Procesamiento de Lenguaje Natural

Procesamiento Básico

Mauricio Toledo-Acosta
mauricio.toledo@unison.mx

Departamento de Matemáticas
Universidad de Sonora

Section 1

Expresiones regulares

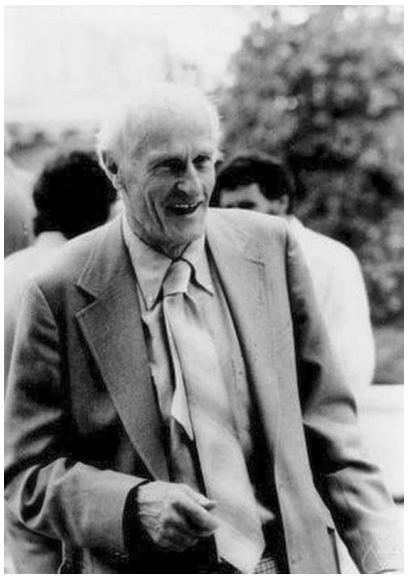
Expresiones regulares

Una **expresión regular** (regex, expresión racional) es una secuencia de caracteres que especifica un patrón de coincidencia en un texto. Los algoritmos de búsqueda de cadenas suelen utilizar este tipo de patrones para realizar operaciones de "búsqueda" o "búsqueda y sustitución" de cadenas, o para validar entradas.

Las expresiones regulares constan de constantes (denotan conjuntos de cadenas) y símbolos de operaciones (denotan operaciones sobre estos conjuntos).

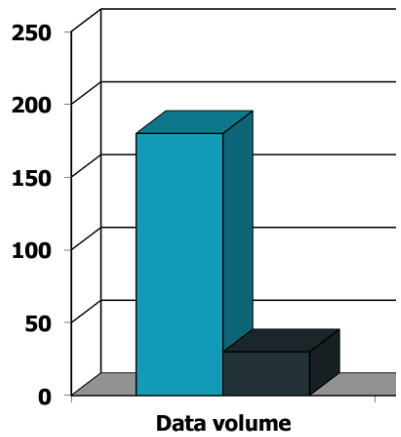
Un poco de historia

Se originaron en 1951, por Stephen Cole Kleene. Usualmente se usa el standard `IEEE POSIX`. Kleene es uno de los fundadores de las ciencias computacionales teóricas.



Utilidad de las expresiones regulares

- Validación de datos.
- Búsqueda, extracción y reemplazo de texto.
- División de Texto
- Transformación de Texto.
- Tareas de PLN (eliminación de stopwords).



Referencias adicionales

- https://regex101.com/
- https://www.programiz.com/python-programming/regex
- https://www3.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html

Tutorial: Metacaracteres

Los metacaracteres son caracteres que un motor RegEx interpreta de forma especial

```
[] . ^ $ * + ? {} () \ |
```

[ ]	Cualquier caracter dentro de los corchetes
.	Cualquier caracter (excepto cambios de línea)
^	Buscar si el caracter siguiente está al inicio de una línea
[^]	Negación de cualquier caracter dentro de los corchetes
$	Buscar si el caracter anterior está al final de una línea

Si queremos buscar los metacaracteres como caracteres se anteceden de un \.

Tutorial: Metacaracteres

[ ] . ^ $ * + ? { } ( ) \ |

*	Busca si el caracter anterior ocurre 0 o más veces
+	Busca si el caracter anterior ocurre 1 o más veces
?	Busca si el caracter anterior ocurre 0 o 1 vez
$\{m, n\}$	*OR*, busca si el caracter anterior ocurre al menos m veces y máximo n veces
|	Busca el caracter antes o después del |
()	Agrupar patrones (expresiones).
\1	*Backreference*, captura el patrón anterior repetido consecutivamente.

Tutorial: Sequencias especiales

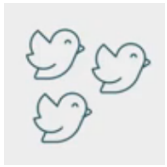
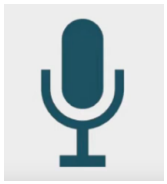
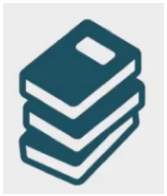
A	Inicio de la string
b	Frontera de palabra
w	Cualquier *word character*
W	Cualquier *non word character*
d	Cualquier dígito
D	Cualquier no dígito

Section 2

Procesamiento básico de texto

Corpus

Un corpus es una colección de textos que se utilizará para alguna tarea de NLP.

Algunos ejemplos de corpus:

- 20newsgroups
- IMDB
- Project Gutenberg
- OntoNotes 5
- Penn Treebank

Técnicas de preprocesamiento

En cualquier aplicación de NLP el preprocesamiento de texto es el primer paso para cualquier técnica de modelado.

