Procesamiento de Lenguaje Natural

TEMA 2

Palabras

Enrique Alfonseca
Pilar Rodríguez

Índice

Introducción (I) – Definiciones

Morfología Es el estudio de la estructura dentro de las palabras:

- Mecanismos para crear nuevas palabras.
- Mecanismos para utilizar las palabras.

Morfema Es la unidad más pequeña a la que se le puede asignar significado.

Raíz, Lema Es el morfema que expresa un concepto semántico (puerta).

Alomorfos Son los morfemas que aportan el mismo significado a la palabra (*-i*, *-is* en servi, patris).

Introducción (II) – Definiciones

Morfema libre Es el que puede constituir una palabra *per se* (casa).

Morfema ligado Es el que sólo ocurre en combinación con otros(*-s* para el plural).

- En algunos idiomas (inglés, español, etc.) han de ocurrir asociados por delante o por detrás a la raíz.
- En otros (árabe, hebreo, etc.) se emplean operaciones no concatenativas (por ejemplo, variar vocales).

Introducción (III)
Clasificación de lenguajes

Inflectional languages (Lenguajes flexionales): distintos contenidos de significación se juntan en un solo morfema ligado, que se afija al lema (Lenguas indoeuropeas).

Isolating languages (Lenguajes aislantes): no hay morfemas ligados (Chino mandarín).

Agglutinative languages (Lenguajes aglutinantes): todos los morfemas ligados son afijos que se van engarzando unos con otros para formar la palabra (Finés, Turco).

Polysynthetic languages (Lenguajes polisintéticos): expresan más información estructural de manera morfológica (lenguas Inuit).

Morfología (IV)

Morfología flexional

Determinados contextos sintácticos exigen que la palabra lleve una cierta inflexión determinando su función gramatical.

Todas las formas de una palabra se llaman el *paradigma*:

	Singular	Plural
Nominativo	servus	servi
Vocativo	serve	servi
Acusativo	servum	servos
Genitivo	servi	servorum
Dativo	servo	servis
Ablativo	servo	servis

Morfología (V)
Morfología flexional

- *Funcional*: un *afijo* actúa como una función (p. ej., el sufijo *-s* que forma el plural).
- Preserva la categoría sintáctica de la palabra.
- *Completa*: Con raras excepciones, todas las palabras tienen todas las formas de su paradigma (llover).
- *Productiva*: nuevas palabras automáticamente utilizan las reglas de inflexión.

Según la inflexión, las palabras se clasifican como:

- **Partículas** o palabras sin inflexión (preposiciones, conjunciones, adverbios...)
- **Verbos**, que siguen una conjugación.
- **Nominales**, o palabras que siguen declinaciones (nombres, adjetivos y pronombres).

Morfología (VI)

Morfología derivacional

Derivación es un proceso mediante el cual se crean palabras nuevas.

Se realiza uniendo un morfema ligado a una forma base:

comer	comest-ible	concebir	conceb-ible
eat	eat-able	conceive	conceiv-able
essen	ess-bar	absehen	abseh-bar

Morfología (VII)
Morfología derivacional

- *Relacional*: el mismo sufijo puede tener resultados diferentes (criticise vs. localise).
- No necesariamente preserva la categoría sintáctica.
- *Incompleta*: no todas las palabras pueden acomodar el mismo conjunto de afijos. Por ejemplo, -ible no es aplicable a todos los verbos). En inglés, -ity sólo se aplica a palabras de origen latino (rarity, gravity, *reddity, *weirdity).
- Recursiva:
  hospital → hospitalizar → hospitalización → pseudohospitalización

Morfología (VIII)
Afijación

Un **afijo** es un morfema ligado que consta de un conjunto de fonemas.

