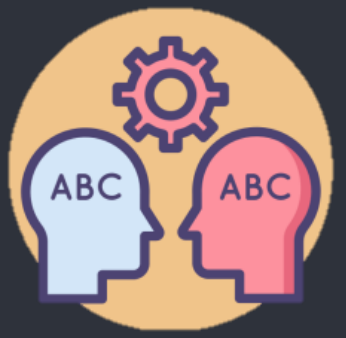
Procesamiento de Lenguaje Natural{



[NLP]

<Preprocesamiento de Texto>

}

Ivan Rojas Gonzalez – Universidad Panamericana

La Agenda de hoy {

01 Preprocesamiento de texto

<Por que se necesita?>

02 Tokenizar, lematizar, stemmizar

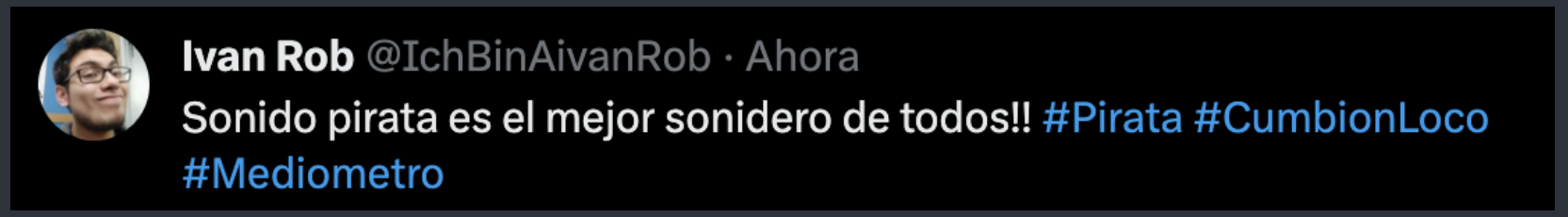
<Unificando el formato de nuestro texto>

03 Todo en una misma bolsa

<Pipelines de preprocesamiento>

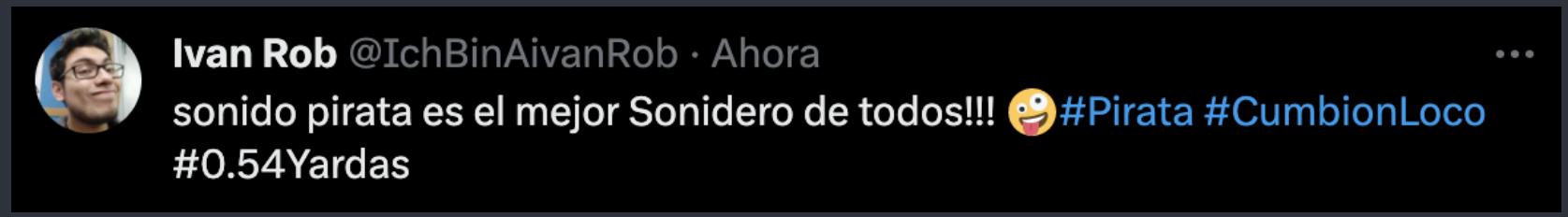
}

Preprocesamiento de Texto{



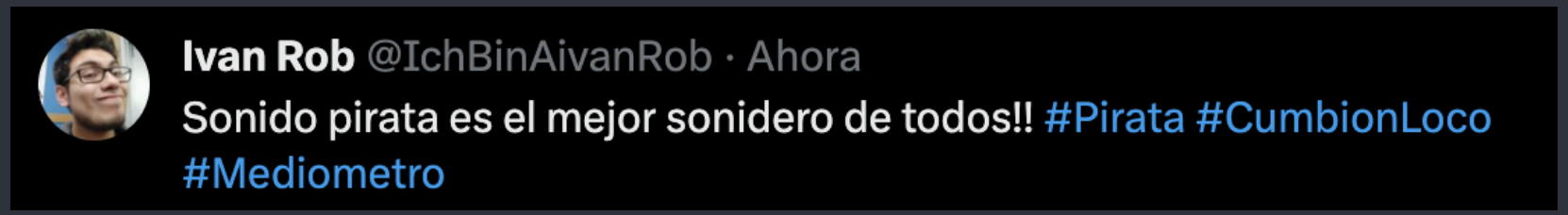
=

?



