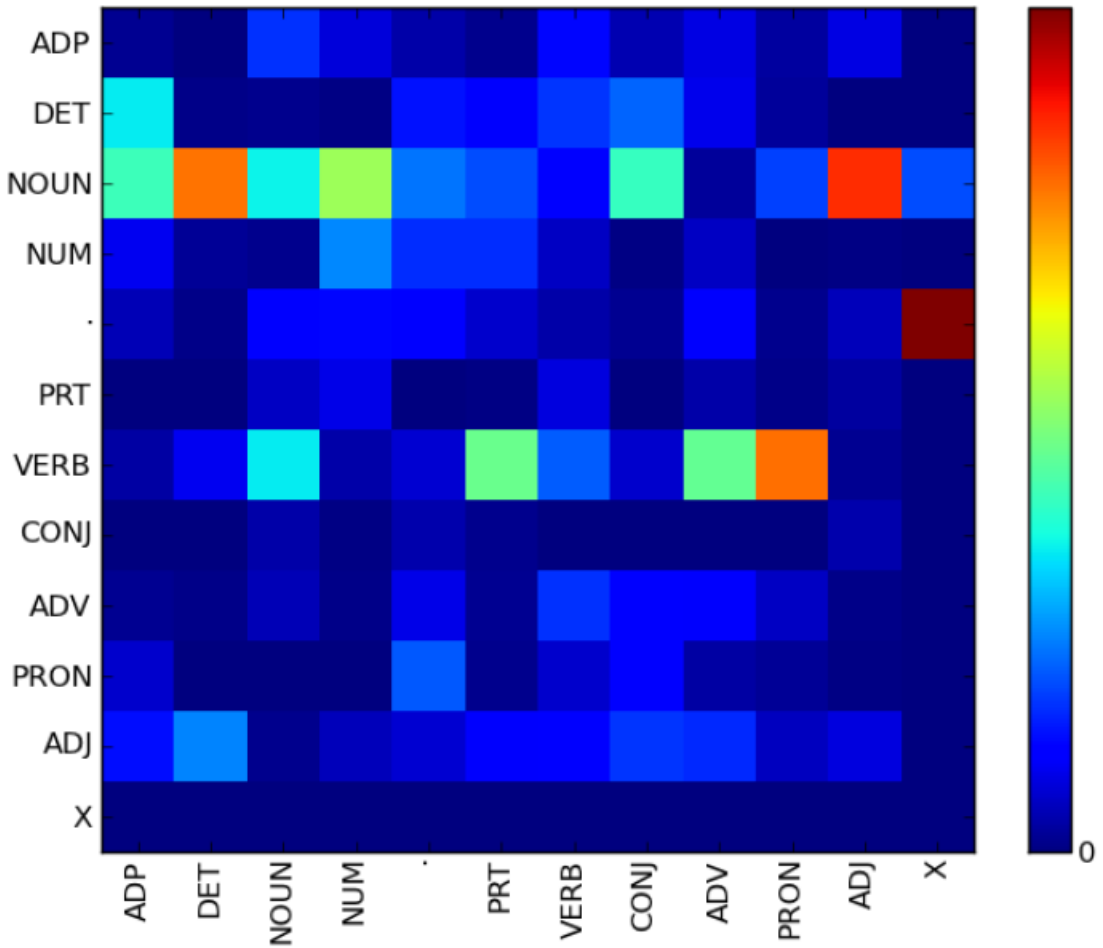

Figure: Matriz de transición de un modelo entrenado de POS Tagging. Columnas son estado anterior; filas, estado actual.

Matriz de Confusión

Permite visualizar los errores de clasificación por clase.

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Matriz de Confusión

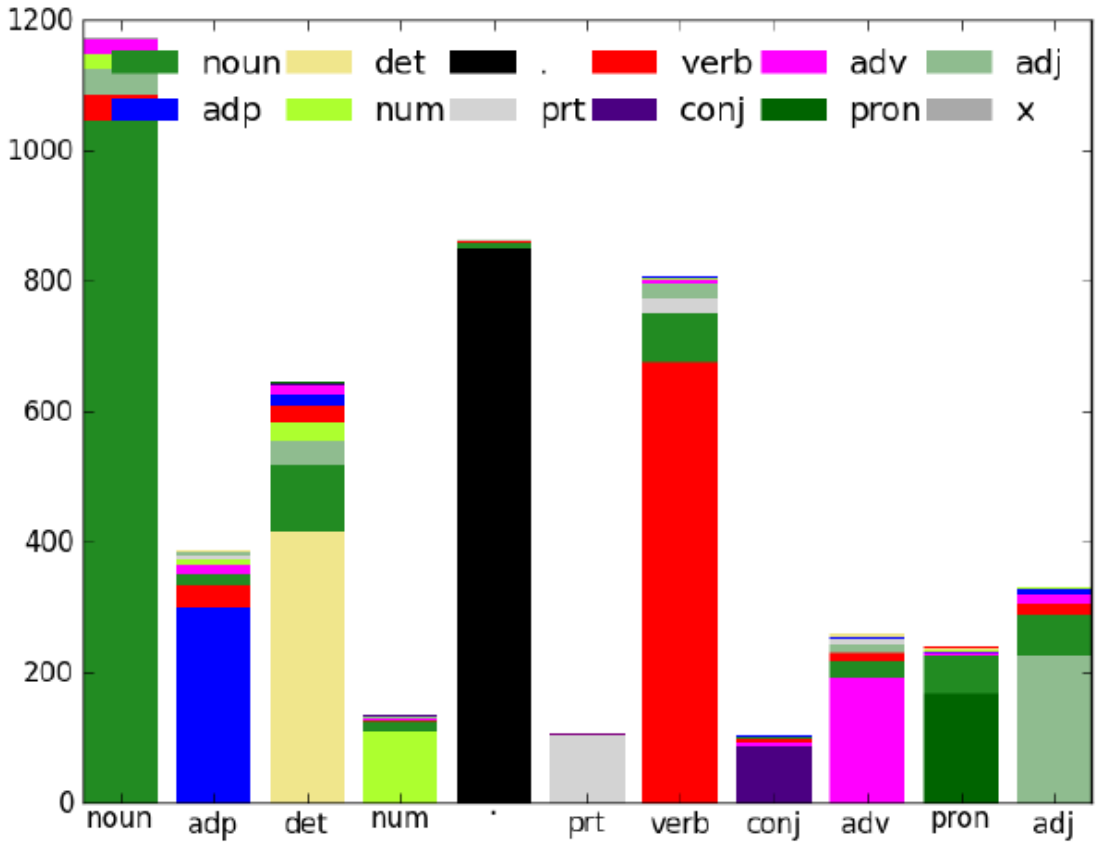

Figure: Representación en barras. Cada barra corresponde a una clase predecida y sus componentes a las clases verdaderas.

Named Entity Recognition

- **Entidad**: grupo de palabras que nombran una persona, lugar, organización, una fecha, número telefónico, etc
- **BIO-format / IOB-format**:
  - B-PER: palabra inicial de entidad Persona
  - I-PER: palabra miembro de entidad Persona
  - O: la palabra no es ninguna entidad

Named Entity Recognition

Figure: Ejemplo de texto anotado en formato BIO con entidades Persona, Lugar, y nombre de Nave.