Prefijación:
Se da cuando se añade un afijo (prefijo) delante del lema.

gramatical → a-gramatical

Sufijación:
Se da cuando se añade un afijo (sufijo) detrás del lema:

gramática → gramatic-al

Morfología (IX)
Afijación

Circunfijación:
Es la combinación de la prefijación y la sufijación que conjuntamente expresan una característica:

sagen *(decir)* → ge-sag-t *(dicho)* (Alemán)

Infijación:
Se da cuando la posición del afijo depende de alguna condición fonológica, por lo que puede aparecer dentro del lema.

fikas *(fuerte)* → fumikas *(ser fuerte)* (Bontoc, Filipinas)

Morfología (X)
Afijación

Reduplicación:

Se da cuando se copia parte o todo el lema, posiblemente con variaciones fonéticas.

- En Javanés, expresa el repetitivo habitual (soler):

bali	regresar	bolabali	regresar a menudo
dolan	recrear	dolandolɛn	recrear a menudo
adus	bañarse	odasadus	bañarse a menudo

- En Yidin (Australia), expresa plural:

mulari	persona iniciada	mulamulari
gindalba	lagarto	gindalgindalba

- En Amharic (Etiopía) expresa el frecuentativo.

Morfología (XI)
Fenómenos no aglutinantes

Ablaut:
Fenómeno heredado del protoindoeuropeo, conlleva la modificación de alguna vocal del lema como proceso morfológico.
mann (*hombre*), Anglosajón:

	Singular	Plural
Nominativo	mann	menn
Acusativo	mann	menn
Genitivo	mannes	manna
Dativo	menn	mannum

Morfología (XII)

Fenómenos no aglutinantes

Umlaut:

Consiste en que alguna vocal del lema se convierte en la vocal frontal equivalente:

Alemán:

Singular	Plural
Mutter	Mütter
Garten	Gärten

Morfología (XIII)

Fenómenos no aglutinantes

Morfología lema-"plantilla":

Se da en lenguajes semíticos.

La raíz consiste en de dos a cuatro consonantes, y las vocales indican las caraterísticas morfológicas:

Árabe *ptr* (escribir):

Activa	**Pasiva**	**Verbo**
katab	kutib	escribir
kattab	kuttib	causar que escriba
ka:tab	ku:tib	mantener correspondencia
taka:tab	tuku:tib	escribirse mútuamente
nka:tab	nku:tib	suscribir
staktab	stuktib	dictar

Morfología (XIV)
Idiomas aglutinativos

Los afijos se van engarzando unos a otros alrededor del lema.

Kivunjo (lengua Bantú) **Näïkìḿlyìïà** (Éste comió eso para beneficio de aquel):

N Marcador indicando que la palabra es el tema de la conversación.
ä Marcador de concordancia con el sujeto (humano singular), de entre los 16 géneros.
ï Tiempo presente (otros tiempos son hoy, hoy antes de ahora, ayer, no antes de ayer, en el pasado remoto, habitualmente, etc.)
kì Marcador de concordancia con el complemento directo (género clase 7)
ḿ Marcador de concordancia con el beneficiado por la acción.
lyì El verbo, *comer*.
ï Marcador "aplicativo", indica que hay un actor más en la acción (en este caso, el beneficiado).
à Modo indicativo.

Morfología Computacional (I)
Motivación

- **Análisis**: obtención de la estructura interna de las palabras (lema y afijos)
  - Para comprobar concordancia en análisis sintácticos.
  - Para correctores ortográficos.
  - Para poner guiones al final de las líneas.
  - Para indexar documentos por lemas de las palabras.
  - Para separar palabras en lenguajes sin blancos (chino, japonés...)
- **Generación**: obtención de palabras con inflexión para generación de textos.

Morfología Computacional (II)

Lexicón completo

Consiste en tener un lexicón completo con todas las palabras del idioma en todas sus formas posibles.

- Simple.
- Aplicable a todos los fenónemos posibles (afijación, ablaut, etc.)

- Redundancia.
- Inabilidad de tratar formas que no estén en el lexicón.
- Algunos lenguajes (Kivunjo) pueden tener alrededor de medio millón de formas para cada palabra.

Morfología Computacional (III)
Lexicón de lemas y reglas

Consiste en tener un lexicón con todos los lemas de un lenguaje, y un conjunto de reglas de inflexión.

