Análisis de Textos

Grupo PLN – InCo

Tokenización
&
Lenguajes Regulares
&
Morfología
&
DME

Unidades de texto

¿Cuáles son las unidades independientes más pequeñas del texto?

– Segmento del discurso unificado habitualmente por el acento, el significado y pausas potenciales inicial y final

– Aquellas que puede existir en forma libre y que conforma el enunciado o mensaje lingüístico. Están dotadas de significado léxico o gramatical, según el caso

*Son entonces **signos lingüísticos***

Palabras

- La **palabra** es un conjunto o secuencia de sonidos articulados que se pueden representar gráficamente con letras, y por lo general, asocian un significado
- Mínima unidad con significado (Aristóteles)

Unidad de texto = Palabra

Palabras, Tipos y Tokens

- Palabras: unidades que hay en un corpus o en un vocabulario

- Tipos (word types): unidades *distintas* que hay en un corpus o en un vocabulario

- Tokens: son las instancias de los tipos en un corpus o en un vocabulario

Palabras, Tipos y Tokens

Vamos a Buenos Aires a pasear por el Delta del Tigre

¿Cuántas palabras?	¿Cuántos tipos?	¿Cuántos tokens?
11	10	8

Palabras, Tipos y Tokens

La Plaza Matriz está ubicada en la Ciudad Vieja.

¿Cuántas palabras? ¿Cuántos tipos? ¿Cuántos tokens?

9	8	7
9	8	8
9	8	9

Los signos de puntuación también nos van a interesar tenerlos identificados en un algoritmo de tokenización

Palabras, Tipos y Tokens

... entró al taller de Aureliano y le preguntó: '¿Qué día es hoy?' Aureliano le contestó que era martes. 'Eso mismo pensaba yo', dijo José Arcadio Buendía. 'Pero de pronto me he dado cuenta de que sigue siendo lunes, como ayer. Mira el cielo, mira las paredes, mira las begonias. También hoy es lunes.' Acostumbrado a sus manías, Aureliano no le hizo caso. Al día siguiente, miércoles, José Arcadio Buendía volvió al taller. 'Esto es un desastre –dijo–. Mira el aire, oye el zumbido del sol, igual que ayer y ...

¿Cuántas palabras? ¿Cuántos tipos? ¿Cuántos tokens?

Palabras, Tipos y Tokens

... entró al taller de Aureliano y le preguntó: '¿Qué día es hoy?' Aureliano le contestó que era martes. 'Eso mismo pensaba yo', dijo José Arcadio Buendía. 'Pero de pronto me he dado cuenta de que sigue siendo lunes, como ayer. Mira el cielo, mira las paredes, mira las begonias. También hoy es lunes.' Acostumbrado a sus manías, Aureliano no le hizo caso. Al día siguiente, miércoles, José Arcadio Buendía volvió al taller. 'Esto es un desastre –dijo–. Mira el aire, oye el zumbido del sol, igual que ayer y ...

¿Cuántas palabras? ¿Cuántos tipos? ¿Cuántos tokens?

Corpus

¿Qué es un Corpus?

- Es una colección de material lingüístico de ejemplos reales de uso de la lengua

- Es de utilidad en diferentes áreas, principalmente en lingüística computacional y lingüística teórica

Corpus

¿Cómo se construye un corpus?

- Recopilación de un conjunto de documentos
- Hay que definir las características deseadas:
  - escrito / oral
  - idioma (un idioma o multilingüe)
  - tipo de texto (prensa, literario, científico, ...)
  - dominio (arte, lingüística, deportes, informática, ...)
  - anotado / no anotado (conjunto de etiquetas)
  - ...

Corpus

Algunos corpus en inglés

- Brown Corpus
- Penn Treebank
- PropBank

Algunos corpus en español

- CREA y CORDE (RAE)
- Corpus del español de Mark Davies
- Ancora
- Adesse
- “Nuestros”

Lenguajes Regulares

English is not a finite state language.
(Chomsky 1957)

Lenguajes Regulares

- Lenguaje para identificar strings de caracteres (Kleene, 1956)
- Herramienta para especificar e identificar textos mediante patrones
- Expresividad limitada, pero *muy* eficientes.

Se requiere de:

- Patrón (***qué*** se quiere buscar)
- Corpus (***dónde*** se quiere buscar)

Lenguajes Regulares

Notación formal para definir lenguajes regulares sobre un alfabeto Σ

- $\varnothing$ es una ER que describe al conjunto $\varnothing$

- **a** es una ER $\forall a \in \Sigma \cup \{\varepsilon\}$

- Si r y s son ER para describir R y S respectivamente entonces:
  - (r|s) es una ER para $R \cup S$, *unión*
  - (r.s) es una ER para R.S, *concatenación*
  - (r*) es una ER para R*, *clausura de Kleene*

Estos son todas las Expresiones Regulares definidas sobre Σ

Lenguajes Regulares

Algunos ejemplos:

ER	Patrón encontrado
cabeza	de la **cabeza** al sombrero
!	Vení para acá**!**
[abc]	solo de no**c**he
[0-9]	Capítulo **1**: Introducción
[^A-Z]	M**a**ñana va a ser un gran día
[^sS]	**T**engo permiso

Lenguajes Regulares

Algunos ejemplos:

