

Modelado del lenguaje para el reconocimiento de voz

- Introducción
- Modelos de lenguaje tipo n -grama
- Estimación de la probabilidad
- Evaluación
- Más allá de las n -gramas

Modelado del lenguaje para el reconocimiento de voz

- Los reconocedores de voz buscan la secuencia de palabra \hat{W} que tenga mayor probabilidad de ser producida a partir de las pruebas acústicas A

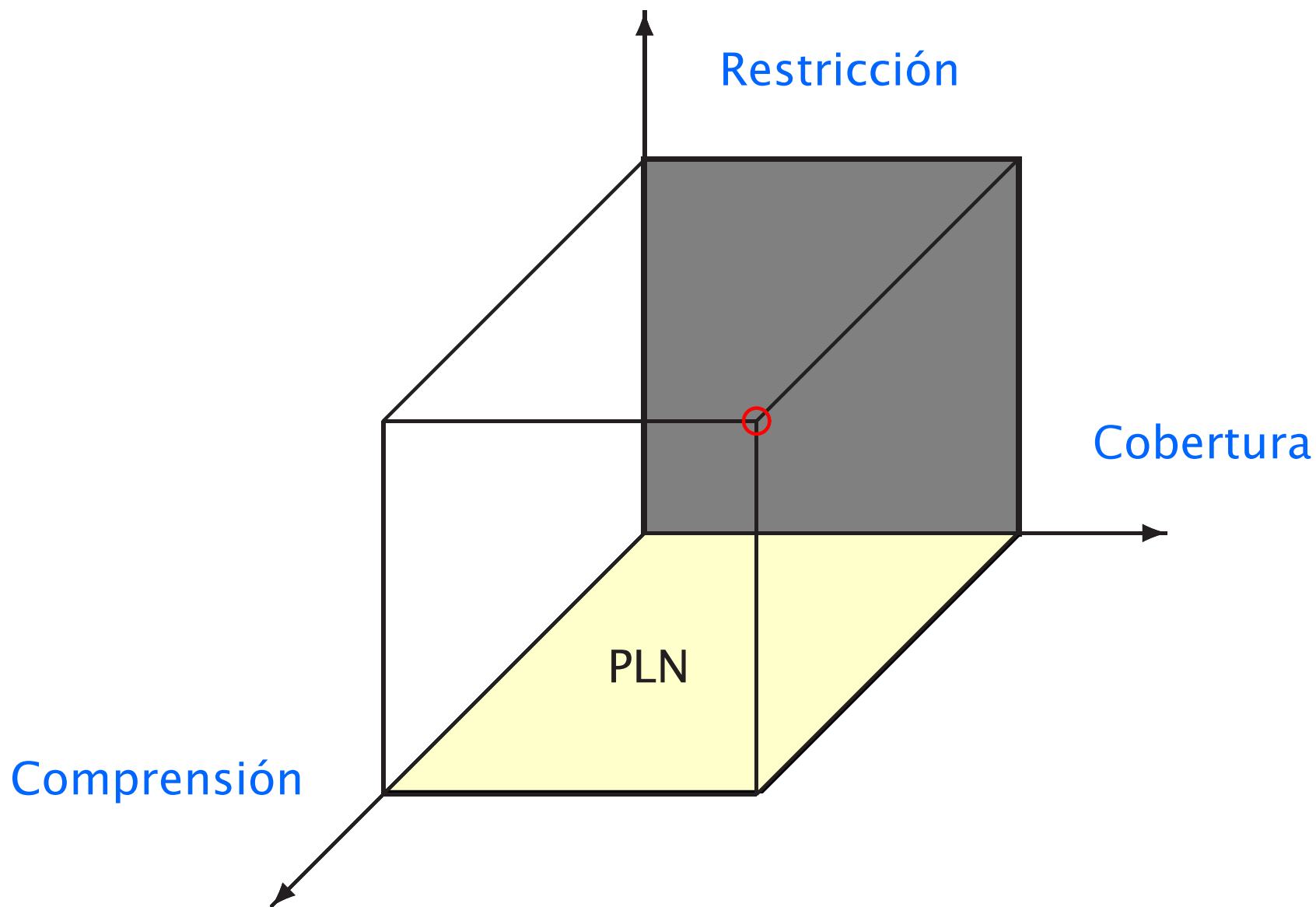
$$P(\hat{W}|A) = \max_W P(W|A) \propto \max_W P(A|W)P(W)$$

- El reconocimiento del habla trata el procesamiento acústico, el modelado acústico, el modelado del lenguaje y la búsqueda.
- Los modelos de lenguaje (LMs) asignan una estimación de probabilidad $P(W)$ a las secuencias de palabra $W = \{w_1, \dots, w_n\}$ sujetas a