Problemas:

- Palabras muy comunes no suelen seguir los paradigmas (el verbo ser, verbos irregulares).
- Excepciones (y arcaísmos), como los verbos fuertes en inglés: *give-gave-given*
- Reglas fonológicas que alteran los sufijos y los lemas:

  $$\text{in}+\text{batible} \rightarrow \text{imbatible}$$

- Los algoritmos de análisis y de generación son totalmente diferentes.
- Los algoritmos son muy específicos de cada idioma.

Morfología Computacional (IV)
Morfología de estado finito

Muchas reglas morfológicas se pueden expresar con expresiones regulares.

Por tanto, se pueden codificar como transductores finitos deterministas.

Formalismos:

- Morfología a dos niveles.
- Morfología paradigmática.
- Sistema DATR.

Morfología Computacional (V)

Morfología a dos niveles

- Existen dos niveles: el de la palabra tal como se escribe o pronuncia, y el nivel léxico, con diacríticos.
- Los diacríticos # y + son los separadores de palabras y morfemas, respectivamente.
- Un conjunto de reglas indican cómo se alterna entre los dos niveles (sirven tanto para análisis como para generación).

```
#bliss+s#
0blisses0
```

```
+:e ⇐ {s x z [{s c} h]}:_s;
```

(Koskenniemi, 1984)

Índice

- **Análisis morfológico**
  - Morfología
  - Morfología computacional
- **PoS tagging**
  - Introducción
  - Listas de transformación
  - Modelos de Markov
  - Otros

Introducción

Las palabras pueden agruparse en clases en función de su comportamiento sintáctico, llamadas *categorías gramaticales* o *partes del lenguaje*.

Por ejemplo,

- los *nombres* generalmente designan personas, lugares, cosas, y otros conceptos físicos y abstractos,
- los *verbos* suelen utilizarse para designar acciones y procesos;
- y los *adjetivos* describen propiedades y estados de los nombres.

Introducción (II)

Las palabras de las mismas categorías gramaticales realizan, en general, las mismas funciones sintácticas en el lenguaje.
Por ejemplo, los nombres pueden actuar de *raíz* o *modificador* en los sintagmas nominales, y de *sujeto* en las oraciones.

El valiente { príncipe, guerrero, sastrecillo, enano, elfo, fontanero, *Shrek* } salvó a la princesa.

Introducción (III)

Clases abiertas y cerradas

Se dice que una parte del lenguaje es una clase abierta cuando continuamente se están añadiendo nuevos miembros a esa clase:

- Nombres
- Verbos
- Adjetivos
- Adverbios

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe.
All mimsy were the borogroves
And the mome raths outgrabe.

Long ago, in a finite state far away, there lived a JOVIAL character named Jack. Jack and his relations were poor. Often their hash table was bare. One day Jack's parent said to him, "Our matrices are sparse. You must go to the market to exchange our RAM for some BASICs." She compiled a linked list of items to retrieve and passed it to him.

So Jack set out. But as he was walking along a path, he met the traveling salesman.

"Whither dost thy flow chart take thou?" prompted the salesman in high-level language.

"I'm going to the market to exchange this RAM for some chips and Apples," commented Jack.

"I have a much better algorithm. You needn't join a queue there; I will swap your RAM for these magic kernels now."

Jack made the trade, then backtracked to his house. But when he told his busy-waiting parent of the deal, she became so angry she started thrashing.

"Don't you even have any artificial intelligence? All these kernels together hardly make up one byte," and she popped them out the window... – Mark Isaak, "Jack and the Beanstack".

Introducción (V)

Clases abiertas y cerradas

Clases cerradas son aquellas que permanecen invariables en largos periodos de tiempo:

- Preposiciones
- Determinantes
- Pronombres
- Conjunciones

Pueden variar en largos períodos de tiempo:
cabe, magüer, ...
Dialectales: a más a más, todo y que

Introducción (VI)

Partes del lenguaje

Es posible subdividirlas tanto como haga falta, en función de las necesidades.

