Generalización (Métodos Avanzados) y corrección ortográfica

Rubén Francisco Manrique
rf.manrique@uniandes.edu.co

Palabras desconocidas: Vocabularios cerrados vs vocabularios abiertos.

- Si sabemos todas las palabras de antemano:
  - Vocabulario V es fijo
  - *Tarea de vocabulario cerrado.*

- A menudo no sabemos esto
  - Fuera de vocabulario = palabras ***OOV***
  - *Tarea de vocabulario abierto*

- A veces se crea un token *<UNK>*, sin embargo, es difícil estimar la probabilidad de este token. ¿Deténgase un momento a pensar la razón?

Backoff e Interpolación

- Recuerden el problema de ***generalización,*** probabilidades de cero cuando un bigrama/trigama/4-gramas/5-gramas no visto en la colección es evaluado. ¡No podemos calcular perplejidad!
- Ya vimos el suavizado por Laplace.

- ***Backoff***
  - Use trigramas si tiene buena evidencia, en caso de que no, use bigramas, si no esta presente llegue al unigrama.

- ***Interpolación***
  - Mezclar unigramas, bigramas y trigramas.

Interpolación Lineal

- Interpolación simple:

$$\hat{P}(w_n|w_{n-1}w_{n-2}) = \lambda_1 P(w_n|w_{n-1}w_{n-2}) + \lambda_2 P(w_n|w_{n-1}) + \lambda_3 P(w_n)$$

$$\sum_i \lambda_i = 1$$

- Interpolación contextual:

$$\hat{P}(w_n|w_{n-2}w_{n-1}) = \lambda_1(w_{n-2}^{n-1}) P(w_n|w_{n-2}w_{n-1}) + \lambda_2(w_{n-2}^{n-1}) P(w_n|w_{n-1}) + \lambda_3(w_{n-2}^{n-1}) P(w_n)$$

Como escoger los lambdas?

- Use un conjunto de validación:

- Escoja los λs que maximizan la probabilidad en el dataset de validación:
  - Primero estime las probabilidades con el dataset de entrenamiento.
  - Luego, busque los mejores λs que maximizan la probabilidad:

$$\log P(w_1 \ldots w_n \mid M(l_1 \ldots l_k)) = \sum_i \log P_{M(l_1 \ldots l_k)}(w_i \mid w_{i-1})$$

Backoff Estúpido (Brants et al. 2007)

- Usado para la construcción del corpus de Google N-gram
- Poda:
  - Solo se guardan N-gramas con count > umbral (40-50)
  - Esto remueve singletons.
- Generalización:
  - No hay descuento use frecuencias relativas.

$$S(w_i \mid w_{i-k+1}^{i-1}) = \begin{cases} \dfrac{\text{count}(w_{i-k+1}^{i})}{\text{count}(w_{i-k+1}^{i-1})} & \text{if} \quad \text{count}(w_{i-k+1}^{i}) > 0 \\ 0.4\,S(w_i \mid w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases}$$

Caso unigramas

$$S(w_i) = \frac{\text{count}(w_i)}{N}$$

Suavizado Good-Turing

- Use las cosas/eventos/términos que ha visto una vez para ayudar a estimar las cosas que usted nunca ha visto.

- Notación: N_c= Frecuencia de frecuencia c
- N_c= el conteo de cosas que se han visto c veces.

- Manuel Yo soy Yo soy Manuel Yo no como

Yo	3
Manuel	2
soy	2
no	1
como	1

$N_1 = 2$

$N_2 = 2$

$N_3 = 1$

Suavizado Good-Turing Intuición

- Suponga que usted un pescador y captura en una zona los siguientes peces:
  - 10 carpas, 3 marlin, 2 aluvinas, 1 trucha, 1 salmon, 1 raya = 18

- Cual es la probabilidad que la siguiente especie capturada sea una trucha.
  - 1/18

- Cual es la probabilidad que la siguiente especie captura sea nueva (i.e. pez gato o bagre).
  - Usemos las cosas que vimos una vez para estimar las nuevas.
  - 3/18 porque $N_1=3$
- Problema: distribución de probabilidades. ¿Cuál es la probabilidad de que la siguiente especie sea una trucha? – Tiene que se menos que ***1/18, pero cuanto menos?***