}

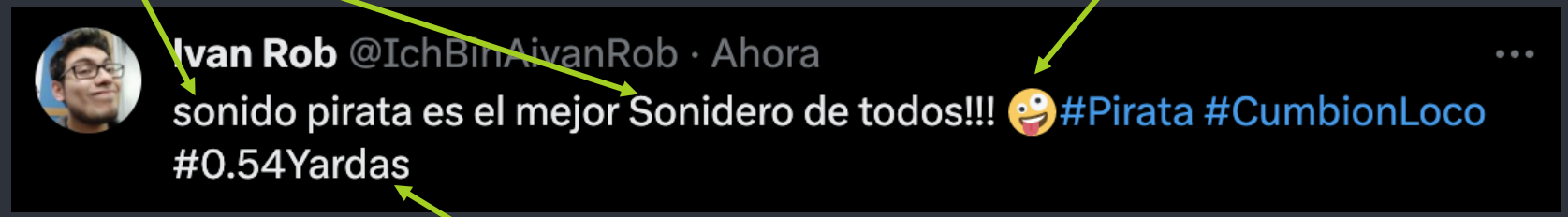
Preprocesamiento de Texto{



Mayusculas y minusculas

≠

Presencia de emojis

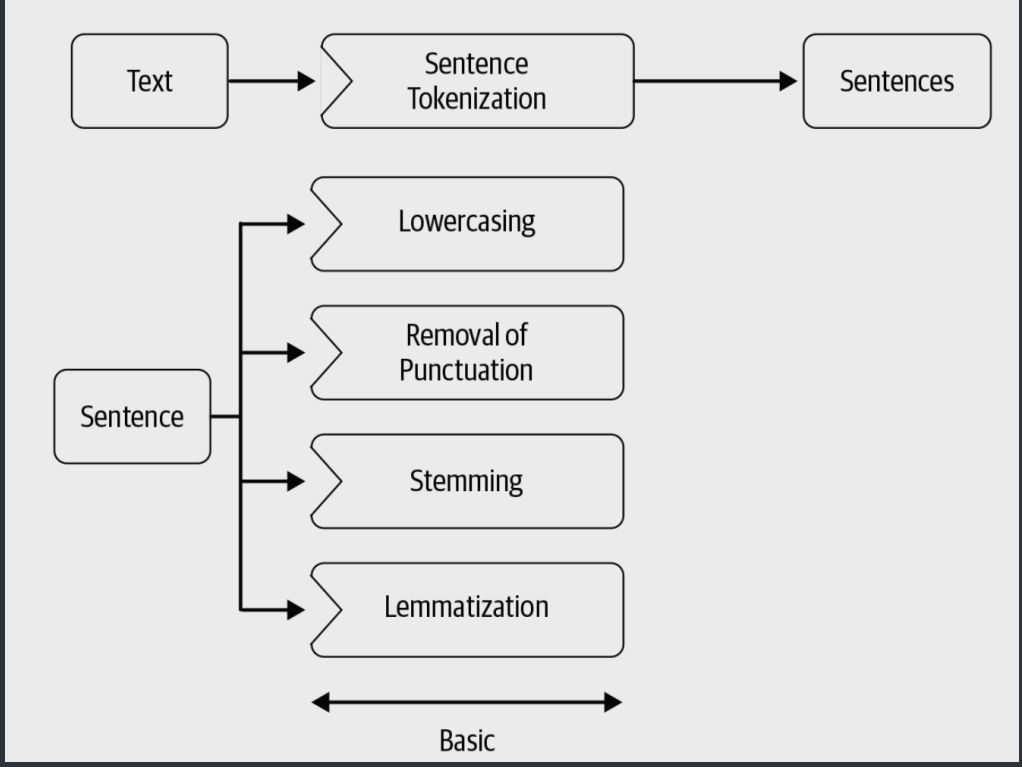


Diferentes hashtags

}

Preprocesamiento de Texto{

<Limpiar el texto con el que estaremos trabajando para que el análisis/modelado dé los mejores resultados posibles. >



}

{Lowercasing}

Lowercasing{

< Convierte todos los caracteres de un string a letras minusculas>

```
nombre = "EL PRINCIPE DEL TACO"
nombre_minus = nombre.lower()
print(nombre_minus)
```

```
el principe del taco
```

}

{Tokenizacion}

Ivan Rojas Gonzalez – Universidad Panamericana

Tokenizacion{

<Dividir una oración en las palabras que la componen. >

```
texto = '''Mi Homero no es comunista. Podrá ser
          mentiroso, puerco, idiota, comunista,
          pero nunca una estrella de porno'''
```