ER	Patrón encontrado
cabeza	de la **cabeza** al sombrero
!	Vení para acá**!**
[abc]	solo de no**c**he
[0-9]	Capítulo **1**: Introducción
[^A-Z]	M**a**ñana va a ser un gran día
[^sS]	**T**engo permiso

¿Cómo hacemos para buscar cualquiera de las ocurrencias de oveja en una lista?

oveja

Oveja

Lenguajes Regulares

Algunos ejemplos:

ER	Patrón encontrado
cabeza	de la **cabeza** al sombrero
!	Vení para acá**!**
[abc]	solo de no**c**he
[0-9]	Capítulo **1**: Introducción
[^A-Z]	M**a**ñana va a ser un gran día
[^sS]	**T**engo permiso

¿Cómo hacemos para buscar cualquiera de las ocurrencias de oveja en una lista?

oveja

Oveja

[oO]veja

Lenguajes Regulares

Algunos ejemplos:

ER	Patrón encontrado
cabeza	de la **cabeza** al sombrero
!	Vení para acá**!**
[abc]	solo de no**c**he
[0-9]	Capítulo **1**: Introducción
[^A-Z]	M**a**ñana va a ser un gran día
[^sS]	**T**engo permiso

¿Cómo hacemos para buscar cualquiera de las ocurrencias de oveja en una lista?

oveja

Oveja

[oO]veja

re.match('[oO]veja',ele):

Lenguajes Regulares

Algunos ejemplos:

ER	Patrón encontrado
cabeza	de la **cabeza** al sombrero
!	Vení para acá**!**
[abc]	solo de no**c**he
[0-9]	Capítulo **1**: Introducción
[^A-Z]	M**a**ñana va a ser un gran día
[^sS]	**T**engo permiso

¿Cómo hacemos para buscar cualquiera de las ocurrencias de oveja en una lista?

oveja

Oveja

[oO]veja

```
lista = ['vaca','oveja','chancho','Vaca','Oveja','cabra']
for ele in lista:
if re.match('[oO]veja',ele):
        print(ele)
```

Lenguajes Regulares

Más ejemplos:

ER	Patrón encontrado
[e^]	símbolo **^** significa que....
a^b	cuando encuentres "**a^b**" es
casas?	Juan había ido a **casa** de
pec.s	tenía **pecas** en toda la cara
/beee*!/	La oveja hace **beeeeee!**
/bee+!/	La oveja hace **beeeeee!**
/\bel\b/	Caminaba por **el** camino de

Lenguajes Regulares

Operadores avanzados

\d → cualquier dígito [0-9]
\w → cualquier carácter alfanumérico o “espacio”
\s → [\b\t\n]
y las mayúsculas son la negación

{n} → **n** ocurrencias del carácter o expresión anterior
{n,m} → de **n** a **m** ocurrencias “ ”
{n,} → al menos **n** ocurrencias “ ”

Lenguajes Regulares

Sustitución: sustituye un patrón por otro

s / ([0-9]+) / 'Hola' /

… en lugar de 30 hay 4
x
… en lugar de Hola hay Hola

s / ([0-9]+) / <\1> /

donde el \1 como segundo patrón, refiere al primero

… tenía el número 10 en la espalda …
x
… tenía el número <10> en la espalda …

Lenguajes Regulares

Sustitución: sustituye un patrón por otro

s / ([0-9]+) / ‘Hola’ /

s / ([0-9]+) / <\1> /

---

patron = re.sub('[0-9]+', 'Hola','Tiene el numero 10 en la espalda')

Tiene el numero Hola en la espalda

patron = re.sub(r'([0-9]+)',r'<\1>','Tiene el numero 10 en la espalda')

Tiene el numero <10> en la espalda

Lenguajes Regulares

ELIZA: programa que simula una conversación entre un paciente y un psicólogo (Weizembaum, 1966)

Paciente: Men are all alike
ELIZA: IN WHAT WAY
Paciente: They're always bugging us about something or other
ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE
Paciente: Well, my boyfriend made me come here
ELIZA: YOUR BOYFRIEND MADE YOU COME HERE
Paciente: He says I'm depressed much of the time
ELIZA: I AM SORRY TO HEAR YOU ARE DEPRESSED

Lenguajes Regulares

Algunas sustituciones usadas:

s/ .* I AM (depressed|sad) .*/ I AM SORRY TO HEAR YOU ARE \1 /

s/ .* MY (mother | father | boyfriend | girlfriend | friend) MADE ME COME HERE .*/ WHY DO YOU THINK YOUR \1 YOU COME HERE /

s/ .* all .*/ IN WHAT WAY /

s/ .* always .*/ CAN YOU THINK OF A SPECIFIC EXAMPLE /

Autómatas Finitos

Máquina de estados

- En particular, nos interesan desde el punto de vista del reconocimiento de ***Lenguajes Regulares***

- Un *lenguaje regular* es el conjunto de strings sobre un alfabeto Σ reconocidos por *autómatas finitos*

Autómatas Finitos

Un AFD es una máquina de estados que se puede representar por la siguiente quíntupla

$M : (Q, \Sigma, \delta, q_o, F)$ donde:

- Q: conjunto de estados
- Σ: alfabeto
- δ: función de transición / $\delta: Q \times \Sigma \rightarrow Q$
- q_o: estado inicial / $q_o \in Q$
- F: conjunto de estados finales (aceptores) / $F \subseteq Q$