Las etiquetas de PoS del Brown Corpus, las del Penn Treebank y las del BNC son las más utilizadas.

part-of-speech	morphological variation	tag
noun	singular	NN
	plural	NNS
	proper, singular	NNP
	proper, plural	NNPS
adjective	normal	JJ
	comparative	JJR
	superlative	JJS
verb	base	VB
	non-3rd, present tense	VBP
	3rd person, present	VBZ
	past tense	VBD
	past participle	VBN
	gerund	VBG

Introducción (VII)

Partes del lenguaje

part-of-speech	morphological variation	tag
personal pronoun	nominative	PRP
	genitive	PRP$
	interrogative	WP
	interr., gen.	WP$
adverb	normal	RB
	comparative	RBR
	superlative	RBS
	interrogative	WRB
predeterminer		PDT
determiner		DT
preposition		IN
conjunction	copulative	CC
...	...	...

Introducción (VIII)

Etiquetado de las partes del lenguaje

- Consiste en etiquetar cada palabra con la parte del lenguaje a la que pertenece.
- Se considera un paso previo al análisis sintáctico.

Ambigüedad:

Time_{NN} flies_{VBZ} like_{IN} an_{DT} arrow_{NN}
Time_{NN} flies_{NNS} like_{VBP} an_{DT} arrow_{NN}
Time_{VB} flies_{NNS} like_{IN} an_{DT} arrow_{NN}
Time_{NN} flies_{NNS} like_{IN} an_{DT} arrow_{NN}

The_{DT} horse_{NN} raced_{VBD} past_{IN} the_{DT} barn_{NN} $\text{fell}_{.}$.

The_{DT} horse_{NN} [raced_{VBP} past_{IN} the_{DT} barn_{NN}] fell_{VBD}.

Introducción (IX)

Fuentes de información

- Información léxica:
  - Palabras que acaban en *ando* suelen ser verbos en gerundio.
  - Palabras que acaban en *s* son a menudo nombres en plural.
- Contexto:
  - Palabras precedidas por una preposición suelen ser determinantes o nombres.
  - Palabras precedidas por un artículo, nombres.
- Etiquetas posibles para cada palabra:
  - Cada palabra puede tomar sólo ciertas etiquetas en cada lenguaje (p.ej., *bebida* como nombre o participio).
  - La asignación a cada palabra de su etiqueta más frecuente: 90% precisión (entrenado en el mismo corpus).

Introducción (X)

Procedimientos:

- Listas de transformación.
- Modelos de Markov
- Otros (Entropía Máxima, árboles de decisión, etc.)

Etiquetado con listas de transformación (I)

(Brill, 1995)

Una lista de transformación (*transformation list*) es una lista de reglas con la siguiente sintaxis:

If *precondition* then *change tag to* XXX

El funcionamiento es el siguiente:

1. Asignar una etiqueta inicial a cada palabra.
2. Para cada regla de la lista (por orden),
   - Aplicarla a cada palabra del texto.

Etiquetado con listas de transformación (II)

Ejemplo:

Initial tagging:

All-NN the-NN boys-NN and-NN the-NN girls-NN came-NN

> If the word is currently tagged as `NN`
> and it ends with an `s`,
> then retag it as `NNS`.

Next tagging:

All-NN the-NN boys-NNS and-NN the-NN girls-NNS came-NN

Etiquetado con listas de transformación (III)

Algoritmo de aprendizaje

Entrenamiento(textoNoAnotado textoAnotado)

Inicializar:

Leer el texto no anotado

Inicializar las etiquetas

(p.ej., todas como nombre singular, `NN`)

Repetir:

Comparar el texto con el anotado

Encontrar la regla que maximice las

correcciones realizadas sobre el texto.

Añadir esta regla al final de la lista.

Aplicarla al texto de entrenamiento.

hasta que la mejora $<$ *umbral*.

Devolver la lista completa.

Etiquetado con listas de transformación (IV)

Algoritmo de etiquetado

Etiquetado(textoNoAnotado)

Inicializar:

Leer el texto no anotado

Inicializar las etiquetas

(p.ej., todas como nombre singular, `NN`)

Para cada regla r**:**

Aplicarla al texto.

Devolver el texto anotado.

