Práctica 1.2: Procesamiento de Texto

UNR - TUIA - Procesamiento de Lenguaje Natural

Ejercicio 1: Limpieza Básica de Mensajes

```
mensajes = [
    "  Hola!!!   ¿¿¿Cómo estás???  ",
    "NECESITO AYUDA URGENTE!!!!!!",
    "Este  es   un   texto   con    espacios    raros",
    "Me  encanta  programar  en  Python :) :) :)"
]

def limpiar_mensaje(texto):
    """
    TODO: Implementar limpieza básica
    1. Eliminar espacios al inicio y final
    2. Convertir a minúsculas
    3. Reducir signos de puntuación repetidos (!!! -> !)
    4. Normalizar espacios múltiples a uno solo
    """
    # Escribir código
    pass

# Resultado esperado:
# "hola! ¿cómo estás?"
# "necesito ayuda urgente!"
# "este es un texto con espacios raros"
# "me encanta programar en python :) :) :)"
```

Ejercicio 2: Contador de Palabras y Análisis Simple

```
texto_simple = """
Python es un lenguaje de programación.
Python es fácil de aprender.
Me gusta Python porque es versátil.
"""

def analizar_texto_basico(texto):
    """
    TODO: Implementar análisis básico
    1. Contar total de palabras
    2. Contar oraciones (puntos)
```

```
    3. Encontrar la palabra más repetida
    4. Calcular longitud promedio de palabras
    """
    resultado = {
        'total_palabras': 0,
        'total_oraciones': 0,
        'palabra_mas_frecuente': '',
        'longitud_promedio': 0
    }
    #Escribir código
    return resultado
```

Ejercicio 3: Eliminación de Stopwords Simple

```
stopwords_es = ['el', 'la', 'de', 'en', 'y', 'a', 'los', 'las', 'un', 'una',
'es', 'por']

oraciones = [
    "El gato está en la casa",
    "La programación es una habilidad importante",
    "Los estudiantes aprenden Python en la universidad"
]

def eliminar_stopwords(texto, stopwords):
    """
    TODO: Eliminar palabras comunes
    1. Convertir texto a minúsculas
    2. Dividir en palabras
    3. Filtrar stopwords
    4. Retornar texto limpio
    """
    # Escribir código
    pass

# Resultado esperado:
# "gato está casa"
# "programación habilidad importante"
# "estudiantes aprenden python universidad"
```

Ejercicio 4: Normalización de Números y Fechas

```
textos_con_numeros = [
    "El producto cuesta $1,234.56 pesos",
    "La fecha es 15/01/2024",
    "Tengo 25 años y mido 1.75 metros",
    "El descuento es del 15.5%"
```

```
]

def normalizar_numeros(texto):
    """
    TODO: Normalizar números en el texto
    1. Encontrar precios ($ seguido de números)
    2. Encontrar fechas (formato dd/mm/yyyy)
    3. Encontrar porcentajes
    4. Reemplazar por etiquetas [PRECIO], [FECHA], [PORCENTAJE]
    """
    #Escribir código
    pass

# Resultado esperado:
# "El producto cuesta [PRECIO] pesos"
# "La fecha es [FECHA]"
# "Tengo 25 años y mido 1.75 metros"
# "El descuento es del [PORCENTAJE]"
```

Ejercicio 5: Tokenización Básica

```
parrafo = "María compró 3 libros. Cada libro costó $150. ¡Le encantaron!"

def tokenizar_basico(texto):
    """
    TODO: Dividir texto en tokens
    1. Separar en oraciones (por . ! ?)
    2. Para cada oración, separar en palabras
    3. Retornar lista de listas
    """
    tokens = []
    # Escribir código
    return tokens

# Resultado esperado:
# [
#   ['María', 'compró', '3', 'libros'],
#   ['Cada', 'libro', 'costó', '$150'],
#   ['Le', 'encantaron']
# ]
```

Ejercicio 6: Corrección Ortográfica Simple

```
correcciones = {
    'hola': 'hola',
    'ola': 'hola',
```

```
    'vien': 'bien',
    'q': 'que',
    'xq': 'porque',
    'tb': 'también',
    'pq': 'porque'
}

mensajes_informales = [
    "ola q tal",
    "todo vien xq llegaste",
    "yo tb quiero ir"
]

def corregir_texto_informal(texto, diccionario):
    """
    TODO: Corregir texto usando diccionario
    1. Dividir en palabras
    2. Buscar cada palabra en el diccionario
    3. Reemplazar si existe corrección
    4. Unir palabras corregidas
    """
    # Escribir código
    pass
```

Ejercicio 7: Análisis de Frecuencia

```
texto_noticia = """
El gobierno anunció nuevas medidas económicas.
Las medidas incluyen reducción de impuestos.
El ministro explicó que las medidas son necesarias.
"""

def obtener_palabras_frecuentes(texto, top_n=3):
    """
    TODO: Encontrar las N palabras más frecuentes
    1. Limpiar y dividir texto
    2. Contar frecuencia de cada palabra
    3. Retornar las top_n más frecuentes
    """
    # Escribir código
    pass

# Resultado esperado (top 3):
# [('medidas', 3), ('el', 2), ('las', 2)]
```

Ejercicio 8: Segmentación Simple por Tamaño

```
texto_largo = "Python es un lenguaje de programación interpretado. " \
              "Es multiparadigma y multiplataforma. " \
              "Fue creado por Guido van Rossum. " \
              "Es muy popular en ciencia de datos."

def segmentar_por_palabras(texto, palabras_por_chunk=5):
    """
    TODO: Dividir texto en chunks de N palabras
    1. Dividir texto en palabras
    2. Agrupar cada N palabras
    3. Retornar lista de chunks
    """
    chunks = []
    # Escribir código
    return chunks

# Resultado esperado (chunks de 5 palabras):
# [
#   "Python es un lenguaje de",
#   "programación interpretado Es multiparadigma y",
#   "multiplataforma Fue creado por Guido",
#   "van Rossum Es muy popular",
#   "en ciencia de datos"
# ]
```

Ejercicio 9: Extracción de Entidades Simples

```
texto_info = """
Juan Pérez trabaja en Google.
María García estudia en la Universidad de Buenos Aires.
Carlos vive en Rosario, Argentina.
"""

def extraer_nombres_propios(texto):
    """
    TODO: Extraer palabras que empiecen con mayúscula
    1. Dividir en palabras
    2. Filtrar palabras que empiezan con mayúscula
    3. Excluir inicio de oraciones
    4. Retornar lista de nombres propios
    """
    nombres = []
    # Escribir código
    return nombres

# Resultado esperado:
```

```
# ['Juan', 'Pérez', 'Google', 'María', 'García', 'Universidad', 'Buenos',
'Aires', 'Carlos', 'Rosario', 'Argentina']
```

Ejercicio 10: Pipeline Básico Completo

```
texto_entrada = """
  HOLA!!!  Mi nombre es Ana...  Tengo 25 años.
  Vivo en Buenos Aires y trabajo en IT.
  Me gusta programar en Python y Java!!!
"""

def pipeline_procesamiento(texto):
    """
    TODO: Aplicar todos los pasos de procesamiento
    1. Limpiar espacios y puntuación excesiva
    2. Convertir a minúsculas
    3. Tokenizar en oraciones
    4. Eliminar stopwords
    5. Contar palabras finales
    """
    resultado = {
        'texto_limpio': '',
        'oraciones': [],
        'sin_stopwords': '',
        'total_palabras': 0
    }
    # Escribir código
    return resultado
```