Autómatas Finitos

Ejemplo: el lenguaje de las ovejas

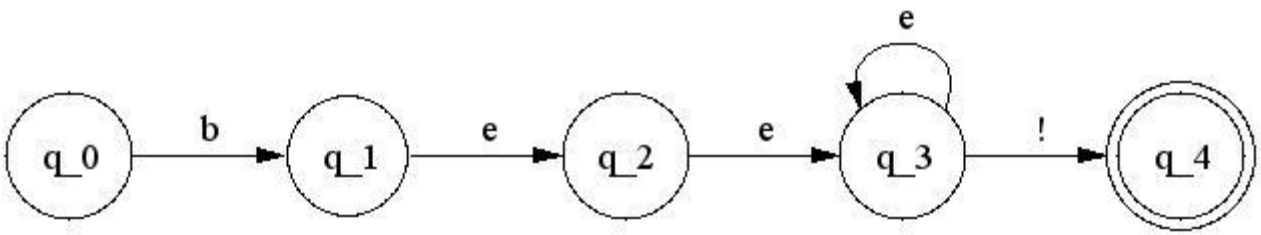
lo podemos ver como secuencias (infinitas) de tiras del tipo

bee!
beee!
beeee!
beeeee!
...

donde la ER asociada sería /bee+!/

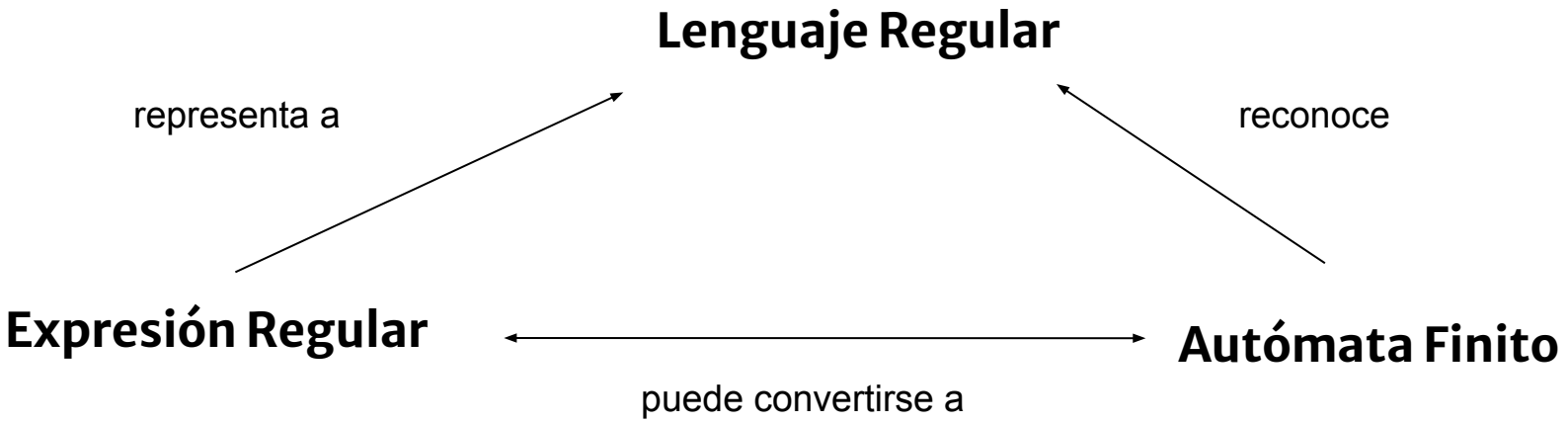
Autómatas Finitos

y el autómata finito....



Autómatas Finitos

Cualquier ER puede ser "implementada" por un AF y recíprocamente



Autómatas Finitos

- No podemos modelar el lenguaje natural con expresiones regulares

- Podemos modelar algunas cosas:
  - Fonología
  - Morfología
  - Sintaxis (algo)

Tokenización

- Identificar las distintas unidades (*tokens*) en un texto
- Los espacios separan tokens, pero…
  - escribió “El príncipe feliz”
  - el 24 de agosto de 1889
  - Hay 10,000 razones para no creer
  - Lo busqué en http://rulzindeed.blogspot.com
  - Je t’aime rock’n’roll
  - New York
  - estado del arte
  - al del
- El estándar de tokenización del Penn Treebank

Tokenización

- El chino y el japonés no marcan los límites de palabras

  小時候沒有人選擇我踢足球

  De chico nadie me elegía para jugar al fútbol

- En alemán algunas palabras compuestas pueden escribirse todas juntas
  - hora pico → hauptverkehrszeit
  - a veces → manchmal
  - jugo de frutas → fruchtsaft

Tokenización

Algoritmo *MaxMatch*

- Basado en una lista de palabras
- Comienza al principio de la entrada
- Elige siempre la palabra más larga en la posición actual de la entrada
- Si no encuentra ninguna palabra, se crea una palabra de una letra
- Avanza al puntero a la primera posición luego de la palabra encontrada

Tokenización

Algoritmo *MaxMatch*

Entrada:

"mesacadelacanchasinmotivo"

Avance (en negrita la entrada ya analizada)

mesacadelacanchasinmotivo
mesacadelacanchasinmotivo
mesacadelacanchasinmotivo
mesacadelacanchasinmotivo
...

