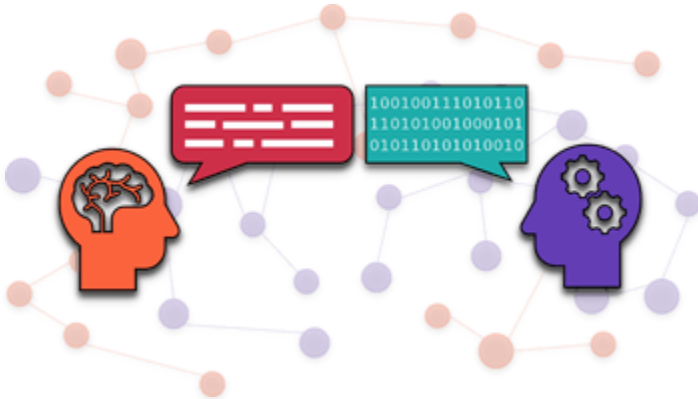
Modelos de Inteligencia Artificial

CIPFP Mislata
Centre Integrat Públic Formació Professional Superior

:: U3 ::

Procesamiento del Lenguaje Natural



1. Preprocesamiento de texto

Curso 2023-24

Tabla de contenidos

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

1. Introducción

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- El **preprocesamiento de texto** es un enfoque para limpiar y preparar datos de texto para su uso en un contexto específico
- Motivación: los datos tienen diferentes orígenes y formatos
- Existen diferentes técnicas para el procesado de textos, pudiendo usar unas y otras en función del objetivo que se persiga
- En Python diversas librerías facilitan esta tarea (NLTK, SpaCy, Stanza, ...)

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

a) Eliminación de ruido:

- Supone eliminar información no deseada:
  - Caracteres especiales
  - Dígitos numéricos
  - Espacios en blanco inicial, final y vertical
  - Formato HTML
  - Puntuación y acentos

- Se puede realizar usando la biblioteca de Python **Regex**

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

a) Eliminación de ruido:

- Con Regex …
  - se importa “**re**”
  - uso del método “**sub**”
    - patrón
    - texto de sustitución
    - texto de entrada

```
import re

text = "<p> Esto es un párrafo </p>"
result = re.sub(r'<.?p>', '', text)

print (result)
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

a) Eliminación de ruido:

- en este ejemplo se suprimen los espacio en blanco con los que empieza la frase
- ampliaremos esta parte en el próximo tema ...

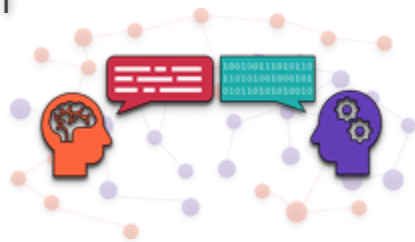
```python
import re

text = "    Esto es un párrafo"
result = re.sub(r'\s{4}', '', text)

print (result)
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

b) Tokenización

- El método para dividir el texto en componentes más pequeños se llama tokenización y los componentes individuales se llaman **tokens**.
- operaciones comunes que requieren tokenización:
  - encontrar cuántas palabras u oraciones aparecen en el texto
  - determinar cuántas veces existe una palabra o frase específica
  - tener en cuenta qué términos es probable que coexistan



2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

b) Tokenización

- En **SpaCy**, supone crear un objeto que contendrá el texto vectorizado, y por tanto, tokenizable
- Cada token tiene una serie de propiedades (por ejemplo, la palabra a la que se refiere)

```python
import spacy

# Cargar el modelo de idioma en español
nlp = spacy.load('es_core_news_sm')

# Texto que quieres tokenizar
texto = "Spacy es una excelente herramienta \
para procesamiento de lenguaje natural en español."

# Aplicar el modelo para obtener los tokens
doc = nlp(texto)

# Imprimir los tokens
for token in doc:
    print(token.text)
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

b) Tokenización

- a nivel de oración se puede tokenizar usando la propiedad "sents" del objeto "doc".

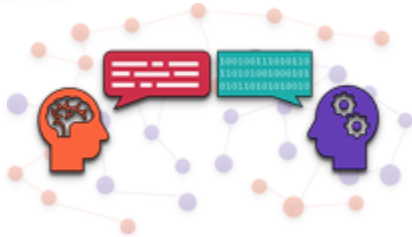
```python
import spacy

# Cargar el modelo de idioma en español
nlp = spacy.load('es_core_news_sm')

# Texto que quieres tokenizar
texto = "Spacy es una excelente herramienta \
para procesamiento de lenguaje natural en español.\
Puede tokenizar a nivel de oración también."