Cálculos de Good-Turing

$$P^*_{GT}(frecuencia\ zero) = \frac{N_1}{N}$$

$$P^*_{GT}(vistos\ c\ veces) = \frac{c^*}{N}$$

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

- No vistos (bagre o pez gato)
- c=0
- MLE P=0/18=0
- $P^*_{GT}(no\ visto) = \frac{N_1}{N} = \frac{3}{18}$

- Visto 1 vez (trucha)
- c=1
- MLE P=1/18
- $c^*(trucha) = 2\frac{N_2}{N_1} = 2\frac{1}{3}$
- $P^*_{GT}(trucha) = \frac{c^*}{N} = \frac{2/3}{18} = 1/27$

Números de Good-Turing

- Números tomados de Church and Gale (2001)
- 22 millones de palabras AP Newswire.

$$c^* = \frac{(c+1)N_{c+1}}{N_c}$$

- Aproximación:

$$c^* = (c - .75)$$

Count c	Good Turing c*
0	.0000270
1	0.446
2	1.26
3	2.24
4	3.24
5	4.22
6	5.19
7	6.21
8	7.24
9	8.25

Descuento Interpolado absoluto

- Combinar Good Turing con interpolación.

discounted bigram Interpolation weight

$$P_{\text{AbsoluteDiscounting}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) - d}{c(w_{i-1})} + \lambda(w_{i-1})P(w)$$

unigram

- d es igual a 0.75.

Otras técnicas de suavizado avanzado

- Kneser-Ney
- Witten-Bell

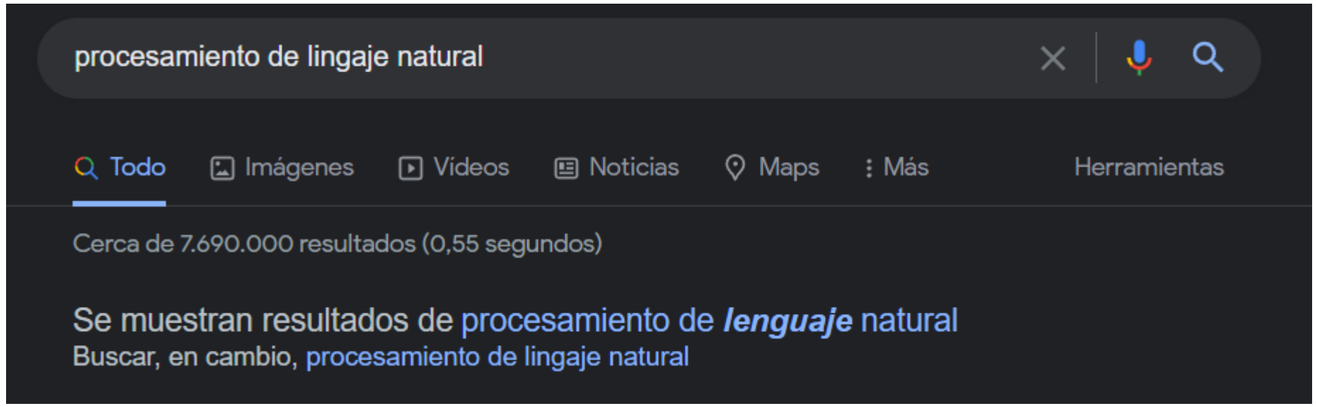
Importante sobre los modelos de n-gramas

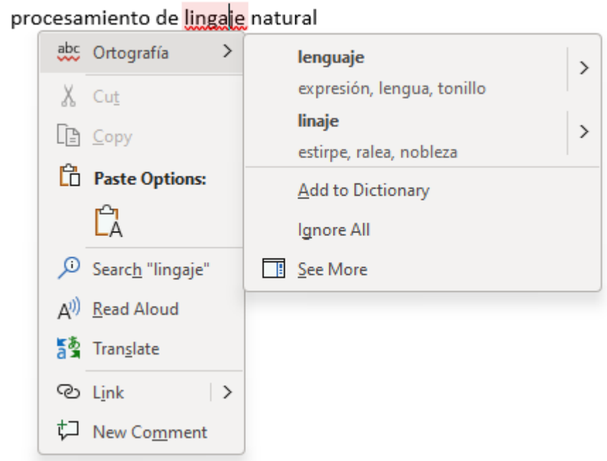
- Es un modelo GENERATIVO.