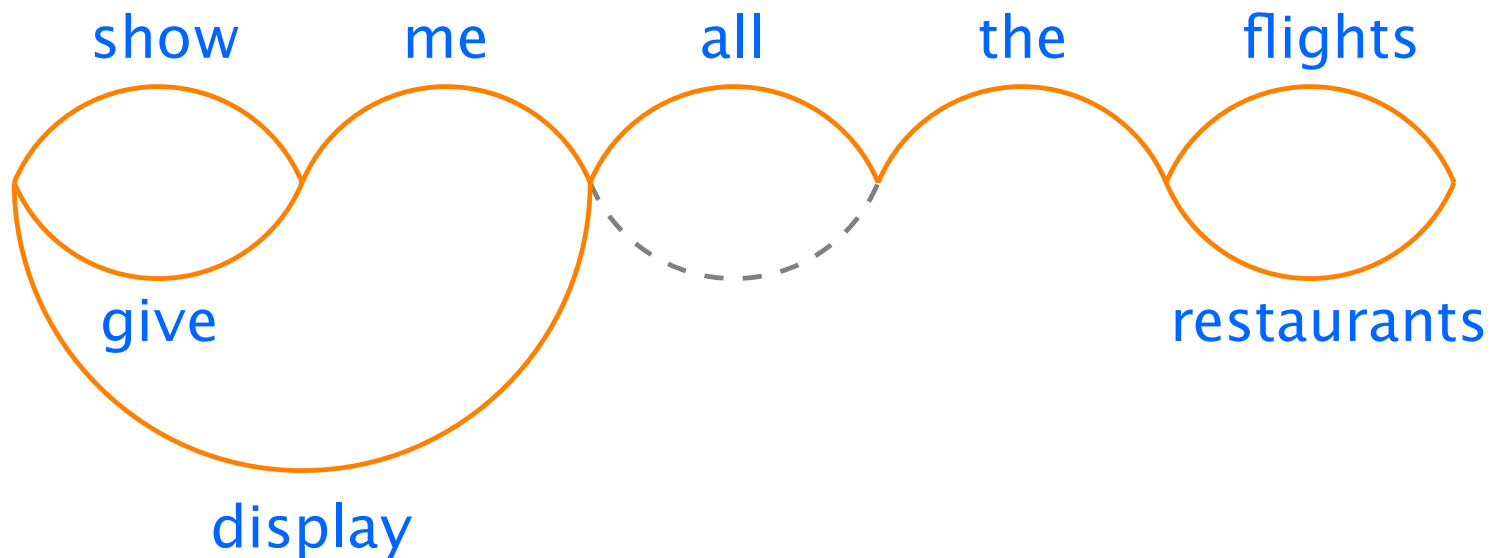
$$\sum_W P(W) = 1$$

- Los modelos de lenguaje facilitan la orientación y restricción de la búsqueda entre las hipótesis de palabras alternativas durante el reconocimiento.

Requisitos del modelo de lenguaje



Redes de estado finito (FSN)

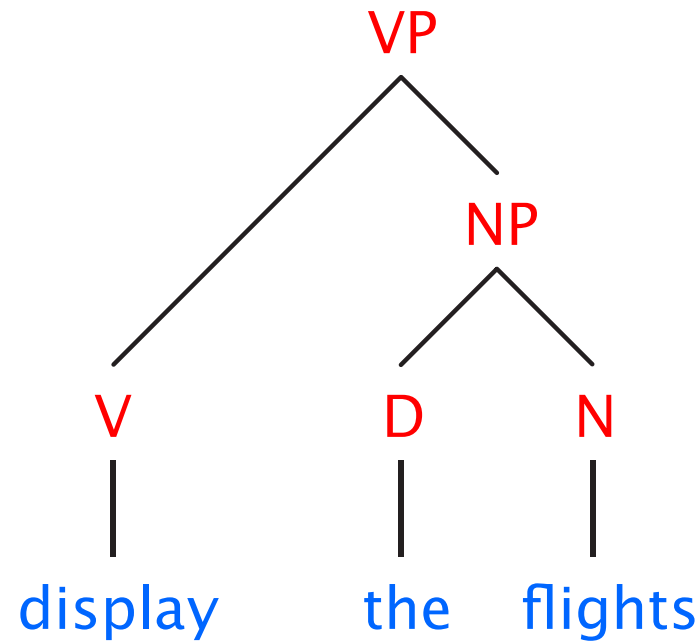


- Espacio del lenguaje definido por una red de palabras o gráfico.
- Descriptible por una gramática de estructura sintagmática **regular**.

$$A \Rightarrow aB \mid a$$

- La cobertura finita puede presentar dificultades para el ASR (Reconocimiento automático del la voz)
- Los arcos del grafo y las reglas se pueden aumentar con probabilidades.

Gramáticas libres de contexto (CFGs)



- Espacio del lenguaje definido por reglas de reescritura libres de contexto.

ej., $A \Rightarrow BC \mid a$

- Representación más potente que en el caso de las redes de estado finito (FSN).
- Las reglas de las gramáticas **CFG estocásticas** han asociado probabilidades que pueden aprenderse automáticamente a partir de un corpus.
- La cobertura finita puede presentar dificultades para el ASR.

Gramáticas de pares de palabras

show → me	me → all → the	the → flights → restaurants
-----------	-------------------	--------------------------------

- Espacio de lenguaje definido por listas de parejas de palabras legales.
- Pueden implementarse de forma eficaz dentro de la búsqueda de Viterbi.
- La cobertura finita puede presentar problemas para el ASR.
- Los **bigramas** definen probabilidades para todas las parejas de palabras y pueden producir un elemento **nulo** $P(W)$ para **todas** las oraciones posibles.

Ejemplo del impacto del modelo de lenguaje (LM) (Lee, 1988)

- Dominio de gestión de recursos
- Independiente del hablante, corpus en discurso continuo.
- Oraciones generadas a partir de una red de estados finitos.
- Vocabulario de 997 palabras.
- Perplejidad de pares de palabras ~ 60 , bigrama ~ 20
- El error incluye sustituciones, eliminaciones e inserciones.

	sin LM	Pares de palabra	Bigrama
% Tasa de error por palabra	29.4	6.3	4.2

Formulación del modelo de lenguaje (LM) para el ASR

- Las probabilidades del modelo de lenguaje $P(W)$ se incorporan normalmente a la búsqueda de ASR lo más pronto posible.
- Dado que la mayoría de las búsquedas se realizan unidireccionalmente, $P(W)$ se formula normalmente como una regla de la cadena

$$P(W) = \prod_{i=1}^n P(w_i | \langle \rangle, \dots, w_{i-1}) = \prod_{i=1}^n P(w_i | h_i)$$

donde $h_i = \{\langle \rangle, \dots, w_{i-1}\}$ es la historia de la palabra para w_i

- h_i se reduce normalmente a las clases de equivalencia $\phi(h_i)$

$$P(w_i | h_i) \approx P(w_i | \phi(h_i))$$

Las clases de buena equivalencia maximizan la información sobre la próxima palabra w_i dada su historia $\phi(h_i)$

- Los modelos de lenguaje que requieren la secuencia completa de palabras W se emplean habitualmente como filtros del posprocesado.