Salida:

[Mesa, ca, de, la, cancha, sin, motivo] ó
[Mesa, ca, del, a, cancha, sin, motivo] ó
[Mes, aca, de, la, cancha, sin, motivo]

Tokenización

¿Cómo evaluamos un tokenizador?

- Input 1: Nuestra segmentación
  [Mesa, ca, de, la, cancha, sin, motivo]
- Input 2: La segmentación correcta *(gold standard)*
  [Me, saca, de, la, cancha, sin motivo]

Word Error Rate (Distancia Mínima de Edición)

¿cuántas palabras deben insertarse, borrarse o sustituirse para ir de Input 1 a Input 2?

(En el ejemplo anterior, 2)

Tokenización

Tokenizador en Python

(utilizando la biblioteca NLTK)

```
import nltk
nltk.download('punkt')
nltk.word_tokenize('Hoy es un lindo dia.')
```

```
['Hoy', 'es', 'un', 'lindo', 'dia', '.']
```

Normalización

Normalización: llevar las palabras a un formato estándar

- Llevar los números a un formato único
- URLs y otras formas con estructura
- Detección de entidades con nombre
- Llevar todo a minúsculas/mayúsculas
- ...

Tradicionalmente, para la tokenización y normalización se han utilizado técnicas modeladas con autómatas finitos.

Morfología

morfología (de *morfo* (forma) y *logía* (ciencia))

Rama de una disciplina que se ocupa del **estudio y la descripción de las formas externas de un objeto**

Se puede aplicar al estudio de:

- los seres vivos (Biología)
- la superficie terrestre (Geomorfología)
- las palabras (Lingüística)

Morfología

morfología (de *morfo* (forma) y *logía* (ciencia))

Rama de una disciplina que se ocupa del **estudio y la descripción de las formas externas de un objeto**

Se puede aplicar al estudio de:

- los seres vivos (Biología)
- la superficie terrestre (Geomorfología)
- las palabras (Lingüística)

f. *Gram.* Parte de la lingüística que se ocupa de la **estructura** o forma de las palabras.

Morfología

- Mecanismos de formación / análisis de la palabras
- **Análisis morfológico**: Reconocer una palabra y construir una representación estructurada

análisis

Gatitos → gato + Masc + Pl + Dim

←

generación

Morfología

- Morfema: fragmento mínimo capaz de expresar significado. Es la mínima unidad con sentido.

- Raíz: es la parte de la palabra que no varía y que indica su significado principal. Es el "morfema "principal" o lexema

Los morfemas se añaden a la raíz para formar nuevas palabras.

Morfología

- Morfema: fragmento mínimo capaz de expresar significado. Es la mínima unidad con sentido

- Raíz: es la parte de la palabra que no varía y que indica su significado principal. Es el "morfema "principal" o lexema

Los morfemas se añaden a la raíz para formar nuevas palabras.

- Afijos: dan significado adicional

Morfología

- Morfema: fragmento mínimo capaz de expresar significado. Es la mínima unidad con sentido

- Raíz: es la parte de la palabra que no varía y que indica su significado principal. Es el "morfema "principal" o lexema

Los morfemas se añaden a la raíz para formar nuevas palabras.

- Afijos: dan significado adicional
  - Prefijos: im+posible

Morfología

- Morfema: fragmento mínimo capaz de expresar significado. Es la mínima unidad con sentido

- Raíz: es la parte de la palabra que no varía y que indica su significado principal. Es el "morfema "principal" o lexema

Los morfemas se añaden a la raíz para formar nuevas palabras.

- Afijos: dan significado adicional
  - Prefijos: im+posible
  - Sufijos: gat+ito+s

Morfología

- Morfema: fragmento mínimo capaz de expresar significado. Es la mínima unidad con sentido

- Raíz: es la parte de la palabra que no varía y que indica su significado principal. Es el "morfema "principal" o lexema

Los morfemas se añaden a la raíz para formar nuevas palabras.

- Afijos: dan significado adicional
  - Prefijos: im+posible
  - Sufijos: gat+ito+s
  - Circunfijos: a+naranj+ado (Parasintéticas)

Morfología

- Morfema: fragmento mínimo capaz de expresar significado. Es la mínima unidad con sentido

- Raíz: es la parte de la palabra que no varía y que indica su significado principal. Es el "morfema "principal" o lexema

Los morfemas se añaden a la raíz para formar nuevas palabras.

- Afijos: dan significado adicional
  - Prefijos: im+posible
  - Sufijos: gat+ito+s
  - Circunfijos: a+naranj+ado (Parasintéticas)
  - Infijos: No hay (hingi => h**um**ingi Lengua Tagaloga)

Morfología

- Lematización: llevar palabras con la misma raíz a una forma canónica. Identificar su estructura interna

  Por ejemplo

  en español:

  soy, son, es → ser

  perro, perra, perros → perro

  en inglés:

  am, are, is → be

- Lema: palabra “representativa”

Morfología

Ejemplos de palabras con morfemas

…

Morfología

Ejemplo de detección de morfemas

Juan comía lentamente de manera irracional y sin desesperación.