- Modelo Discriminativo:
  - Elija pesos de n-gramas para mejorar una tarea, no para ajustarse al conjunto de entrenamiento.

Aplicación
Corrección Ortográfica

Aplicaciones de corrección ortográfica.





Tareas de Ortografía

- Paso 1: Detección de errores.
- Paso 2: Corrección de errores.
  - Autocorrección
    - maor -> amor
  - Sugerencia de corrección.
  - Lista de sugerencias.

Tipos de errores Ortograficos

- Errores de ***NO-PALABRA***
  - maor -> amor
  - graffe -> giraffe
  - La palabra errada no se encuentra en V.
- Errores de ***PALABRA-REAL***
  - Tipográfico
    - casa->caza
    - three->there

Errores ortográficos de NO-PALABRA

- Detección de errores ortográficos que no son palabras:
  - Cualquier palabra que no esté en un diccionario es un error.
  - Cuanto más grande sea el diccionario, mejor.

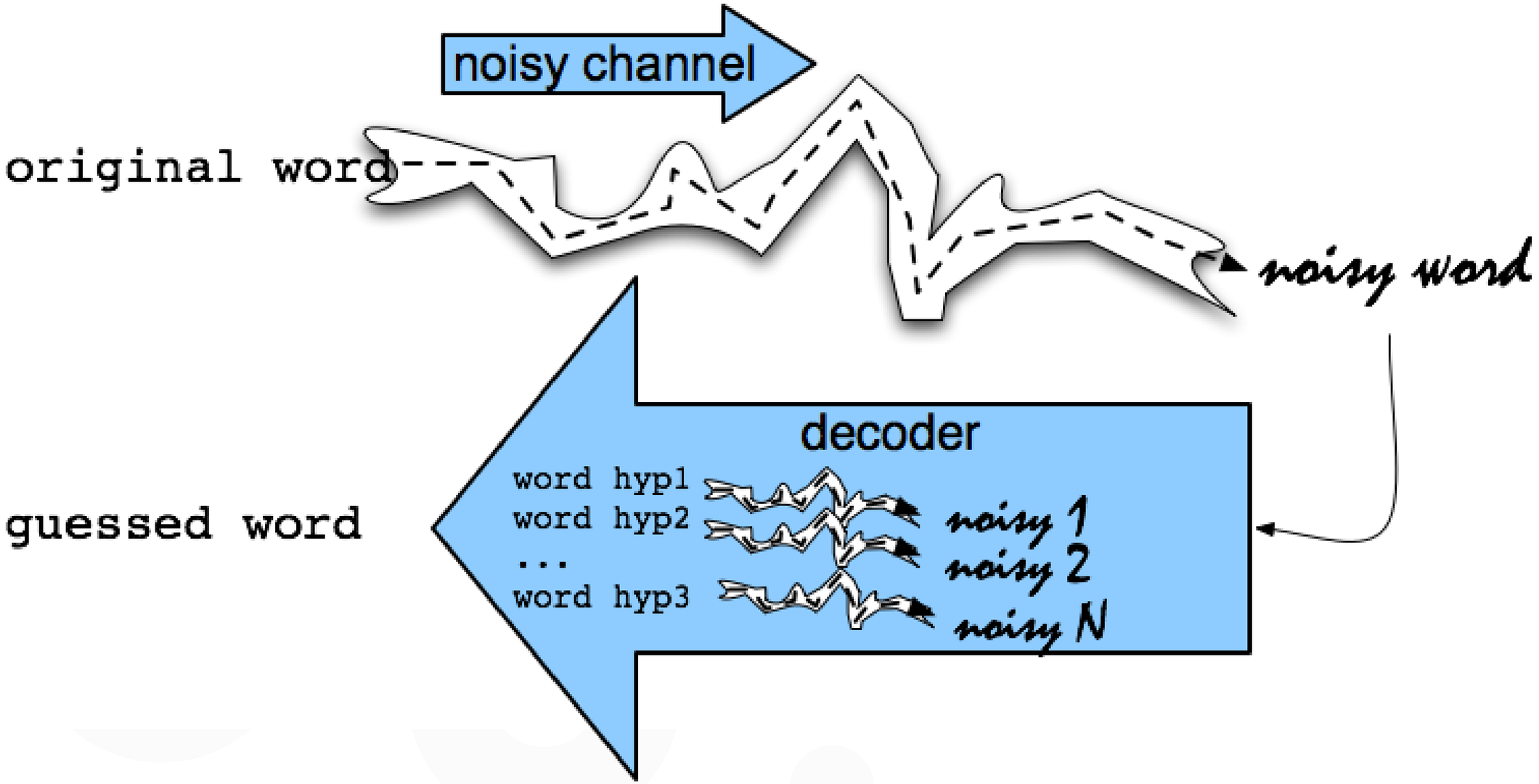
- Corrección de errores ortográficos que no son palabras:
  - Generar candidatos: palabras reales similares a error.
  - Elige la mejor:
    - Distancia de edición ponderada más corta.
    - *Mayor probabilidad de* ***canal ruidoso***