Modelos de lenguaje tipo n -grama

- Los modelos de lenguaje tipo n -grama, utilizan las palabras previas $n - 1$ para representar la historia $\phi(h_i) = \{w_{i-1}, \dots, w_{i-(n-1)}\}$

- Las probabilidades están basadas en frecuencias y cálculos.

$$\text{ej., } f(w_3 | w_1 w_2) = \frac{c(w_1 w_2 w_3)}{c(w_1 w_2)}$$

- Debido a escasos problemas de datos, las n -gramas se suavizan normalmente con frecuencias de menor orden sujetas a

$$\sum_w P(w | \phi(h_i)) = 1$$

- Las bigramas se incorporan fácilmente a la búsqueda de Viterbi.
- Trigramas utilizados en el reconocimiento de extensos vocabularios a mitad de los años 70 y permanecen como el modelo de lenguaje predominante.

Ejemplo de trigrama de IBM (Jelinek, 1997)

1	The	are	to	know	the	issues	necessary
2	This	will		have	this	problems	data
3	One	the		understand	these	the	information
4	Two	would		do	problems		above
5	A	also		get	any		other
6	Three	do		the	a		time
7	Please	need		use	problem		people
8	In			provide	them		operators
9	We			insert	all		tools
96				write			jobs
97				me			MVS
98				resolve			old
1639							reception
1640							shop
1641							important

Ejemplo de trigramma de IBM (continuación)

1	role	and	the	next	be	metting	of
2	thing	from			two	months	<>
3	that	in				years	
4	to	to				meetings	
5	contact	are				to	
6	parts	with				weeks	
7	point	were				days	
8	for	requiring					
9	issues	still					
	•	•					
	•	•					
61		being					
62		during					
63		I					
64		involved					
65		would					
66		within					

Cuestiones sobre las n -gramas: Datos escasos (Jelinek, 1985)

- Corpus en texto de las descripciones patentes de IBM.
- 1 millón y medio de palabras para el entrenamiento.
- 300.000 palabras empleadas para probar los modelos.
- Vocabulario restringido a las 1.000 palabras más frecuentes.
- El 23% de las gramáticas presentes en el corpus de prueba, **faltaban** en el corpus de entrenamiento.
- En general, un vocabulario de tamaño V presentará V^n n -gramas (ej., 20.000 palabras tendrán 400 millones de bigramas, y ocho billones de trigramas).

Interpolación de n -gramas

- Las probabilidades son una combinación lineal de frecuencias.

$$P(w_i|h_i) = \sum_j \lambda_j f(w_i|\phi_j(h_i)) \quad \sum_j \lambda_j = 1$$

ej.,
$$P(w_2|w_1) = \lambda_2 f(w_2|w_1) + \lambda_1 f(w_2) + \lambda_0 \frac{1}{V}$$

- λ calculada con el algoritmo de EM sobre datos presentados.

- Pueden utilizarse distintas λ para diversas historias h

- Se puede utilizar la formulación simplística de λ' $\lambda = \frac{c(w_1)}{c(w_1) + k}$

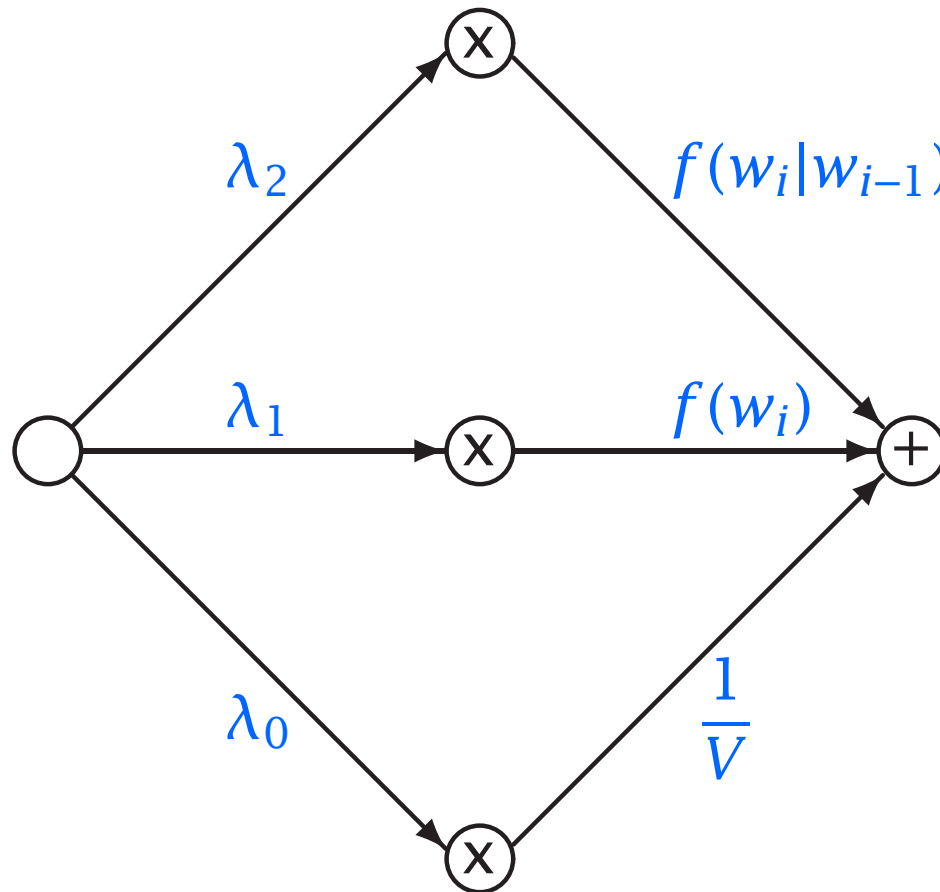
- Las estimaciones se pueden resolver recursivamente:

$$P(w_3|w_1 w_2) = \lambda_3 f(w_3|w_1 w_2) + (1 - \lambda_3)P(w_3|w_2)$$

$$P(w_3|w_2) = \lambda_2 f(w_3|w_2) + (1 - \lambda_2)P(w_3)$$

Ejemplo de interpolación

$$P(w_i|w_{i-1}) = \lambda_2 f(w_i|w_{i-1}) + \lambda_1 f(w_i) + \lambda_0 \frac{1}{V}$$