Morfología

Ejemplo de detección de morfemas

*Juan com**ía** lenta**mente** de manera **ir**racional y sin **des**esperac**ión**.*

Morfología

- Stemming: cortar las palabras. Mucho más simple, en los hechos ha funcionado

- Stemmer de Porter (1980): una serie de reglas de reescritura en cascada

Morfología

Algoritmo de Porter (ejemplo de reglas)

Step 1a

sses	→ ss	caresses	→ caress
ies	→ i	ponies	→ poni
ss	→ ss	caress	→ caress
s	→ ø	cats	→ cat

Step 1b

(*v*)ing	→ ø	walking	→ walk
		sing	→ sing
(*v*)ed	→ ø	plastered	→ plaster
…			

Step 2 (for long stems)

ational	→ ate	relational	→ relate
izer	→ ize	digitizer	→ digitize
ator	→ ate	operator	→ operate
…			

Step 3 (for longer stems)

al	→ ø	revival	→ reviv
able	→ ø	adjustable	→ adjust
ate	→ ø	activate	→ activ
…			

Morfología

Hay algunos problemas a resolver:

- Morfotáctica: los morfemas pueden combinarse de acuerdo a ciertas reglas

  inevitable

  *inelefante

  inelefantemente ?

- Alteraciones ortográficas: los morfemas pueden cambiar según el contexto

  Pez → Pezs → Peces

  Maní → Manís → Maníes

Morfología

Morfología Flexiva

- Mecanismo de producción de palabras dentro de una misma clase

  com – o / com – ía / com – eré

- En español no se agrega significado extra

- Las flexiones aportan información relativa a:

  Género / Número Persona / Tiempo / Modo

Ejemplo: etiquetas Eagle

Morfología

Etiquetas Eagle

son un estándar para la anotación morfosintáctica de lexicones y corpus para todas las lenguas europeas

Forma	Lema	Etiqueta
alegres	alegre	AQ0CS00
hábilmente	hábil	RG000
el	el	DA0MS0
ninguna	ninguno	DI3FS00
ningunos	ninguno	DI3MP00
Juan	juan	NP00000
es	ser	VAIP3S0

Morfología

Morfología Derivativa (o léxica)

- Combinar una raíz con un afijo, para generar una palabra de *otra* clase, o con *otro* significado
  - descubrir (verbo) → descubrimiento (sustantivo)
  - estable (adjetivo) → estabilizar (verbo)
    → estabilización (sustantivo)
    → desestabilización (sustantivo)
- Es un mecanismo *productivo*

Morfología

Apenas él le amalaba el noema, a ella se le agolpaba el clémiso y caían en hidromurias, en salvajes ambonios, en sustalos exasperantes. Cada vez que él procuraba **relamar** las incopelusas, se enredaba en un grimado quejumbroso y tenía que **envulsionarse** de cara al nóvalo, sintiendo cómo poco a poco las arnillas se espejunaban, se iban **apeltronando**, **reduplimiendo**, hasta quedar tendido como el trimalciato de ergomanina al que se le han dejado caer unas fílulas de cariaconcia. Y sin embargo era apenas el principio, porque en un momento dado ella se **tordulaba** los hurgalios, consintiendo en que él aproximara suavemente sus orfelunios….

(*Rayuela – Julio Cortázar*)

Morfología

Tipos de lenguajes según su morfología:

- Flexionales: inglés, español
- Aislantes: chino (solamente un morfema por palabra)
- Aglutinantes: turco (pegan muchos morfemas)

  Ejemplo:

  uygarlık edemeyenler arasında olsaydınız

  comportándote como si estuvieras entre aquellos que no pudimos civilizar
- Polisintéticos: lenguas indígenas americanas; esquimales (jupik)

POS-tagger

- Proceso que analiza el texto a nivel de palabra
- Desambiguan según el contexto
- Ofrecen información categorial y morfológica, dependiendo de los rasgos de cada categoría
- Asignan a cada palabra su lema:
    - verbos → infinitivo
    - nombres → singular
    - adjetivos y determinantes → masculino singular
    - categorías invariantes → lema= palabra
- Muchos taggers incorporan información extra: reconocimiento y clasificación de nombres propios (NER), reconocimiento de fechas, expresiones multipalabra.