Errores ortográficos de PALABRA-REAL

- Para cada palabra w en la sentencia, generar una lista de candidatos:
  - Encuentre las palabras candidatas con escritura similar.
  - Incluir la misma palabra w en el conjunto candidato.

- Escoger el mejor candidato:
  - ***Canal ruidoso.***
  - Clasificador.

Modelo de Canal Ruidoso



Modelo de Canal Ruidoso

- Nosotros vemos una observación x de una palabra mal escrita.
- Encuentre la palabra correcta.

Estimador MAP
(Maximun a posteriori estimation)

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w \mid x)$$

$$= \underset{w \in V}{\operatorname{argmax}} \frac{P(x \mid w)P(w)}{P(x)}$$

$$= \underset{w \in V}{\operatorname{argmax}} P(x \mid w)P(w)$$

Modelo de lenguaje

Modelo de ruido

Caso 1: Errores ortográficos de NO-PALABRA

Caso 1: Errores ortográficos de NO-PALABRA

acress

- Generación de candidatos

- Palabras con escritura similar.
  - Una distancia de edición pequeña.

- Palabras con pronunciación similar
  - Distancia de edición de pronunciación similar.

Damerau-Levenshtein distancia edición

- Distancia de edición mínima entre dos strings, donde las ediciones son:

- Inserción
- Borrado
- Sustitución
- Transposición de dos letras adyacentes.

Palabras a una distancia de edición de 1
"acress"

Error	*Candidate Correction*	*Correct Letter*	*Error Letter*	*Type*
acress	actress	t	–	deletion
acress	cress	–	a	insertion
acress	caress	ca	ac	transposition
acress	access	c	r	substitution
acress	across	o	e	substitution
acress	acres	–	s	insertion
acress	acres	–	s	insertion

Generación de candidatos

- 80% de los errores esta a una distancia de edición de 1.
- 98% de los errores están a una distancia de edición de 2.

- Cuando se generan candidatos también se permite la inserción de espacio y guion.
  - estacuchara -> esta cuchara
  - inlaw -> in-law

- Recuerdan la formula: $= \underset{w \hat{I} V}{\operatorname{argmax}} P(x \mid w) P(w)$

- Modelo de lenguaje: Unigramas, bigramas, trigrmas,..,etc.

Probabilidad estimada según el modelo de canal ruidoso

- P(x|w) la vamos a estimar como la probabilidad de edición.
  - (borrado/inserción/sustitución/transposición)

- Suponga una palabra como una secuencia de caracteres:
  - Palabra mal escrita: $x = x_1, x_2, x_3, \ldots, x_n$
  - Palabra correcta: $\mathrm{w} = w_1, w_2, w_3, \ldots, w_n$

- Suponga que tenemos una base de entrenamiento con los errores de edición.
  - Peter Norvig's list of errors
  - Peter Norvig's list of counts of single-edit errors

Probabilidad estimada según el modelo de canal ruidoso

- Definamos las siguientes funciones:

```
del[x,y]:     count(xy typed as x)
ins[x,y]:     count(x typed as xy)
sub[x,y]:     count(x typed as y)
trans[x,y]:   count(xy typed as yx)
```

- Para cada una de estas matrices podemos construir una matriz de confusión.