Unigramas

```
texto.split()
✓ 0.1s

['Mi',
 'Homero',
 'no',
 'es',
 'comunista.',
 'Podrá',
 'ser',
 'mentiroso,',
 'puerco,',
 'idiota,',
 'comunista,',
 'pero',
 'nunca',
 'una',
 'estrella',
 'de',
 'porno']
```

Tokenizacion{

<Dividir una oración en las palabras que la componen. >

```
import nltk
nltk.download('punkt')
sentence = 'Mi hijo no es comunista podrá ser tonto, estúpido, inútil, comunista pero nunca una estrella porno'
tokens = nltk.word_tokenize(sentence)
print(tokens)
```

✓ 0.0s

```
['Mi', 'hijo', 'no', 'es', 'comunista', 'podrá', 'ser', 'tonto', ',', 'estúpido', ',', 'inútil', ',', 'comunista', 'pero', 'nunca', 'una', 'estrella', 'porno']
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\ivani\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Las comas y puntuaciones se toman como su propio token

}

{Stopwords}

Stopwords{

< Palabras que no aportan significado o contexto a la oracion>

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words_en = stopwords.words('English')
stop_words_es = stopwords.words('Spanish')
```

```
print(stop_words_en)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she
```

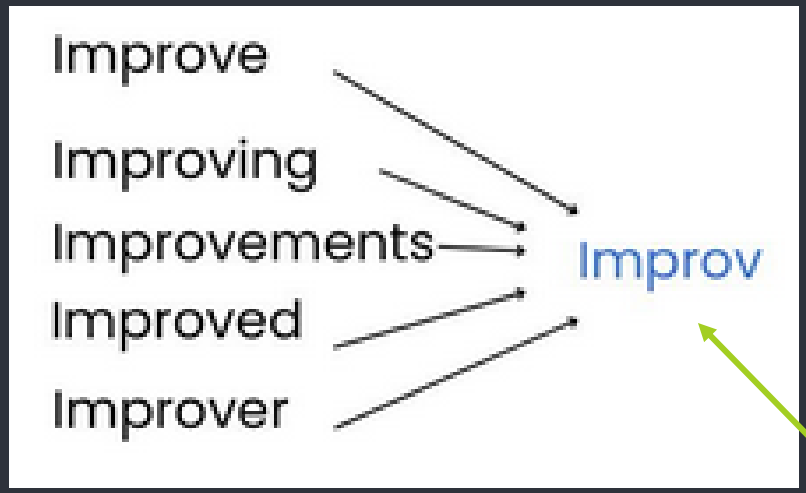
+ Code + Markdown

```
print(stop_words_es)
```

```
['de', 'la', 'que', 'el', 'en', 'y', 'a', 'los', 'del', 'se', 'las', 'por', 'un', 'para', 'con', 'no', 'una', 'su', 'al', 'lo', 'como', 'más', 'pero', 'sus', 'le', 'ya', 'o', 'este', 'sí', 'por
```

Stemming{

< Reducir palabras que comparten el mismo significado a su raiz.
Es simple y rapido>



Las raices no siempre son palabras que si existan

}

Stemming{

< Reducir palabras que comparten el mismo significado a una base>

```
import nltk
stemmer = nltk.stem.PorterStemmer()
print(stemmer.stem("improve"))
print(stemmer.stem("improving"))
print(stemmer.stem("improvement"))
print(stemmer.stem("improved"))
print(stemmer.stem("improver"))
```

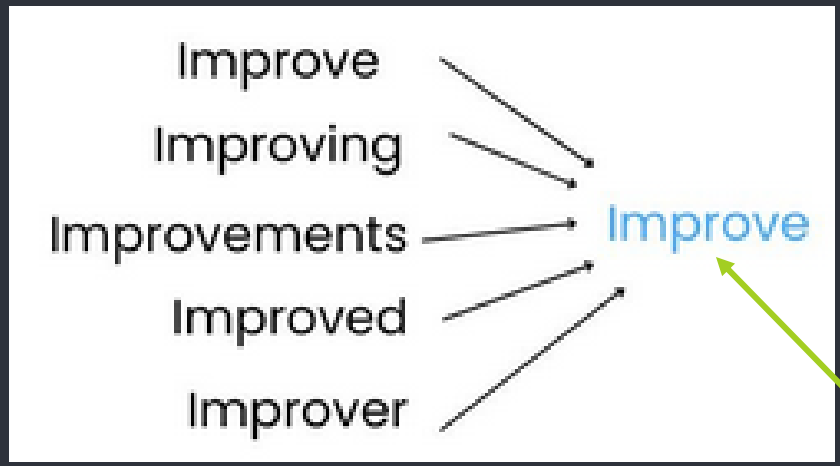
✓ 0.0s

```
improv
improv
improv
improv
improv
```