# Aplicar el modelo para obtener el documento
doc = nlp(texto)

# Imprimir las oraciones
for oracion in doc.sents:
    print(oracion.text)
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

c) Normalización

- Aporta técnicas adicionales para el preprocesamiento de texto
- Ejemplos
  - conversión mayúsculas / minúsculas
  - eliminación de palabras irrelevantes
  - derivación (“Stemming”) → eliminación de prefijos y sufijos de una palabra
  - lematización → sustitución de un token de una sola palabra por su raíz



2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

→ mayúsculas o minúsculas

- se pueden usar los métodos “upper” y “lower” del objeto “string”

```
cadena = 'eStO eS unA cAdeNA dE cAracTeREs'

print(cadena.upper())
# 'ESTO ES UNA CADENA DE CARACTERES'

print(cadena.lower())
# 'esto es una cadena de caracteres'
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

→ eliminación de palabras irrelevantes

- La propiedad "is_stop" de cada token permita detectar si la palabra es una "stopword".

```
# LIMPIEZA DE TEXTO
import spacy

# Texto que quieres procesar
texto = "Spacy es una excelente herramienta \
para procesamiento de lenguaje natural en español."

# Aplicar el modelo para obtener el documento
doc = nlp(texto)

# Filtrar las stopwords y imprimir las palabras restantes
palabras_filtradas = [token.text for token in doc if not token.is_stop]
print("Palabras sin stopwords:", palabras_filtradas)
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

→ derivación (“stemming”)

- La derivación es la tarea de normalización de preprocesamiento del texto que se ocupa de eliminar los afijos de las palabras (prefijos y sufijos).
- SpaCy carece de esta función, pero mediante la librería puede realizarse **NTLK**.

```
# DERIVACIÓN
import spacy
import nltk
from nltk.stem import PorterStemmer

# Descargar los recursos necesarios para NLTK
nltk.download('punkt')

# Cargar el modelo de idioma en español de Spacy
nlp = spacy.load('es_core_news_sm')

# Crear un objeto PorterStemmer de NLTK
stemmer = PorterStemmer()

# Texto que quieres procesar
texto = "Spacy es una excelente herramienta \
para procesamiento de lenguaje natural en español."

# Aplicar el modelo de Spacy para obtener el documento
doc = nlp(texto)

# Obtener las palabras lematizadas con Spacy y realizar "stemming" con NLTK
palabras_stemming = [stemmer.stem(token.lemma_) for token in doc]

print("Palabras lematizadas y stemmizadas:", palabras_stemming)
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

→ lematización

- La lematización es un método de conversión de palabras a su forma raíz.
- Cada token en SpaCy contiene información sobre su lema.

```python
import spacy

# Cargar el modelo de idioma en español
nlp = spacy.load('es_core_news_sm')

# Texto que quieres procesar
texto = "Spacy es una excelente herramienta \
para procesamiento de lenguaje natural en español."