Interpolación eliminada

1. Inicializar λ (ej., distribución uniforme)
2. Calcular la probabilidad $P(j|w_i)$ de que la estimación de la frecuencia j th se use cuando la palabra w_i sea generada.

$$P(j|w_i) = \frac{\lambda_j f(w_i|\phi_j(h_i))}{P(w_i|h_i)} \quad P(w_i|h_i) = \sum_j \lambda_j f(w_i|\phi_j(h_i))$$

3. Recalcular λ para n_i palabras en los datos presentados.

$$\lambda_j = \frac{1}{n_i} \sum_i P(j|w_i)$$

4. Repetir hasta que se produzca la convergencia.

N -gramas de retroceso (Katz, 1987)

- Las estimaciones del modelo de lenguaje (LM) se emplean cuando los cálculos son grandes.
- Las estimaciones de cálculos bajos se reducen (descuentan) para facilitar la masa de probabilidad para las secuencias no ocultas.
- Las estimaciones de cómputo están basadas en la grama de pesos $(n - 1)$
- El descuento está basado normalmente en la estimación de la fórmula de Turing-Good.

$$P(w_2|w_1) = \begin{cases} f(w_2|w_1) & c(w_1 w_2) \geq \alpha \\ f_d(w_2|w_1) & \alpha > c(w_1 w_2) > 0 \\ q(w_1)P(w_2) & c(w_1 w_2) = 0 \end{cases}$$

- El factor $q(w_1)$ elegido tal que $\sum_{w_2} P(w_2|w_1) = 1$
- Las n -gramas de orden superior se computan recursivamente.

Estimación de Turing-Good

- Probabilidad de que una palabra se de r veces de N , dado θ

$$p_N(r|\theta) = \binom{N}{r} \theta^r (1 - \theta)^{N-r}$$

- Probabilidad de que una palabra se de $r + 1$ veces de $N + 1$

$$p_{N+1}(r + 1|\theta) = \frac{N + 1}{r + 1} \theta p_N(r|\theta)$$

- Suponga que n_r palabras que aparezcan r veces poseen el mismo valor de θ

$$p_N(r|\theta) \approx \frac{n_r}{N} \quad p_{N+1}(r + 1|\theta) \approx \frac{n_{r+1}}{N}$$

- Asumiendo un N grande, podemos resolver θ o r^* **descontada**

$$\theta = P_r = \frac{r^*}{N} \quad r^* = (r + 1) \frac{n_{r+1}}{n_r}$$

Ejemplo de Turing-Good (Church y Gale, 1991)

- La estimación de GT (Turing-Good) para un elemento que se da r veces de N es

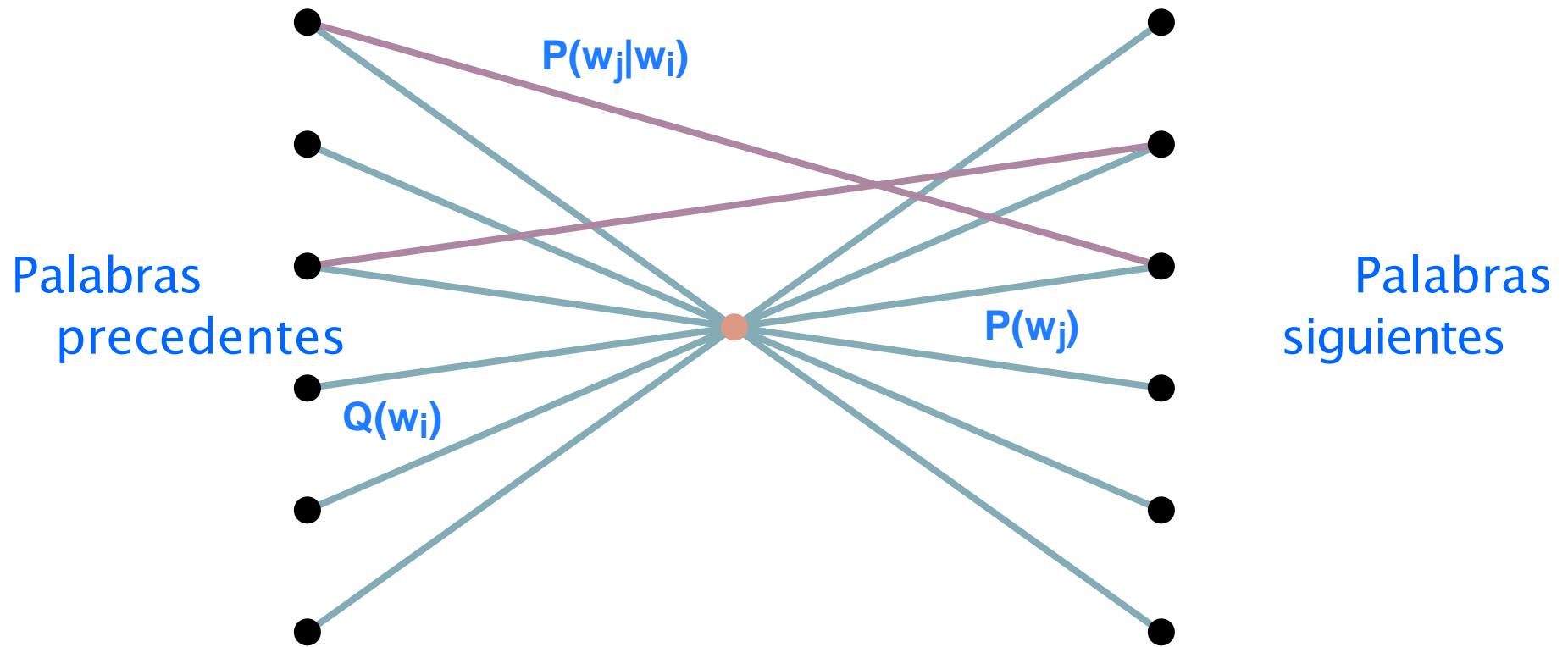
$$P_r = \frac{r^*}{N} \quad r^* = (r + 1) \frac{n_{r+1}}{n_r}$$

donde n_r es el número de elementos que se da r veces

- Considere los cálculos del bigrama de un corpus de 22 millones de palabras extraídas de artículos de noticias AP (vocabulario de 273,000 palabras)

r	n_r	r^*
0	74,671,100,000	0.0000270
1	2,018,046	0.446
2	449,721	1.26
3	188,933	2.24
4	105,668	3.24
5	68,379	4.22

Integración en la búsqueda de Viterbi



Las bigramas se pueden incorporar eficientemente en la búsqueda de Viterbi utilizando un nodo intermedio entre palabras.