Matriz de confusión para función sub[x,y]
Carácter x (incorrecto) sustituido por y (correcto)

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

X	Y (correct)																								
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

Modelo de canal

- Kernighan, Church, Gale 1990

$$P(x|w) = \begin{cases} \frac{\text{del}[w_{i-1}, w_i]}{\text{count}[w_{i-1} w_i]}, & \text{if deletion} \\ \frac{\text{ins}[w_{i-1}, x_i]}{\text{count}[w_{i-1}]}, & \text{if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, & \text{if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, & \text{if transposition} \end{cases}$$

Modelo de canal para "acress"

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|word)
actress	t	-	c|ct	.000117
cress	-	a	a|#	.00000144
caress	ca	ac	ac|ca	.00000164
access	c	r	r|c	.000000209
across	o	e	e|o	.0000093
acres	-	s	es|e	.0000321
acres	-	s	ss|s	.0000342

Probabilidad de modelo de canal ruidoso para "acress" – Modelo de lenguaje Unigramas

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|word)	P(word)	10^9 *P(x|w)P(w)
actress	t	-	c|ct	.000117	.0000231	2.7
cress	-	a	a|#	.00000144	.000000544	.00078
caress	ca	ac	ac|ca	.00000164	.00000170	.0028
access	c	r	r|c	.000000209	.0000916	.019
across	**o**	**e**	**e|o**	**.0000093**	**.000299**	**2.8**
acres	-	s	es|e	.0000321	.0000318	1.0
acres	-	s	ss|s	.0000342	.0000318	1.0

$$= \underset{w \in V}{\operatorname{argmax}} P(x \mid w) P(w)$$

El modelo de lenguaje importa!

"a stellar and versatile **acress** whose combination of sass and glamour…"

- ¿Qué pasa si usamos un modelo de bigramas?
  - Modelo de lenguaje construido con el corpus de inglés americano contemporáneo con suavizado Laplace.

P(actress|versatile)=.000021 P(whose|actress) = .0010
P(across|versatile) =.000021 P(whose|across) = .000006

- A veces ponderan los bigramas anterior y posterior, en cuyo caso:

P("versatile actress whose") = .000021*.0010 = 210 x10^{-10}
P("versatile across whose") = .000021*.000006 = 1 x10^{-10}

Caso 2: Errores ortográficos de PALABRA-REAL

Caso 2: Modelo de lenguaje de palabra real

- The design ***an*** construction of the system…
- `The study was conducted mainly` ***`be`*** `John Black.`
- La ***caza*** de Samuel se incendio.