# Aplicar el modelo para obtener el documento
doc = nlp(texto)

# Imprimir las palabras lematizadas
for token in doc:
    print(f"Palabra: {token.text}, Lema: {token.lemma_}")
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

```
Palabra:  Spacy          Lema:  Spacy
Palabra:  es             Lema:  ser
Palabra:  una            Lema:  uno
Palabra:  excelente      Lema:  excelente
Palabra:  herramienta    Lema:  herramienta
Palabra:  para           Lema:  para
Palabra:  procesamiento  Lema:  procesamiento
Palabra:  de             Lema:  de
Palabra:  lenguaje       Lema:  lenguaje
Palabra:  natural        Lema:  natural
Palabra:  en             Lema:  en
Palabra:  español        Lema:  español
Palabra:  .              Lema:  .
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

→ etiquetado morfológico (“Part-of-Speech Tagging”)

- Para mejorar el rendimiento de la lematización, necesitamos localizar cada palabra en el lugar que ocupa en nuestra cadena de texto.
- Ciertas librerías como “spaCy” incorporan funciones para la identificación de tipos de términos (sustantivos, adjetivos, verbos, adverbios, …)

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

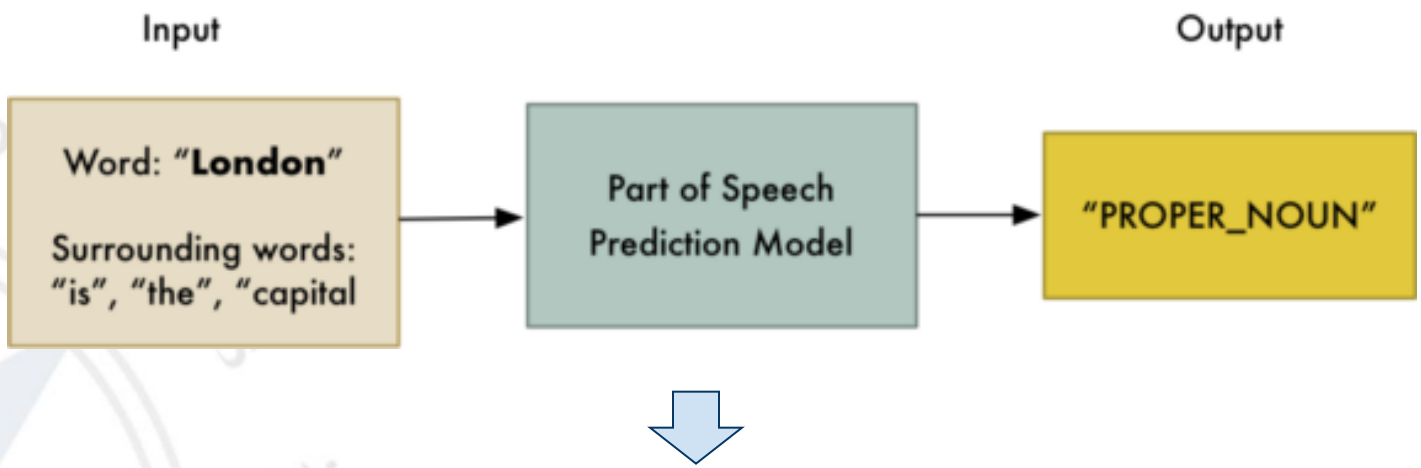
¿Cómo se puede saber qué función gramatical desempeña cierto término?

¿Cómo se puede abordar técnicamente el etiquetado morfológico?

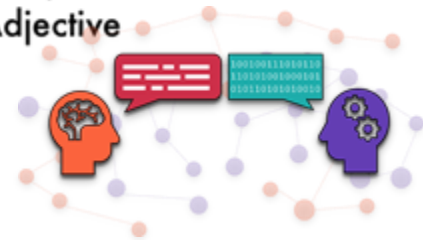
2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

→ etiquetado morfológico (“Part-of-Speech Tagging”)



London	is	the	capital	and	most	populous ...
Proper Noun	Verb	Determiner	Noun	Conjunction	Adverb	Adjective



2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

→ etiquetado morfológico (“Part-of-Speech Tagging”)

- Ejemplo: script que realiza el etiquetado de cierta frase

```
# ETIQUETADO DEL DISCURSO
import spacy

# Texto que quieres procesar
texto = "Spacy es una excelente herramienta \
para procesamiento de lenguaje natural en español."

# Aplicar el modelo para obtener el documento
doc = nlp(texto)

# Imprimir el etiquetado morfológico de cada palabra
for token in doc:
    print("Palabra: ", "{:15}".format(token.text), "Etiqueta POS: ", "{:15}".format(token.pos_))
```

2. Técnicas de preprocesamiento de texto

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

```
Palabra:  Spacy          Etiqueta POS:  PROPN
Palabra:  es             Etiqueta POS:  AUX
Palabra:  una            Etiqueta POS:  DET
Palabra:  excelente      Etiqueta POS:  ADJ
Palabra:  herramienta    Etiqueta POS:  NOUN
Palabra:  para           Etiqueta POS:  ADP
Palabra:  procesamiento  Etiqueta POS:  NOUN
Palabra:  de             Etiqueta POS:  ADP
Palabra:  lenguaje       Etiqueta POS:  NOUN
Palabra:  natural        Etiqueta POS:  ADJ
Palabra:  en             Etiqueta POS:  ADP
Palabra:  español        Etiqueta POS:  NOUN
Palabra:  .              Etiqueta POS:  PUNCT
```