- Interpolado: $Q(w_i) = (1 - \lambda_i) P(w_i|w_{i-1}) + \lambda_i P(w_i)$
- Retroceso: $Q(w_i) = q(w_i)$

Evaluación de los modelos de lenguaje

- Exactitud en el reconocimiento
- Valoración cualitativa
 - Generación aleatoria de oraciones
 - Reorganización oracional
- Medidas de información teóricas

Generación aleatoria de oraciones: Bigrama de un entorno de vuelos aéreos

Show me the flight earliest flight from Denver
How many flights that flight leaves around is the Eastern Denver
I want a first class
Show me a reservation the last flight from Baltimore for the first
I would like to fly from Dallas
I get from Pittsburgh
Which just small
In Denver on October
I would like to San Francisco
Is flight flying
What flights from Boston to San Francisco
How long can you book a hundred dollars
I would like to Denver to Boston and Boston
Make ground transportation is the cheapest
Are the next week on AA eleven ten
First class
How many airlines from Boston on May thirtieth
What is the city of three PM
What about twelve and Baltimore

Generación aleatoria de oraciones: Trigrama de un entorno de vuelos aéreos

What type of aircraft

What is the fare on flight two seventy two

Show me the flights I've Boston to San Francisco on Monday

What is the cheapest one way

Okay on flight number seven thirty six

What airline leaves earliest

Which airlines from Philadelphia to Dallas

I'd like to leave at nine eight

What airline

How much does it cost

How many stops does Delta flight five eleven o'clock PM that go from

What AM

Is Eastern from Denver before noon

Earliest flight from Dallas

I need to Philadelphia

Describe to Baltimore on Wednesday from Boston

I'd like to depart before five o'clock PM

Which flights do these flights leave after four PM and lunch and <unknown>

Reorganización oracional (Jelinek, 1991)

- Palabras desordenadas de una oración.
- Hallar el orden más probable con el modelo de lenguaje.
- Resultados con el LM (modelo de lenguaje) del trigramma.
 - Oraciones cortas a partir del dictado espontáneo.
 - El 63% de la oraciones reorganizadas son idénticas.
 - El 86% presentan el mismo significado.

Reorganización oracional de IBM

would I report directly to you

I would report directly to you

now let me mention some of the disadvantages

let me mention some of the disadvantages now

he did this several hours later

this he did several hours later

this is of course of interest to IBM

of course this is of interest to IBM

approximately seven years I have known John

I have known John approximately seven years

these people have a fairly large rate of turnover

of these people have a fairly large turnover rate

in our organization research has two missions

in our missions research organization has two

exactly how this might be done is not clear

clear is not exactly how this might be done

Cuantificación de la complejidad del modelo de lenguaje (LM)

- Un LM es mejor que otro si puede pronosticar un corpus W de **prueba** de n palabras con una probabilidad más alta $P(W)$
- Para los LM representables por la regla de la cadena, las comparaciones están normalmente basadas en el promedio de probabilidad logarítmica (LP) por palabra

$$LP = -\frac{1}{n} \log_2 \hat{P}(W) = -\frac{1}{n} \sum_i \log_2 \hat{P}(w_i | \phi(h_i))$$

- Una representación más intuitiva de LP es la **perplejidad**

$$PP = 2^{LP}$$

(un modelo de lenguaje tendrá una PP equivalente al tamaño del vocabulario)

- La PP se interpreta con frecuencia como un factor medio de ramificación

Ejemplos de perplejidad

Entorno	Tamaño	Tipo	Perplejidad
Dígitos	11	Todas las palabras	11
Gestión de recursos	1,000	Bigrama de pares de palabras	60 20
Comprensión en el dominio de vuelos aéreos	2,500	Bigrama de 4 gramas	29 22
Dictado WSJ	5,000	Bigrama	80
		Trigrama	45
	20,000	Bigrama	190
		Trigrama	120
Centralita telefónica con operador humano	23,000	Bigrama	109
		Trigrama	93
Caracteres NYT	63	Unigrama	20
		Bigrama	11
Cartas Shannon	27	Humano	~ 2

Entropía del lenguaje

- El promedio de probabilidad logarítmica LP está relacionado con lo más incierto del lenguaje, cuantificado por su **entropía**

$$H = -\lim_{n \rightarrow \infty} \frac{1}{n} \sum_W P(W) \log_2 P(W)$$

- Si W se obtiene a partir de una fuente bien comportada (ergódica), $P(W)$ convergerá en el valor esperado y H será

$$H = -\lim_{n \rightarrow \infty} \frac{1}{n} \log_2 P(W) \approx -\frac{1}{n} \log_2 P(W) \quad n \gg 1$$

- La entropía H es un límite teórico menor sobre la probabilidad logarítmica LP

$$-\lim_{n \rightarrow \infty} \frac{1}{n} \sum_W P(W) \log_2 P(W) \leq -\lim_{n \rightarrow \infty} \frac{1}{n} \sum_W P(W) \log_2 \hat{P}(W)$$

Entropía del lenguaje humano (Shannon, 1951)

- Un intento de estimar la entropía del lenguaje de los humanos
- Implicaba la averiguación de las palabras próximas con el fin de medir la distribución de probabilidad de los sujetos
- Las cartas se utilizaban para simplificar los experimentos

T	H	E	R	E	I	S	N	O	R	E	V	E	R	S	E			
1	1	1	5	1	1	2	1	1	2	1	1	15	1	17	1	1	1	2
O	N	A	M	O	T	O	R	C	Y	C	L	E	A	...				
1	3	2	1	2	2	7	1	1	1	1	4	1	1	1	1	1	3	...