- 25-40% de los errores ortográficos son palabras reales Kukich 1992

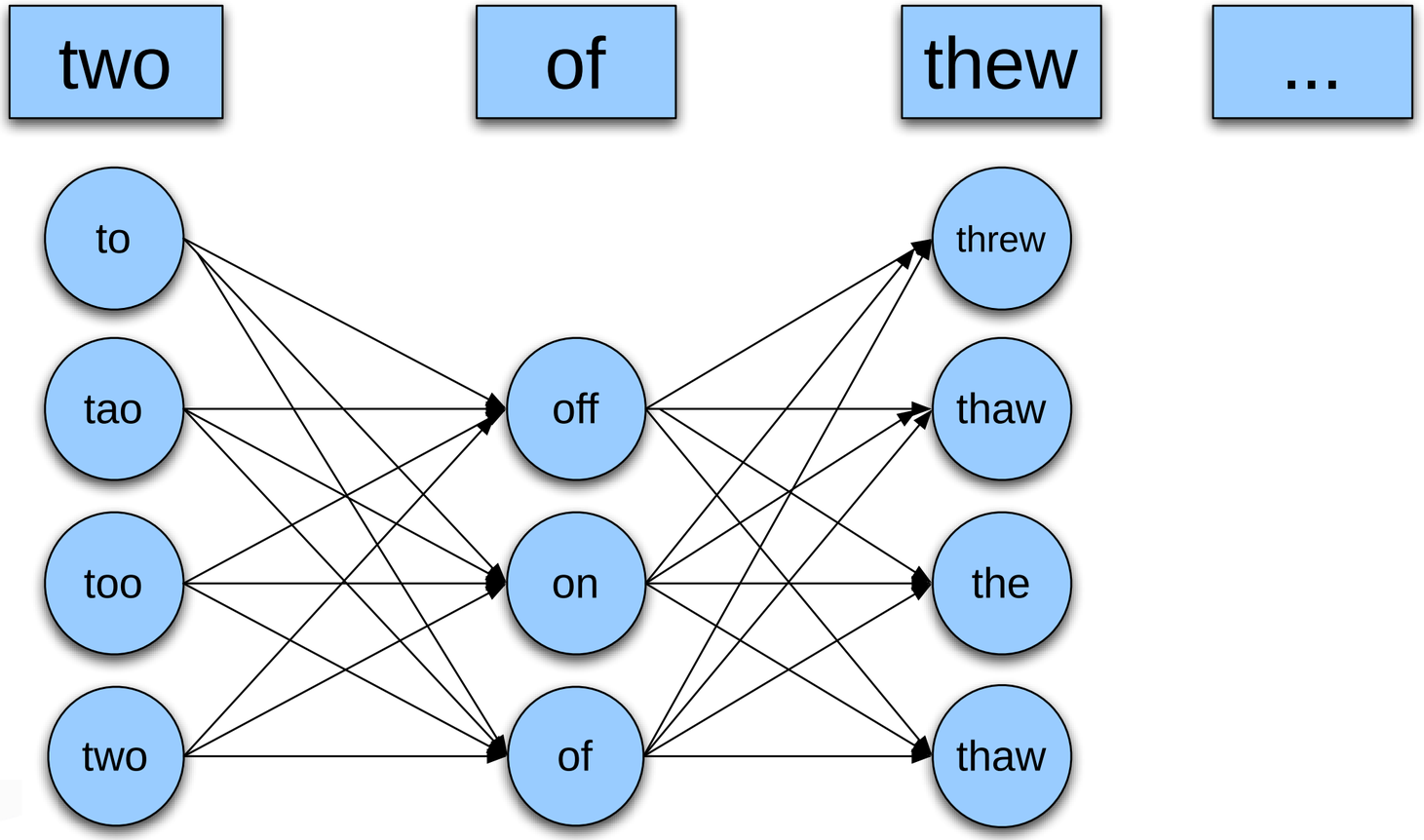
Modelo de canal ruidoso

- Dada una sentencia $w = w_1, w_2, w_3, \ldots, w_n$ de palabras (no se confundan con la notación para el caso 1)
- Genere para cada palabra de la sentencia un conjunto de candidatos.

Candidatos(w_1) = {w_1, w'_1 , w''_1 , w'''_1 ,…}
Candidatos(w_2) = {w_2, w'_2 , w''_2 , w'''_2 ,…}
Candidatos(w_n) = {w_n, w'_n , w''_n , w'''_n ,…}