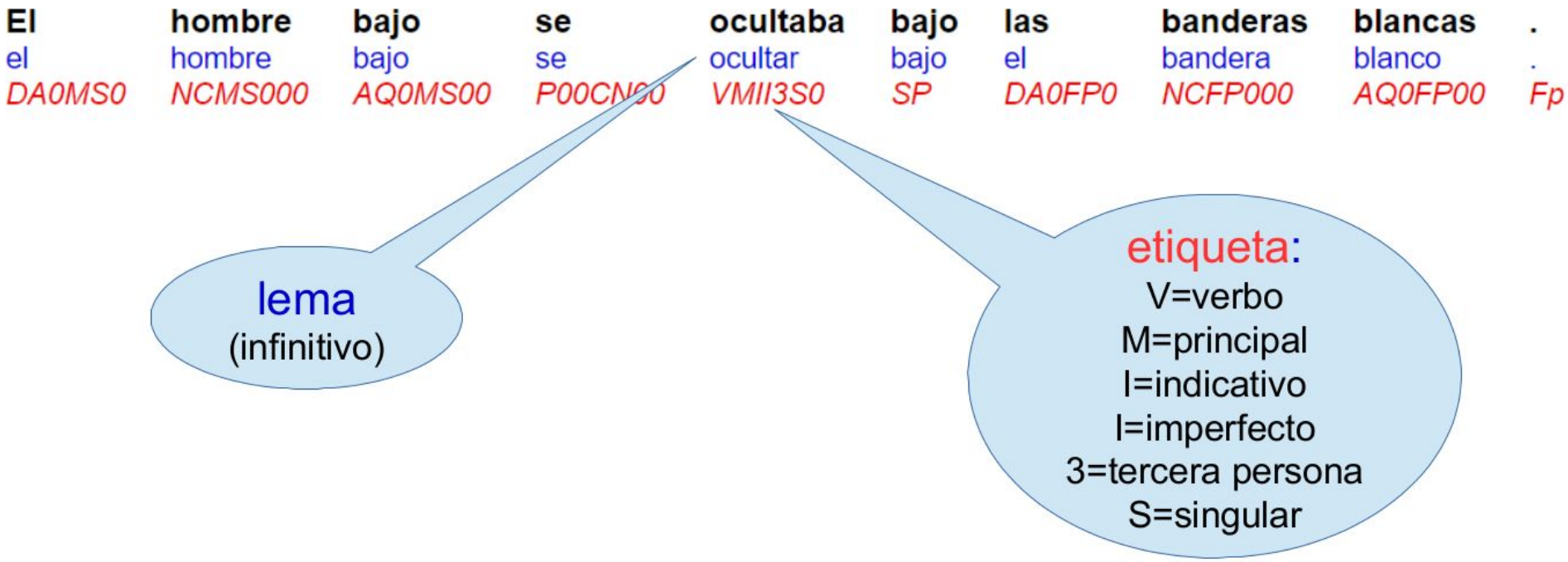
POS-tagger

Ejemplo del POS-tagger de FreeLing

El	hombre	bajo	se	ocultaba	bajo	las	banderas	blancas	.
el	hombre	bajo	se	ocultar	bajo	el	bandera	blanco	.
DA0MS0	*NCMS000*	*AQ0MS00*	*P00CN00*	*VMII3S0*	*SP*	*DA0FP0*	*NCFP000*	*AQ0FP00*	*Fp*

POS-tagger

Ejemplo del POS-tagger de FreeLing



POS-tagger

Ejemplo del POS-tagger de FreeLing

- el perro de mi suegra

el	**perro**	**de** *SP*	**mi**	**suegra** *suegro*
el	perro	_	mi	NCFS000
DA0MS0	*NCMS000*		*DP1CSS*	

- la perra de mi suegra

la	**perra**	**de** *SP*	**mi**	**suegra** *suegro*
el	perro	_	mi	NCFS000
DA0FS0	*NCFS000*		*DP1CSS*	

https://nlp.lsi.upc.edu/freeling/demo/demo.php

POS-tagger

Vamos el sábado a Buenos Aires a pasear por el Delta del Tigre.

POS

Vamos	**el**	**sábado**	**un** *SP*	**buenos_aires**	**un** *SP*	**pasear**	**por**	**el**	**Delta_de_el_Tigre**	**.**
ir	el	[S:??/??/??:??.??:??]	_	buenos_aires	_	pasear	por	el	delta_de_el_tigre	.
VMIP1P0	*DA0MS0*	*W*		*NP00G00*		*VMN0000*	*SP*	*DA0MS0*	*NP00G00*	*Fp*

Vamos	**el**	**sábado**	**un** *SP*	**buenos_aires**	**un** *SP*	**pasear**	**por**	**el**	**delta**	**de** *SP*	**el**	**tigre**	**.**
ir	el	[S:??/??/??:??.??:??]	_	buenos_aires	_	pasear	por	el	delta	_	el	tigre	.
VMIP1P0	*DA0MS0*	*W*		*NP00G00*		*VMN0000*	*SP*	*DA0MS0*	*NCMS000*		*DA0MS0*	*NCMS000*	*Fp*

Vamos	**el**	**sábado**	**un** *SP*	**buenos_aires**	**un** *SP*	**pasear**	**por**	**el**	**delta**	**de** *SP*	**el**	**Tigre**	**.**
ir	el	[S:??/??/??:??.??:??]	_	buenos_aires	_	pasear	por	el	delta	_	el	tigre	.
VMIP1P0	*DA0MS0*	*W*		*NP00G00*		*VMN0000*	*SP*	*DA0MS0*	*NCMS000*		*DA0MS0*	*NP00G00*	*Fp*

POS-tagger

Vamos el sábado a Buenos Aires a pasear por el Delta del Tigre.

Análisis Morfológico

Vamos	el	sabado	a	Buenos Aires	a	pasear	por	el	Delta_de_el_Tigre	.
ir *VMIP1P0* 0.965015	el *DA0MS0* 1	[S:??/??/??:??.??:??] *W* 1	un *SP* 0.998775	buenos_aires *NP00G00* 1	un *SP* 0.998775	pasear *VMN0000* 1	por *SP* 1	el *DA0MS0* 1	delta_de_el_tigre *NP00G00* 1	. *Fp* 1
vamos *yo* 0.03207			a *NCFS000* 0.0012246		a *NCFS000* 0.0012246					
ir *VMM01P0* 0.00291545										

Distancia de Mínima Edición

Distancia de Mínima Edición

- Encontrar una noción de distancia entre palabras
- Encontrar la palabra más "próxima"
- Por ejemplo, para autocorrección
  - Francia/Croacia
  - cena/pena/cana/ana
  - inelefantemente/indefectiblemente
- Entre todas las posibles palabras del diccionario, sugerir la más cercana