- $\hat{H} = -\sum \hat{P}(i) \log_2 \hat{P}(i)$ $\hat{P}(1) = \frac{24}{37}$ $\hat{P}(2) = \frac{6}{37}$ $\hat{P}(3) = \frac{2}{37}$
- Shannon estimó $H \approx \hat{H}$ bit por carta

¿Por qué funcionan tan bien las n-gramas?

- Probabilidades basadas en datos (cuantos más, mejor)
- Parámetros definidos automáticamente a partir de corpus
- Incorporación de sintaxis, semántica y pragmática
- Muchos lenguajes tienen una fuerte tendencia hacia el orden de palabras estándar y por tanto, son sustancialmente locales
- Relativamente fácil de integrar en métodos de búsqueda de avance como el de Viterbi (bigrama) o A^*

Problemas de las n-gramas

- Incapaces de incorporar restricciones a larga distancia
- No muy apropiadas para los lenguajes flexibles en cuanto al orden de palabras
- No pueden acomodar fácilmente
 - Nuevos elementos de vocabulario
 - Entornos alternativos
 - Cambios dinámicos (ej., discurso)
- No tan buenas como los humanos en tareas basadas en:
 - La identificación y corrección de errores del reconocedor
 - La predicción de palabras siguientes (o cartas)
- No capturan el significado para la comprensión del discurso.

Agrupamiento de palabras

- Muchas palabras presentan un comportamiento estadístico parecido
 - ej., días de la semana, meses, ciudades, etc.
- El rendimiento de la n-grama se puede mejorar agrupando palabras
 - El agrupamiento duro coloca una palabra en un único grupo
 - El agrupamiento suave permite que una palabra pertenezca a múltiples grupos
- Los grupos pueden crearse manualmente o automáticamente
 - Los grupos creados manualmente funcionan bien en entornos pequeños
 - Los grupos automáticos han sido creados de forma ascendente o descendente

Agrupamiento de palabras ascendente (Brown et al., 1992)

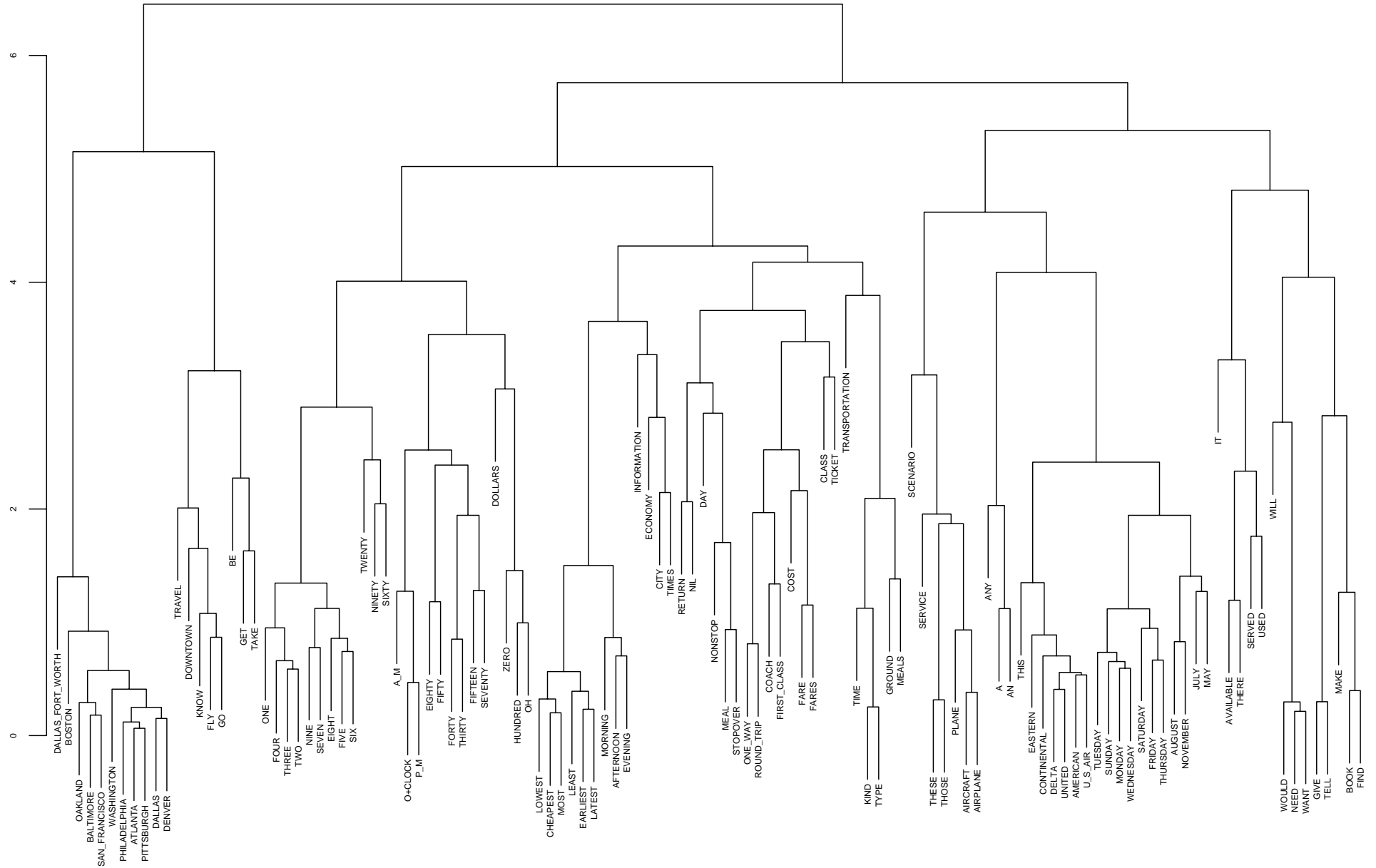
- Los grupos de palabras se pueden generar automáticamente mediante la creación de grupos de un modo óptimo paso a paso o siguiendo el estilo de avance rápido
- Grupos ascendentes creados teniendo en cuenta el impacto sobre la métrica de las palabras de fusión w_a y w_b para formar un nuevo grupo w_{ab}
- Métrica de ejemplo para un modelo de lenguaje bigrama:
 - Descenso mínimo en el promedio de información común

$$I = \sum_{i,j} P(w_i w_j) \log_2 \frac{P(w_j | w_i)}{P(w_j)}$$

- Aumento mínimo en la entropía condicional del grupo de entrenamiento. ?