Etiquetado con listas de transformación (V)

Ejemplo de reglas

The/NN good-looking/NN dogs/NN barked/NN

Rule	**Text**
NN s fhassuf 1 NNS x	The/NN good-looking/NN dogs/NN barked/NN
NN . fchar CD x	The/NN good-looking/NN dogs/NNS barked/NN
NN - fchar JJ x	The/NN good-looking/JJ dogs/NNS barked/NN
NN ed fhassuf 2 VBN x	The/NN good-looking/JJ dogs/NNS barked/VBN
...	...
NN the fhassuf 3 DT x	The/DT good-looking/JJ dogs/NNS barked/VBD
...	...
VBN NNS prevword VBD	The/DT good-looking/JJ dogs/NNS barked/VBD

Etiquetado con listas de transformación (VI)

- Son más expresivas que las listas de decisión y los árboles de decisión.
- Una regla puede deshacer lo que ha hecho otra regla anterior en un caso particular.
- El aprendizaje es muy lento, pues hay que evaluar a cada paso muchas posibilidades.
- El etiquetado de textos nuevos, en cambio, se puede realizar en tiempo lineal (número de reglas $\times$ número de palabras).
- No da varias posibles etiquetaciones en casos dudosos.
- Precisión: alrededor del 95-96%.

Etiquetado con modelos de Markov (I)

Notación:

w^l La $l^{\underline{a}}$ palabra del lenguaje.

t^j La $j^{\underline{a}}$ etiqueta.

w_i La palabra en la posición i del corpus.

t_i La etiqueta asignada a w_i.

Procedimiento:

Dado un texto con I palabras, se trata de obtener la secuencia de etiquetas $\{t_1, ..., t_i, ..., t_I\}$:

$\text{argmax}_{(t_1,...,t_n)}\ P(t_1, ..., t_n | w_1, ..., w_n)$

Etiquetado con modelos de Markov (II)

$$\begin{aligned}
&\text{argmax}_{(t_1,...,t_n)}\ P(t_1,...,t_n|w_1,...,w_n) \\
&\quad = \text{argmax}_{(t_{1...n})}\ \frac{P(w_{1...n}|t_{1...n})P(t_{1...n})}{P(w_{1...n})} \\
&\quad = \text{argmax}_{(t_{1...n})} P(w_{1...n}|t_{1...n})P(t_{1...n})
\end{aligned}$$

Podemos hacer dos hipótesis para simplificar el problema:

- Las palabras son independientes unas de otras.
- La identidad de una palabra sólo depende de su etiqueta.

$$= \text{argmax}_{(t_{1...n})} \prod_{i=1}^{n} P(w_i|t_i)P(t_{1...n})$$

Etiquetado con modelos de Markov (III)

Utilizando cadenas de Markov de segundo orden, se trata de maximizar:

$$\left[\prod_{i=1}^{I} P(t_i|t_{i-1}, t_{i-2}) P(w_i|t_i) \right] P(t_{T+1}|t_T)$$

donde t_{-1} y t_{T+1} son marcadores de inicio y fin de frase.

Estimación de los parámetros (I)

Primeramente, se puede obtiener la frecuencia de unigramas, bigramas y trigramas utilizando los estimadores de *maximum likelihood*:

$$\hat{P}(t_3) = \frac{f(t_3)}{N}$$

$$\hat{P}(t_3|t_2) = \frac{f(t_2, t_3)}{f(t_2)}$$

$$\hat{P}(t_3|t_2, t_1) = \frac{f(t_1, t_2, t_3)}{f(t_1, t_2)}$$

Igualmente, la probabilidad por cada palabra:

$$\hat{P}(w_3|t_3) = \frac{f(w_3, t_3)}{f(t_3)}$$

Estimación de los parámetros (II)

Las probabilidades obtenidas no se pueden aplicar directamente debido al problema de datos escasos (*sparse data problem*).

Aunque las probabilidades estimadas $\hat{P}$ tienden a la probabilidad real conforme aumenta el tamaño del corpus de entrenamiento,

> por grande que sea el corpus, siempre habrá fenómenos lingüísticos que no aparezcan en él.

Los estimadores $\hat{P}$ siempre asignan probabilidad 0 a fenómenos no observados.

