Introducción a Natural Language Processing

Modelos de Lenguaje

Ronald Cárdenas Acosta

Setiembre, 2016

Outline

Outline

Modelos de Lenguaje (*Language Models*)

Objetivo

Asignar una probabilidad a una oración

Aplicaciones

- Traducción: P('atrapo al raton gato el') $<$ P('el gato atrapo al raton')
- Correción ortográfica: P('el gato atrapo al rtaon') $<$ P('el gato atrapo al raton')
- Reconocimiento de voz: P('el tubo de agua') $>$ P('el tuvo de agua')
- Generación de resumenes, question-answering, etc.

Modelos de Lenguaje Probabilísticos

- Se representa la oración como una secuencia de palabras.
  $P(S) = P(w_1, w_2, .., w_n)$
- Alternativamente, se puede hallar la probabilidad de la siguiente palabra:
  $P(w_3|w_1, w_2)$
- Un modelo que calcula cualquiera de estas probabilidades se llama *Modelo de Lenguaje*.

Outline

Recordatorio de Probabilidades

La regla de la cadena:

$$P(A, B, C, D) = P(A).P(B|A).P(C|A, B).P(D|A, B, C)$$

En general:

$$P(x_1, x_2, ...x_n) = P(x_1).P(x_2|x_1)...P(x_n|x_1, x_2, ..., x_{n-1})$$

Probabilidad de una oración: Regla de la cadena

$$P(w_1 w_2 ... w_n) = \prod_i P(w_i | w_1, w_2, ..., w_{i-1})$$

- Ejemplo:
  P(hoy esta soleado en Lima) =
  P(hoy).P(esta|hoy).P(soleado|hoy,esta).
  P(en|hoy,esta,soleado).P(Lima|hoy,esta,soleado,en)
- ¿Como estimar estas probabilidades? Conteo de ocurrencias en corpus
  $P(\text{Lima}|\text{hoy,esta,soleado,en}) = \frac{cuenta(hoy,esta,soleado,en,Lima)}{cuenta(hoy,esta,soleado,en)}$
- Si la oración es muy larga, el número de oraciones posibles se vuelve muy largo, y la cuenta muy baja.

Suposición de Markov

$$P(w_1 w_2 ... w_n) \approx \prod_i P(w_i | w_{i-k}, ..., w_{i-1})$$

- Donde k es el número de palabras anteriores a considerar (*grado de suposición de Markov*).
- Para $k = 1$:
  $P(Lima|hoy, esta, soleado, en) \approx P(Lima|en)$
- Para $k = 2$:
  $P(Lima|hoy, esta, soleado, en) \approx P(Lima|soleado, en)$

Modelos N-grama

- **Modelo Unigrama**: Cuando $k = 1$

$$P(w_1 w_2 ... w_n) \approx \prod_i P(w_i)$$

- **Modelo Bigrama**: Cuando $k = 2$

$$P(w_1 w_2 ... w_n) \approx \prod_i P(w_i | w_{i-1})$$

- **Modelo Trigrama**: Cuando $k = 3$

$$P(w_1 w_2 ... w_n) \approx \prod_i P(w_i | w_{i-2}, w_{i-1})$$

Outline

Estimación de Probabilidades: Bigramas

Recibe el nombre de *Maximum Likelihood Estimate* (estimación de máxima probabilidad)

$$P(w_i|w_{i-1}) = \frac{cuenta(w_{i-1}, w_i)}{cuenta(w_{i-1})}$$

Se suele agregar indicadores de inicio y termino de oración. Ejemplo: $< START >$ *hoy esta soleado en Lima* $< STOP >$. De manera que se debe considerar también:
$P(hoy| < START >)$: probabilidad que hoy sea la primera palabra
$P(< STOP > |Lima)$: probabilidad que Lima sea la última palabra

Outline

Evaluación de modelos de lenguaje

- Evaluación Extrínseca:
  evaluar la tarea en la que el modelo es parte (traducción, corrector ortográfico)
- Evaluación Intrínseca:
  estimar el grado de ajuste de las probabilidades a la data [Perplexity]

Perplexity

- Cuantifica las ramificaciones de las posibles combinaciones de palabras:
  "Factor de ramificación"
- Se calcula usando probabilidades logarítmicas para evitar *underflow*
- Minimizar el Perplexity es equivalente a maximizar la probabilidad
- Mientras menor Perplexity, mejor.

Perplexity

Probabilidad de la data de testeo, normalizado por el número de palabras.

$$PP(test) = P(w_1 w_2 ... w_N)^{-\frac{1}{N}}$$

Expresado normalmente en prob. logarítmicas ($log(P(test - data))$):

$$PP(test) = 2^{-L}$$

Donde:

$$L = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} log(p(w_i | w_1 ... w_{i-1}))$$