$$H = - \sum_{i,j} P(w_i w_j) \log_2 P(w_j | w_i)$$

Ejemplo de agrupamiento de palabras



Modelos n -gramas de clase de palabra

- Las n -gramas de clase de palabras agrupan las palabras en clases de equivalencia

$$W = \{w_1, \dots, w_n\} \rightarrow \{c_1, \dots, c_n\}$$

- Si los grupos no se solapan, $P(W)$ queda aproximado por

$$P(W) \approx \prod_{i=1}^n P(w_i | c_i) P(c_i | \langle \rangle, \dots, c_{i-1})$$

- Menos parámetros que las n -gramas de palabras
- Relativamente fácil de añadir nuevas palabras a los grupos existentes
- Si se desea, se pueden combinar linealmente n -grams de palabras

Agrupamiento predictivo (Goodman, 2000)

- Para n -gramas de clase de palabra: $P(w_i|h_i) \approx P(w_i|c_i)P(c_i|c_{i-1} \dots)$
- El agrupamiento predictivo es exacto: $P(w_i|h_i) = P(w_i|h_i c_i)P(c_i|h_i)$
- La historia h_i puede agruparse de forma distinta para los dos términos
- Este modelo puede ser más extenso que el de n -grama, pero se ha demostrado que produce buenos resultados cuando se combina con el de reducción

N-gramas de clase sintagma (PCNG) (McCandless, 1994)

- Las reglas probabilísticas libres de contexto analizan sintácticamente los sintagmas

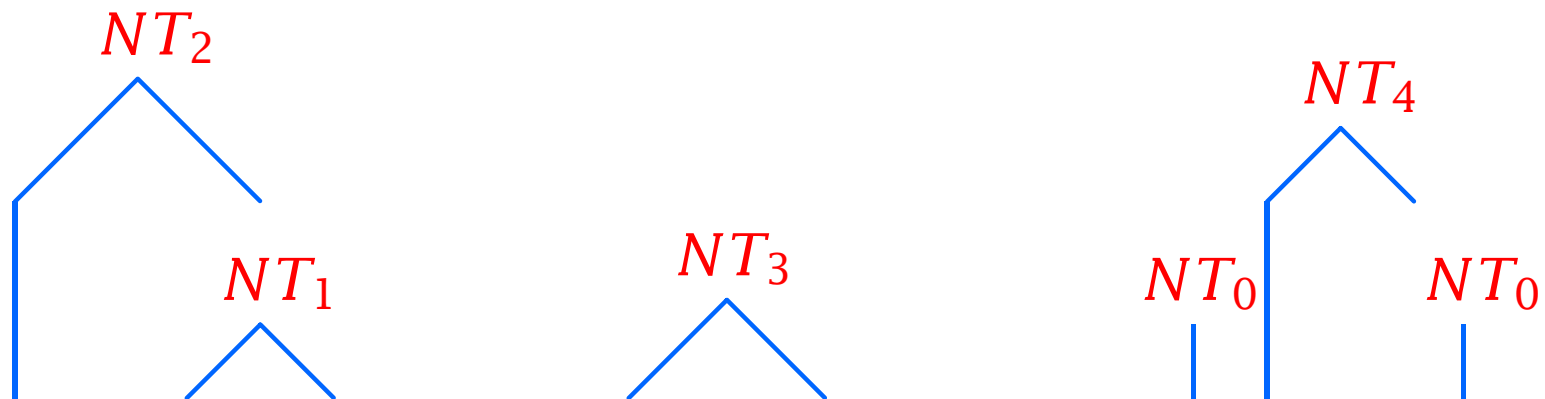
$$W = \{w_1, \dots, w_n\} \rightarrow \{u_1, \dots, u_m\}$$

- La n -grama genera la probabilidad de las unidades resultantes
- $P(W)$ es producto del análisis sintáctico y de las probabilidades de la n -grama

$$P(W) = P_r(W)P_n(U)$$

- Representación intermedia entre las n -gramas de palabras y las gramáticas estocásticas libres de contexto
- Las reglas libres de contexto se pueden aprender automáticamente

Ejemplo de PCNG



Please show me the cheapest flight from Boston to Denver



NT_2 the NT_3 from NT_0 NT_4

Experimentos de PCNG

- Entorno de servicio de información sobre vuelos aéreos (ATIS)
- Comprensión del lenguaje hablado, espontáneo
- 21,000 en entrenamiento, 2,500 en desarrollo, 2,500 en oraciones de prueba
- Vocabulario de 1,956 palabras

Modelo de lenguaje	NºReglas	Nº Parámetros	Perplejidad
Bigrama de palabras	0	18430	21.87
+ Palabras compuestas	654	20539	20.23
+ Clases de palabras	1440	16430	19.93
+ Sintagmas	2165	16739	15.87
Trigrama PCNG	2165	38232	14.53
4-grama PCNG	2165	51012	14.40

- Clases de equivalencia representadas en un árbol de decisión
 - Los nodos de la rama contienen preguntas para la historia h_i
 - Los nodos de la hoja poseen clases de equivalencia
- La formulación de la n -grama de palabras se ajusta al modelo de árbol de decisión
- Criterio de mínima entropía empleado para la construcción
- Se precisa una computación significativa para generar los árboles

Modelos de lenguaje exponenciales

- $P(w_i|h_i)$ modelado como el producto de rasgos de pesos $f_j(w_i|h_i)$

$$P(w_i|h_i) = \frac{1}{Z(h_i)} e^{\sum_j \lambda_j f_j(w_i|h_i)}$$

donde las λ son parámetros, y

$Z(h_i)$ es un factor de normalización

- Los rasgos de valores binarios pueden expresar relaciones arbitrarias

$$\text{ej., } f_j(w_i|h_i) = \begin{cases} 1 & w_i = A \ \& \ w_{i-1} = B \\ 0 & \text{else} \end{cases}$$