Distancia de Mínima Edición

- DME (Minimum Edit Distance):

  mínimo número de operaciones de edición (inserción, borrado, sustitución) necesarias para transformar un string en otro

  Método: Programación dinámica

Distancia de Mínima Edición

- En el método original, cada operación de edición tiene costo 1
- En 1966 Levenshtein propone operaciones con distinto costo
    - inserción = 1
    - borrado = 1
    - sustitución = 2
- Es importante para obtener ese mínimo, que las palabras estén lo más “alineadas” posible

Distancia de Mínima Edición

Ejemplos intuitivos:

C R O A C I A

F R A N C I A

s s s DME=6

I T A L *I A

F R A N C I A

s s si DME = 7

F R A N C I A

I T A L *I A

s s s b DME = 7

Distancia de Mínima Edición

- Sean X → palabra$_1$ de largo n

  Y → palabra$_2$ de largo m

- Se define una matriz **d** (n+1 x m+1)

- Se inicializan la fila y columna 0
  - d(0,0) = 0
  - para i de 1 .. n d(i,0) = i
  - para j de 1 .. m d(0,j) = j

n = 4 m = 3

0	1	2	3	4
1				
2				
3				

Distancia de Mínima Edición

Algoritmo:

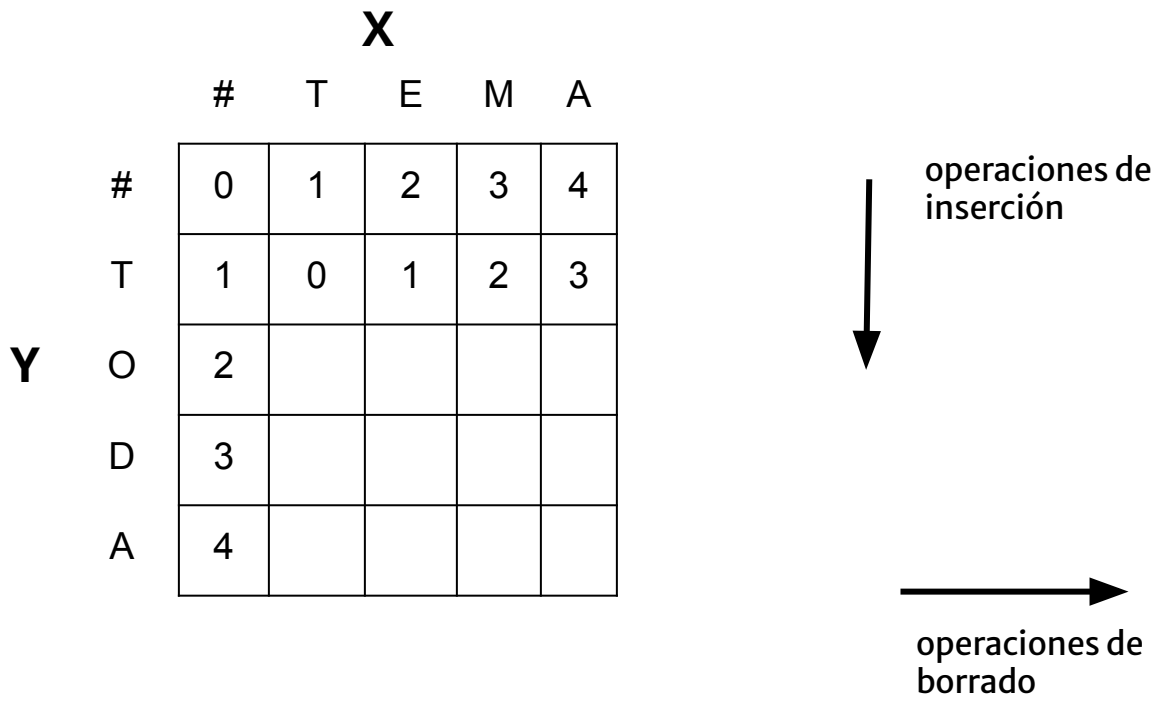
para cada i de 1 .. n

para cada j de 1 .. m

$$d(i,j) = \min \begin{cases} d(i-1, j) + 1 \\ d(i, j-1) + 1 \\ d(i-1,j-1) + \begin{cases} 0 & \text{si } X(i) = Y(j) \\ & \text{ó} \\ 2 & \text{si } X(i) \mathrel{!}= Y(j) \end{cases} \end{cases}$$

Distancia de Mínima Edición

Ejemplo: calcular la DME entre TEMA y TODA (n = 4 m = 4)



Distancia de Mínima Edición

Ejemplo: calcular la DME entre TEMA y TODA

	#	T	E	M	A
#	0	1	2	3	4
T	1	0	1	2	3
O	2	1	2	3	4
D	3	2	3		
A	4				

Distancia de Mínima Edición

Ejemplo: calcular la DME entre TEMA y TODA

	#	T	E	M	A
#	0	1	2	3	4
T	1	0	1	2	3
O	2	1	2	3	4
D	3	2	3	4	5
A	4	3	4	5	**4**

DME(x,y)

Distancia de Mínima Edición

Ejemplo: calcular la DME entre TEMA y TODA

	#	T	E	M	A
#	**0**	1	2	3	4
T	**1**	**0**	1	2	3
O	2	1	**2**	3	4
D	3	2	3	**4**	5
A	4	3	4	5	**4**

→ DME(x,y)