Estimación de los parámetros (III)

El problema se mitiga con técnicas de *smoothing*:

- Ley de Laplace
- Leyes de Lidstone y Jeffreys-Perks
- Estimador *Held-out*
- Deleted interpolation
- Otros...

Estimación de los parámetros (IV)

Ley de Laplace

$$P_{Lap}(t_1, ..., t_n) = \frac{f(t_1, ..., t_n) + 1}{N + B}$$

donde N es el número de trigramas en el corpus, y B es el número de trigramas diferentes.

Problema: Da demasiado peso a los n-gramas no vistos.

Ejemplo: Tenemos tres etiquetas: `NN`, `JJ` y `DT` (27 posibles trigramas):

Trigrama	**Frecuencia**	$\hat{P}$	P_{Lap}
DT JJ NN	10	0.5	0.23
DT NN NN	9	0.45	0.21
JJ NN NN	1	0.05	0.043
otro	0	0	0.03

Incluso a trigramas gramaticalmente incorrectos!: `NN NN DT`

Estimación de los parámetros (V)

Ley de Lidstone

$$P_{Lap}(t_1, ..., t_n) = \frac{f(t_1, ..., t_n) + \lambda}{N + B\lambda}$$

donde N es el número de trigramas en el corpus, y B es el número de trigramas diferentes.
Se puede demostrar que equivale a una interpolación lineal entre el estimador de *maximum likelihood* y una función de probabilidad uniforme.

Ley de Jeffreys-Perks: Tomar $\lambda = 0.5$

$$P_{Lap}(t_1, ..., t_n) = \frac{f(t_1, ..., t_n) + 0.5}{N + 0.5 \times B}$$

Estimación de los parámetros (VI)

Problemas:

- Escoger el valor de λ apropiado.
- Sigue dando probabilidad positiva a fenómenos que nunca pueden ocurrir.

Ejemplo: Tenemos tres etiquetas: `NN`, `JJ` y `DT` (27 posibles trigramas). Con $\lambda = 2$,

Trigrama	**Frecuencia**	$\hat{P}$	P_{Lap}
DT JJ NN	10	0.5	0.33
DT NN NN	9	0.45	0.28
JJ NN NN	1	0.05	0.045
otro	0	0	0.015

Estimación de los parámetros (VII)

Estimador Held-out

El corpus de entrenamiento se divide en dos partes:

- Una primera parte, para calcular las frecuencias de los n-gramas: $f_1(t_1, ..., t_n)$
- Una segunda parte (*held out*), para ver, al encontrarnos texto nuevo, cómo varían los estimadores: $f_2(t_1, ..., t_n)$.

N_r = número de n-gramas con frecuencia $f_1 = r$.
T_r = número de veces que aparecen en la segunda parte todos los trigramas que aparecían r veces en la primera.

$$T_r = \sum_{\{t_{1...n}: f_1(t_{1...n})=r\}} f_2(t_{1...n})$$

$$P_{ho}(t_{1...n}) = \frac{T_r}{N_r N}$$

Estimación de los parámetros (VIII)

Ejemplo: Tenemos tres etiquetas: `NN`, `JJ` y `DT` (27 posibles trigramas).

Trigrama	**f**	f_1	f_2	$\hat{P}$	P_{Lap}
DT JJ NN	10	5	5	0.5	0.45
DT NN NN	9	5	4	0.45	0.45
JJ NN NN	1	0	1	0.05	0.004
otro	0	0	0	0	0.004

Estimación de los parámetros (IX)

Interpolación por eliminación (deleted interpolation

Similar al Held-out, pero cada parte en que se divide el corpus hace de held-out para la otra parte.

Supongamos que dividimos el corpus en dos, a y b:

N_r^a = número de n-gramas con frecuencia $f_a = r$.

T_r^{ab} = número de veces que aparecen los $n$gramas de la parte a con frecuencia r en la parte b.

$$P_{ho}(t_{1...n}) = \frac{T_r^{ab}}{N_r^a N} \text{o bien} \frac{T_r^{ba}}{N_r^b N}$$

$$P_{del}(t_{1...n}) = \frac{T_r^{ab} + T_r^{ba}}{N(N_r^a + N_r^b)}$$

Estimación de los parámetros (X)

Ejemplo: Tenemos tres etiquetas: `NN`, `JJ` y `DT` (27 posibles trigramas).