- Tokenización
- Lematización
- Stop words removal
- Etiquetado POS
- Etiquetado NER
- Análisis de dependecias

Tokenización (Tokenization)

La **tokenización** es el proceso de dividir un texto en unidades más pequeñas llamadas tokens. Estos tokens pueden ser palabras, caracteres, símbolos o frases. La tokenización es un paso fundamental en el procesamiento del texto.

Lematización

La **lematización** es el proceso de reducir una palabra a su forma base (lema). Se utiliza para:

- Reducir la dimensionalidad del espacio de características, al mapear palabras relacionadas a un solo lema.
- Mejorar la precisión de los modelos de lenguaje, al tratar palabras con el mismo significado como una sola entidad.
- Facilitar la comparación y el análisis de textos, al estandarizar la forma de las palabras.

$$\text{Correr, corre, corriendo, corredor} \longrightarrow \texttt{correr}$$

$$\text{Feliz, felicidad, felices} \longrightarrow \texttt{feliz}$$

Dificultades en la tokenización


```
Finland's capital    →  Finland Finlands Finland's ?
what're, I'm, isn't  →  What are, I am, is not
Hewlett-Packard      →  Hewlett Packard ?
state-of-the-art     →  state of the art ?
Lowercase            →  lower-case lowercase lower case ?
San Francisco        →  one token or two?
m.p.h., PhD.         →  ??
```

Dificultades en la tokenización

French

- ***L'ensemble*** → one token or two?
  - ***L*** ? ***L'*** ? ***Le*** ?
  - Want ***l'ensemble*** to match with ***un ensemble***

German noun compounds are not segmented

- ***Lebensversicherungsgesellschaftsangestellter***
- 'life insurance company employee'
- German information retrieval needs **compound splitter**

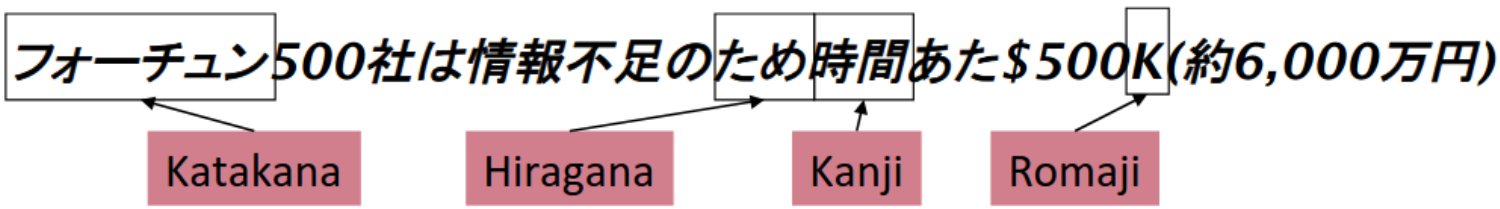
Dificultades en la tokenización

Chinese and Japanese no spaces between words:

- 莎拉波娃现在居住在美国东南部的佛罗里达。
- 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- Sharapova now lives in US southeastern Florida

Further complicated in Japanese, with multiple alphabets intermingled

- Dates/amounts in multiple formats



Stemming

El **stemming** es el proceso de reducir las palabras a su raíz o tronco, eliminando sufijos y prefijos. El objetivo es identificar la forma base de una palabra, independientemente de su conjugación, número o género. Se utiliza para:

- Reducir la dimensionalidad del espacio de características en tareas de clasificación de texto.
- Mejorar la eficiencia en la indexación de texto.
- Facilitar la búsqueda de información.

$$\text{Correr, corre, corriendo, corredor} \longrightarrow \texttt{corri}$$

El stemming puede ser más rápido, aunque menos preciso, que la lematización. Además de producir palabras posiblemente no validas.

Etiquetado POS y Parsing

El **POS Tagging (Part-of-Speech Tagging)** es el proceso de identificar la categoría gramatical de cada palabra en un texto, como: Sustantivo (NOUN), verbo (VERB), adjetivo (ADJ), etc. El objetivo del POS Tagging es etiquetar cada palabra con su correspondiente categoría gramatical, lo que permite comprender mejor el significado y la estructura del texto.

El **parsing**, también conocido como análisis sintáctico, es el proceso de analizar una secuencia de tokens para determinar su estructura gramatical. En otras palabras, es el proceso de identificar las relaciones entre las palabras o símbolos en una secuencia para entender su significado.

Entender, Semántica...

En cada idioma

https://universaldependencies.org/

Named Entity Recognition

NER (Named Entity Recognition) es el proceso de identificar y clasificar entidades nombradas en un texto en categorías predefinidas como:

- Nombres de personas (PER)
- Nombres de lugares (LOC)
- Nombres de organizaciones (ORG)
- Fechas (DATE)
- Monedas (MONEY)
- ...

El objetivo de NER es extraer información relevante de un texto y clasificarla en categorías significativas para su posterior análisis o procesamiento.

Nubes de palabras

Es un técnica exploratoria que nos permite visualizar información sobre la frecuencia de las palabras en un texto.

¿De qué trata el texto anterior?