Detección de errores

- Detección de palabras inexistentes (*tmate*)
- Corrección aislada (*tmate* -> *tomate*)
- Detección y corrección dependiente del contexto

  (*calor* -> *color*)

Detección de errores

Pueden deberse a:

- Inserción (*toomate*)
- Borrado (*tmate*)
- Sustitución (*tpmate*)
- Trasposición (*tmoate*)

Estudio (Kernighan, 1990)

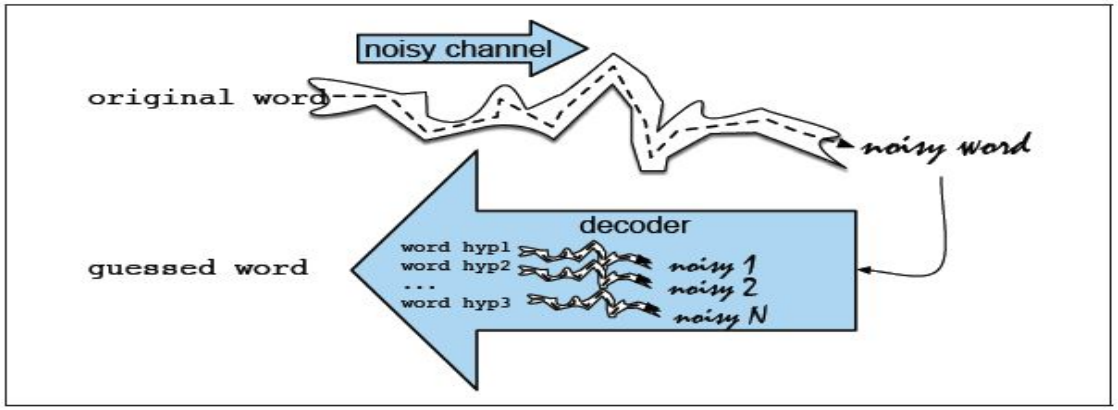
1% a 3% palabras con errores

de estos, el 80% eran por borrado o inserción

Detección de errores

Detección de palabras inexistentes → corrección

- Diccionario
- Método del Canal Ruidoso (probabilístico)



$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w|O)$$

Modelos probabilísticos

Regla de Bayes

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} \frac{P(O|w)P(w)}{P(O)}$$

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} \frac{P(O|w)P(w)}{P(O)} = \underset{w \in V}{\operatorname{argmax}} P(O|w)\,P(w)$$

Corrección de errores

Algoritmo bayesiano (Kernighan – 1990)

- Hipótesis: errores son por inserción, borrado, sustitución y transposición
- Aplico todas las transformaciones posibles a la palabra observada y busco lista de candidatos válidos considerando la DME

		Transformation			
Error	Correction	Correct Letter	Error Letter	Position (Letter #)	Type
acress	actress	t	—	2	deletion
acress	cress	—	a	0	insertion
acress	caress	ca	ac	0	transposition
acress	access	c	r	2	substitution
acress	across	o	e	3	substitution
acress	acres	—	s	5	insertion
acress	acres	—	s	4	insertion

Distancia de Damerau–Levenshtein

Corrección de errores

$$\hat{w} = \underset{w \in C}{\operatorname{argmax}} \overbrace{P(x|w)}^{\text{channel model}} \overbrace{P(w)}^{\text{prior}}$$

w	count(w)	p(w)
actress	9,321	.0000231
cress	220	.000000544
caress	686	.00000170
access	37,038	.0000916
across	120,844	.000299
acres	12,874	.0000318

$$P(w) = \frac{Count(w) + 0{,}5}{N + 0{,}5V}$$

probabilidad “a priori” que aparezca la palabra en el corpus

Modelos probabilísticos

Tenemos P(w), que es la probabilidad a priori

Pero… cómo calculamos P(x|w)?

- En un corpus de errores, ¿cuántas veces se sustituye?
- Matriz de confusión que contiene las cantidades de ocurrencias en que una cosa se confundió con otra

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|w)
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 que se borre una "**t**" después de una "**c**"

Modelos probabilísticos

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|w)	P(w)	10^9*P(x|w)P(w)
actress	t	-	c|ct	.000117	.0000231	2.7
cress	-	a	a|#	.00000144	.000000544	0.00078
caress	ca	ac	ac|ca	.00000164	.00000170	0.0028
access	c	r	r|c	.000000209	.0000916	0.019
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

Modelos probabilísticos

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|w)	P(w)	10^9*P(x|w)P(w)
actress	t	-	c|ct	.000117	.0000231	2.7
cress	-	a	a|#	.00000144	.000000544	0.00078
caress	ca	ac	ac|ca	.00000164	.00000170	0.0028
access	c	r	r|c	.000000209	.0000916	0.019
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

Corrección de errores

...was called a "stellar and versatile **acress** *whose combination of sass and glamour has defined her ....*

Corrección de errores

...was called a "stellar and versatile **acress** *whose combination of sass and glamour has defined her ....*

daría

...was called a "stellar and versatile **across** *whose combination of sass and glamour has defined her ....*

Corrección de errores

...was called a "stellar and versatile **acress** *whose combination of sass and glamour has defined her ....*

daría

...was called a "stellar and versatile **across** *whose combination of sass and glamour has defined her ....*

debiera dar

...was called a "stellar and versatile **actress** *whose combination of sass and glamour has defined her ....*