}

Lemmetization{

< Reducir palabras que comparten el mismo significado a un lemma, mas lento puesto que implica hacer lookups a vocabulario ya establecido>



Los lemmas son palabras que si existen

}

Lemmetization{

< Reducir palabras que comparten el mismo significado a un lemma, mas lento puesto que implica hacer lookups a vocabulario ya establecido>

```python
import nltk
nltk.download('wordnet')
from nltk.stem.wordnet import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
print(lemmatizer.lemmatize("improve"))
print(lemmatizer.lemmatize("improving"))
print(lemmatizer.lemmatize("improvement"))
print(lemmatizer.lemmatize("improved"))
print(lemmatizer.lemmatize("improver"))
```

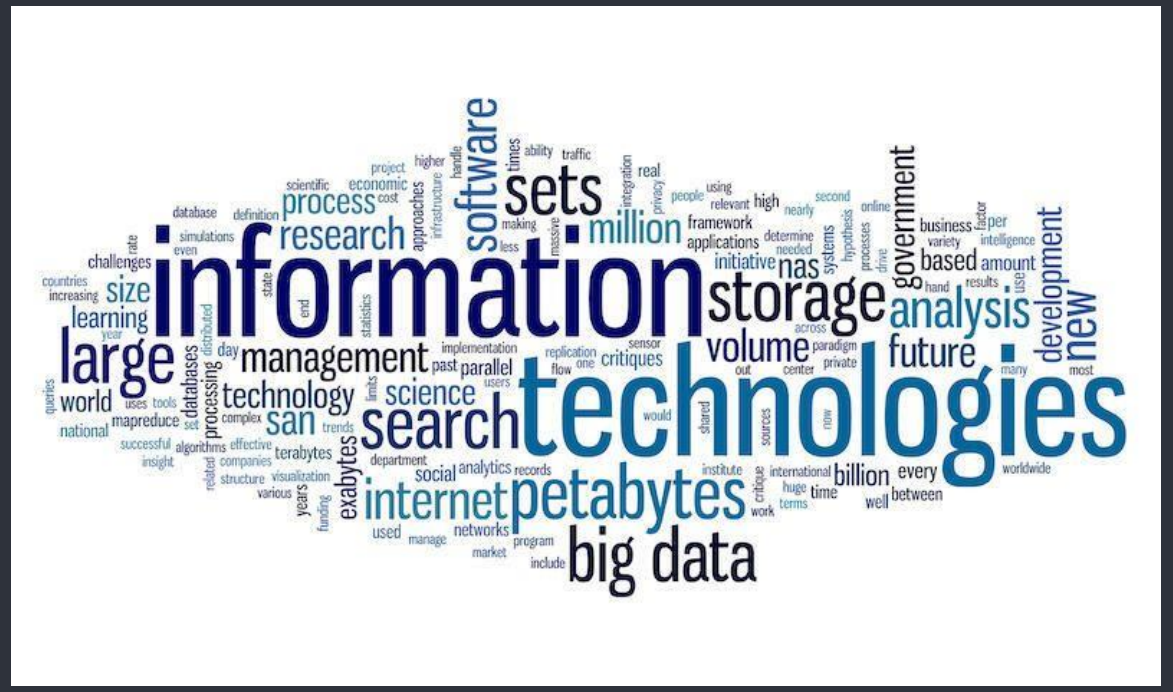
✓ 0.0s

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\ivani\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
improve
improving
improvement
improved
improver
```

}

Wordclouds{

<Herramienta util para visualizar terminos frecuentes y darse una idea general de el contexto del corpus analizado>



}

Wordclouds{

<Herramienta util para visualizar terminos frecuentes y darse una idea general de el contexto del corpus analizado>

```python
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
stopwords = set(STOPWORDS)
wordcloud = WordCloud(width = 1200, height = 1200,
                      colormap = 'YlOrRd',                # font colors from palette
                      background_color ='black',          # background color
                      stopwords = stopwords,              # remove common words
                      #normalize_plurals = True,           # plurals to singular
                      collocations = False,               # no repetition
                      min_font_size = 6).generate(final_string)


#  3.  PLOT WORD CLOUD
#------------------------------
plt.figure(figsize = (12, 12))
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout()
plt.show()
```

Q&A