Trigrama	**f**	f_1	f_2	$\hat{P}$	P_{Lap}
DT JJ NN	10	5	5	0.5	0.474
DT NN NN	9	5	4	0.45	0.474
JJ NN NN	1	0	1	0.05	0.002
otro	0	0	0	0	0.002

Estimación de los parámetros (XI)

Interpolación lineal

Es probable que un trigrama $t_1t_2t_3$ no ocurra nunca en el corpus pero que, o bien t_2t_3, o bien t_3 sean muy frecuentes.

Eso indica que quizá no han coincidido nunca con t_1, pero que hay cierta probabilidad de que ocurra.

$$P_{li}(t_3|t_2,t_1) = \lambda_1\hat{P}(t_3|t_2,t_1) + \lambda_2\hat{P}(t_3|t_2) + \lambda_3\hat{P}(t_3)$$

Variaciones:

- Hacer depender λ_i de la historia previa.
- Si el estimador de trigramas se considera fiable, no interpolar.

Etiquetado con modelos de Markov (IV)

Una vez hemos estimado los parámetros, se trataba de encontrar la secuencia de etiquetas que maximice

$$\left[\prod_{i=1}^{I} P(t_i|t_{i-1}, t_{i-2}) P(w_i|t_i) \right] P(t_{T+1}|t_T)$$

- Probar todas las posibles combinaciones: exponencial.
- En tiempo polinómico con el algoritmo de Viterbi.

Etiquetado con modelos de Markov (V)

(Brants, 2001)

- Cadenas de Markov de tercer orden.
- Estimación de probabilidades con interpolación lineal.
- Los parámetros λ se obtienen con *deleted interpolation*.
- Incorpora modelos de probabilidad $P(w|t_i)$ para palabras que no estuvieran en el corpus de entrenamiento.
- Incorpora al modelo de Markov el hecho de que las palabas estén capitalizadas o no.

Precisión: 96.7%.

En frases poco dudosas, llega al 99% de precisión.

En casos muy dudosos (si las probabilidades son similares), puede dar varias etiquetas con probabilidades.

Otros modelos (I)

Entropía Máxima

Las palabras se van etiquetando una a una.

Para cada palabra, podemos considerar:

- Características léxicas: prefijos, sufijos, etc.
- La historia reciente: palabras que la han precedido, y las etiquetas asignadas a esas

A partir de esa informacíon, ha de ser posible asignarle una etiqueta, para pasar a la palabra siguiente.

Otros modelos (II)

Entropía Máxima

Se consideran características binarias sobre las palabras y su historia h_i:

$$f_j(h_i, t_i) = \begin{cases} 1 & if\ suffix(w_i) = \texttt{ing}\ and\ t_i = \texttt{VBG} \\ 0 & otherwise \end{cases}$$

Con ellas, se entrena un modelo sobre la probabilidad de asignar la etiqueta t a la palabra actual, dada la historia h:

$$p(h,t) = \pi\mu \prod_{j=1}^{k} \alpha_j^{f_j(h,t)},$$

donde π es el factor de normalización, y μ, α son los parámetros del modelo.

Otros modelos (III)

Entropía Máxima

- Ratnaparkhi (1996) consiguió un 96.6% de precisión.
- Más lento de entrenar y utilizar que las cadenas de Markov.
- Al no existir algoritmos como Viterbi para estos modelos, utilizó una búsqueda en haz (*beam search*), quedándose a cada paso con las N etiquetas más probables.

Otros modelos (IV)

- Árboles de decisión,
  listas de decisión.
  $\Rightarrow$ Menos expresivas que listas de transformación
- *Hidden Markov Models* para aprendizaje no supervisado.
- Aprendizaje basado en memoria (k nearest neighbor).
- Redes neuronales.
- Bootstrapping.
- Combinación de varios métodos.
- EngCG: Reglas definidas a mano por expertos.