

Pre-procesamiento de texto

Curso de procesamiento de lenguaje natural

Julio Weissman

Maestría en Tecnologías de la Información
Universidad Nacional de Misiones

9 de marzo de 2018



Alguna definiciones

- Un documento es una cadena de caracteres
 - Un tweet
 - Un libro
 - Un correo electrónico
- Un *corpus* es una colección de documentos (congruentes entre si)
- Los documentos pueden dividirse en párrafos, frases y/o palabras
 - Pueden estar anotados
 - Pueden estar asignados a una categoría
- Un vocabulario es un conjunto cerrado de palabras de dimensión V

- 1 Normalmente la información en forma de texto en lenguaje natural viene en formatos complicados y trae muchos *artefactos* que influyen de manera muy importante en el resultado de una tarea de PLN.
- 2 Se requiere manejo de información en formatos `json` o `xml`, así como el manejo de bases de datos.
- 3 El acondicionamiento del texto es fundamental (capitalización, minúsculas, corrección, eliminación de argot ...)

- 1 Las expresiones regulares (*regex*) juegan un rol *sorprendentemente* importante en PLN
- 2 El primer paso en una tarea de PLN seguido implica el uso de *regex* sofisticadas
- 3 Son difíciles de corregir
- 4 Si quieres practicar, puedes consultar [este tutorial de *regex* en *python*](#)

Algunas expresiones regulares

Disyunciones

Pattern	Matches
<code>[wW]oodchuck</code>	Woodchuck, woodchuck
<code>[1234567890]</code>	Any digit

Rangos

Pattern	Matches	
<code>[A-Z]</code>	An upper case letter	<u>D</u> renched Blossoms
<code>[a-z]</code>	A lower case letter	<u>my</u> beans were impatient
<code>[0-9]</code>	A single digit	Chapter <u>1</u> : Down the Rabbit Hole

Algunas expresiones regulares

Negaciones

Pattern	Matches	
[^A-Z]	Not an upper case letter	Oyfn pripetchik
[^Ss]	Neither 'S' nor 's'	I have no exquisite reason"
[^e^]	Neither e nor ^	Look here
a^b	The pattern a carat b	Look up a^b now

Disyunciones con |

Pattern	Matches
groundhog woodchuck	
yours mine	yours mine
a b c	= [abc]
[gG]roundhog [Ww]oodchuck	

Algunas expresiones regulares

* + ? .

Pattern	Matches	
colou?r	Optional previous char	<u>color</u> <u>colour</u>
oo*h!	0 or more of previous char	<u>oh!</u> <u>ooh!</u> <u>oooh!</u> <u>ooooh!</u>
o+h!	1 or more of previous char	<u>oh!</u> <u>ooh!</u> <u>oooh!</u> <u>ooooh!</u>
baa+		<u>baa</u> <u>baaa</u> <u>baaaa</u> <u>baaaaa</u>
beg.n		<u>begin</u> <u>begun</u> <u>begun</u> <u>beg3n</u>

Inicio y fin de cadena

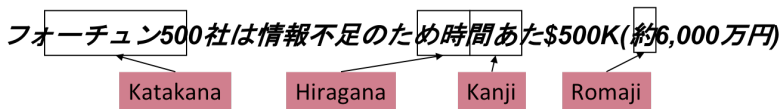
Pattern	Matches
^[A-Z]	<u>P</u> alo Alto
^[^A-Za-z]	<u>1</u> <u>"Hello"</u>
\. \$	The end <u>.</u>
. \$	The end <u>?</u> The end <u>!</u>



Veamos unos ejemplo no tan triviales

¿Que es una palabra?

- Una secuencia de caracteres con significado
- En español, la puntuación y los caracteres sirven para separar palabras.
- ¿Y los neologismos, los *hashtags*, las direcciones url, las fechas,...?
- ¿La puntuación debe de integrarse en la palabra?



Token

Unidad útil de caracteres para el procesamiento semántico. Puede ser palabras, frases, símbolos, etc.

Tokenización

Proceso de separar un documento en tokens.

- 1 El método más popular en inglés es el [Treebank tokenization](#)
- 2 En español el proceso de tokenización es relativamente simple, si no se trata con documentos especializados
- 3 En idiomas como alemán, turco, japonés o chino, la tokenización es un problema difícil

Veamos unos ejemplos



Regresamos a la libreta

Normalización de *tokens*

- Algunos métodos basados en reglas para normalizar tokens con mismo significado
 - Minúsculas en todas las palabras
 - Manejo de acrónimos (EUA, E.U.A., US, U.S., USA, U.S.A.)
 - Requiere de desarrollar muchas reglas particulares
- Encontrar el mismo token para diferentes formas con el mismo significado semántico (niños, niño, niña, niñas, . . .)
 - Basado en análisis morfológico (lematización)
 - Basado en reglas heurísticas (*stemming*)

- Proceso de remover y remplazar sufijos de una palabra en varios pasos a fin de obtener la *raíz* de la palabra (llamada tallo, o *stem*)
- Método heurístico para ir recortando las palabras
- Típicamente falla en formas irregulares
- El método más típico para inglés es el [Porter's stemming algorithm](#)
- El método más típico en español es el [Snowball spanish stemming algorithm](#)

Lematización

- Se refiere al uso de un análisis lingüístico formal basado en un vocabulario cerrado y en reglas morfológicas.
- Difícil de hacer, requiere de mucho esfuerzo.
- Convierte el token a la forma base de la palabra, tal y como esté definido en un diccionario de referencia (lexema).
- Existen pocos lematizadores en español ([freeling](#), spacy).