- Cuando $E(f(wh))$ equivale al valor empírico esperado, las estimaciones del modelo de lenguaje (ML) para las λ , corresponden a la distribución de máxima entropía
- Las soluciones del ML son iterativas, y pueden ser sumamente lentas
- La perplejidad demostrada y la tasa de error por palabra (WER) aumenta en grandes tareas de vocabulario

Modelos de lenguaje adaptativos

- Los modelos de lenguaje basados en caché incorporan la estadística de las palabras utilizadas recientemente con un modelo de lenguaje estático

$$P(w_i|h_i) = \lambda P_c(w_i|h_i) + (1 - \lambda)P_s(w_i|h_i)$$

- Los modelos de lenguaje **basados en desencadenamiento** aumentan las probabilidades de la palabra cuando las palabras claves se observan en la historia h_i
 - Los desencadenamientos automáticos facilitan información significativa
 - Métrica de información empleada para hallar desencadenamientos.
 - Incorporados en la formulación de máxima entropía

Ejemplos de desencadenamiento (Lau, 1994)

- Desencadenamientos determinados automáticamente a partir de un corpus WSJ (de 37 millones de palabras) empleando el promedio de información común
- Siete desencadenamientos por palabra utilizados en el modelo de lenguaje

Palabra	Desencadenamiento
stocks	stocks index investors market dow average industrial
political	political party presidential politics election president campaign
foreign	currency dollar japanese domestic exchange japan trade
bonds	bonds bond yield treasury municipal treasury's yields

Reducción del modelo de lenguaje

- Los modelos de lenguaje de n -gramas pueden llegar a ser muy grandes (ej., 6B/ n -grama)
- Existen técnicas sencillas que pueden reducir el tamaño del parámetro
 - Reducir n -gramas con muy pocas ocurrencias
 - Reducir n -gramas que poseen un pequeño impacto en el modelo de entropía
- Ejemplo de reducción de trigrama basado en recuentos.
 - Transcripción de un programa de noticias (ej., TV, programas de radio)
 - 25K de vocabulario; 166M palabras para el entrenamiento ($\sim 1GB$), 25K palabras de prueba

Cómputo	Bigramas	Trigramas	Estados	Arcos	Tamaño	Perplejidad
0	6.4M	35.1M	6.4M	48M	360MB	157.4
1	3.2M	11.4M	2.2M	17M	125MB	169.4
2	2.2M	6.3M	1.2M	10M	72MB	178.1
3	1.7M	4.4M	0.9M	7M	52MB	185.1
4	1.4M	3.4M	0.7M	5M	41MB	191.9

Reducción de entropía (Stolcke, 1998)

- Emplea distancia KL para reducir n -gramas con bajo impacto en entropía

$$D(P \parallel P') = \sum_{i,j} P(w_i|h_j) \log \frac{P(w_i|h_j)}{P'(w_i|h_j)} \quad \frac{PP' - PP}{PP} = e^{D(P \parallel P')} - 1$$

1. Seleccionar el umbral de reducción θ
 2. Computar el aumento de perplejidad desde la reducción de cada n -grama
 3. Eliminar n -gramas bajo θ , y recalcular los pesos de retroceso
- Ejemplo: recurriendo a las N -mejores listas de noticias del programa con 4-gramas

θ	Bigramas	Trigramas	4-gramas	Perplejidad	% WER _(Tasa de error por palabra)
0	11.1M	14.9M	0	172.5	32.9
0	11.1M	14.9M	3.3M	163.0	32.6
10^{-9}	7.8M	9.6M	1.9M	163.9	32.6
10^{-8}	3.2M	3.7M	0.7M	172.3	32.6
10^{-7}	0.8M	0.5M	0.1M	202.3	33.9

Perplejidad frente a tasa de error (Rosenfeld et al., 1995)

- Conversaciones telefónicas desde centralita con operador humano
- 2.1 millones de palabras para entrenamiento, 10,000 palabras para pruebas
- 23,000 palabras de vocabulario, bigrama de perplejidad 109
- Búsqueda del retículo de palabras generado por el bigrama (10% de tasa de error)

Condición del trigramas	Perplejidad	% de tasa de error
Entrenado con conjunto de entrenamiento	92.8	49.5
Entrenado con conjuntos de entrenamiento y de pruebas	30.4	38.7
Entrenado con conjuntos de prueba	17.9	32.9
No se suaviza el parámetro	3.2	31.0
Retículo perfecto	3.2	6.3
Otro retículo	3.2	44.5

- X. Huang, A. Acero y H. -W. Hon, *Spoken Language Processing*, Prentice-Hall, 2001.
- K. Church & W. Gale, A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams, *Computer Speech & Language*, 1991.
- F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, 1997.
- S. Katz, Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. ASSP-35*, 1987.
- K. F. Lee, *The CMU SPHINX System*, Ph.D. Thesis, CMU, 1988.
- R. Rosenfeld, Two Decades of Statistical Language Modeling: Where Do We Go from Here?, *IEEE Proceedings*, 88(8), 2000.
- C. Shannon, Prediction and Entropy of Printed English, *BSTJ*, 1951.

- L. Bahl et al., A Tree-Based Statistical Language Model for Natural Language Speech Recognition, *IEEE Trans. ASSP-37*, 1989.
- P. Brown et al., Class-based n -gram models of natural language, *Computational Linguistics*, 1992.
- R. Lau, Adaptive Statistical Language Modelling, S.M. Thesis, MIT, 1994.
- M. McCandless, Automatic Acquisition of Language Models for Speech Recognition, S.M. Thesis, MIT, 1994.
- R. Rosenfeld et al., Language Modelling for Spontaneous Speech, Johns Hopkins Workshop, 1995.
- A. Stolcke, Entropy-based Pruning of Backoff Language Models, <http://www.nist.gov/speech/publications/darpa98/html/lm20/lm20.htm>, 1998.