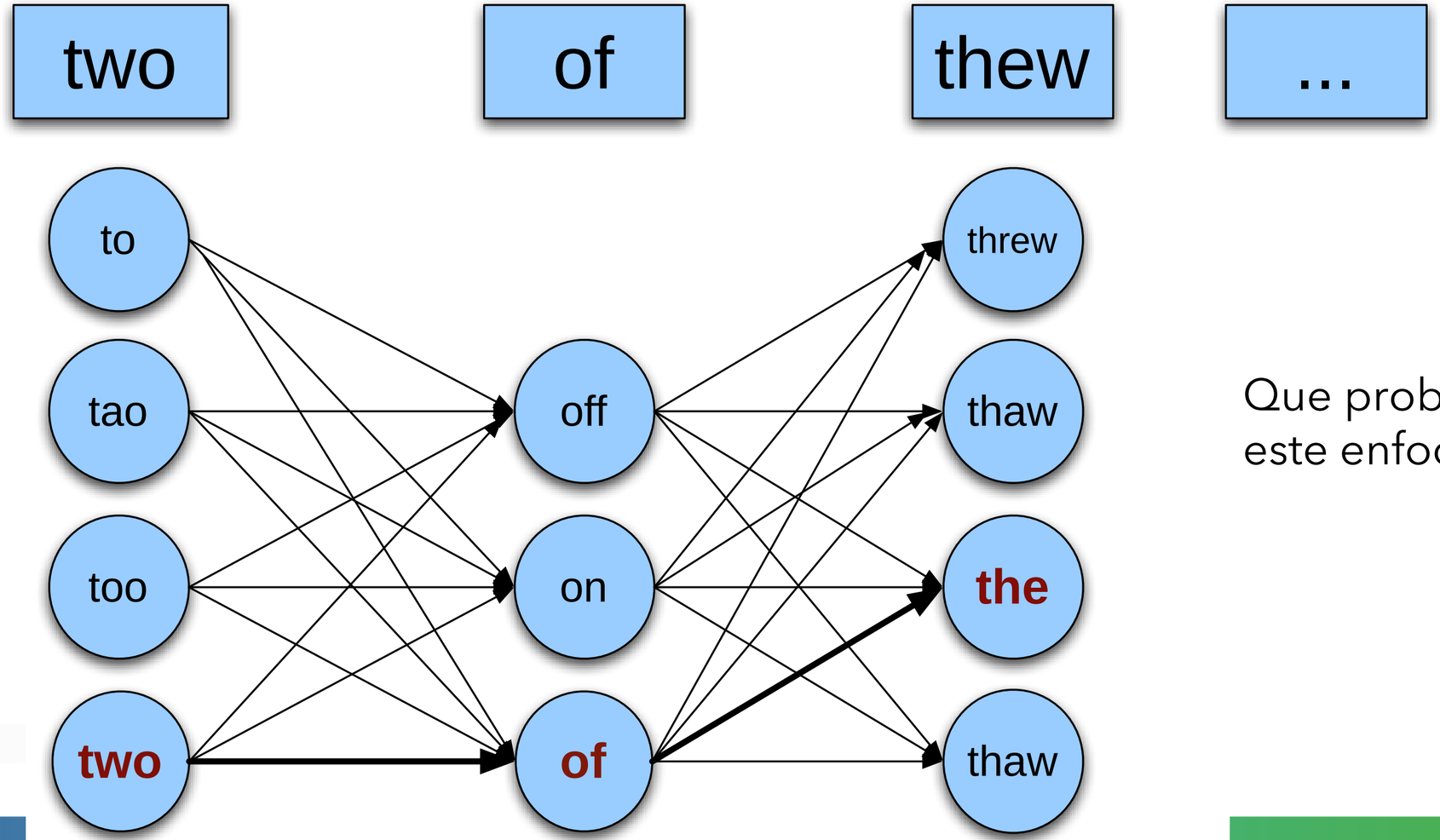
Modelo de canal ruidoso

- Escoja la secuencia W que maximiza $P(W)$.



Modelo de canal ruidoso

- Escoja la secuencia W que maximiza $P(W)$.



Que problema le ven a este enfoque?

Modelo de canal ruidoso. Simplificación un error por sentencia.

- De todos los posibles paths solo seleccionar aquellos en los que solo una palabra de la sentencia original es reemplazada.

w_1, $\boldsymbol{w''_2}$,w_3,w_4	two ***off*** thew
w_1,w_2,$\boldsymbol{w'_3}$,w_4	two of ***the***
$\boldsymbol{w'''_1}$,w_2,w_3,w_4	***too*** of thew

…

- Escoja la secuencia W que maximiza P(W)

De donde tomamos las probabilidades

- Modelo de lenguaje.

- Modelo de canal ruidos.
  - Igual que para el caso 1.
  - Lo único que cambia es que ahora necesitamos la probabilidad de no error P(x|x)

Probabilidad de no error

- Cual es la probabilidad de canal para una palabra correcta?
  - P(“the”|”the”)
  - Es la 1-la probabilidad de error de la aplicación.

- Depende de la aplicación.
  - .90 (1 error en 10 palabras)
  - 0.95 (1 error en 20 palabras)
  - 0.995 (1 error en 200 palabras)

Ejemplo “thew”

x	w	x|w	P(x|w)	P(w)	10^9 P(x|w)P(w)
thew	the	ew|e	0.000007	0.02	144
thew	thew		0.95	0.00000009	90
thew	thaw	e|a	0.001	0.0000007	0.7
thew	threw	h|hr	0.000008	0.000004	0.03
thew	thwe	ew|we	0.000003	0.00000004	0.0001

Referencias

- Jurafsky D. and Martin J. (2021) Speech and Language Processing (3rd ed. draft). Online: https://web.stanford.edu/~jurafsky/slp3/
- (Chapter 2: Speech and Language Processing)

- Yoav Goldberg (2017). Neural Network Methods in Natural Language Processing.

- In Deng, L., & In Liu, Y. (2018). Deep learning in natural language processing.