# Procesamiento del Lenguaje Natural

## Caracterización de textos

# Requisitos

- **Formato**
  - Archivo ipynb.
  - “Si no se usa el archivo proporcionado, la actividad será calificada con cero puntos”.
- **Muestreo**
  - Aleatoriedad.
  - Suficientemente “grande”.
  - En torno a los 10k registros.
- **Evaluación**
  - Código ejecutado.
  - Código sin errores de bulto.
  - Descripción clara de lo que se ha hecho.
  - Análisis de los resultados obtenidos.

# Pregunta 1

¿Cuántos registros contiene el corpus?

- **Explicación**
  - Para responder esta pregunta, debías analizar el corpus (total o muestra) y contar el número total de registros.
  - Este número corresponde a la cantidad de filas o documentos en el archivo del corpus.
  - El objetivo de esta pregunta era asegurarte de que sabes realizar una operación básica de conteo sobre un dataset.

- **Código ejemplo**
  - len(corpus)

# Pregunta 2

¿Cuántas palabras totales hay en los comentarios del corpus?

- **Explicación**
  - Aquí se esperaba que sumaras el número de palabras en todos los comentarios del corpus.
  - El proceso incluía tokenizar los textos, es decir, dividirlos en palabras, y luego realizar la suma total.
  - Esto te permite comprender el tamaño del corpus en términos de volumen textual.

- **Código ejemplo**
  - sum(len(comentario.split()) for comentario in corpus)

# Pregunta 3

**¿Cuál es el número promedio de palabras en cada comentario?**

- **Explicación**
  - Para responder esta pregunta, debías dividir el número total de palabras (obtenido en la Pregunta 2) entre el número total de registros (de la Pregunta 1).
  - Esto calcula la media aritmética del número de palabras por comentario y te permite evaluar la extensión promedio de los textos.

- **Código ejemplo**
  - total_palabras / len(corpus)

# Pregunta 4

**Considerando dos grupos de comentarios (odio y no odio), ¿cuál es el número promedio de palabras en los comentarios de cada grupo?**

- **Explicación**
  - En esta pregunta debías segmentar el corpus en dos grupos: comentarios etiquetados como "odio" y "no odio".
  - Luego, debías calcular la media del número de palabras para cada grupo, lo que implicaba repetir el procedimiento de la Pregunta 3, pero de manera independiente para cada subconjunto.

- **Código ejemplo**
  - ['INTENSIDAD'] > 0
  - promedio_palabras_odio = sum(len([token for token in nlp(c) if not token.is_punct]) for c in corpus_odio) / len(corpus_odio)
  - promedio_palabras_no_odio = sum(len([token for token in nlp(c) if not token.is_punct]) for c in corpus_no_odio) / len(corpus_no_odio)

# Pregunta 5

**Considerando dos grupos de comentarios (odio y no odio), ¿cuál es el número promedio de oraciones en los comentarios de cada grupo?**

- **Explicación**
  - Utiliza SpaCy para dividir los comentarios en oraciones (usando nlp y el atributo .sents).
  - Luego, calcula el promedio de oraciones por comentario en los dos grupos.
  - Este análisis permite comparar la complejidad estructural entre los grupos.

- **Código ejemplo**
  - promedio_oraciones_odio = sum(len(list(nlp(c).sents)) for c in corpus_odio) / len(corpus_odio)
  - promedio_oraciones_no_odio = sum(len(list(nlp(c).sents)) for c in corpus_no_odio) / len(corpus_no_odio)

# Pregunta 6

**Considerando dos grupos de comentarios (odio y no odio), ¿cuál es el porcentaje de comentarios que contienen entidades NER en cada grupo?**

- **Explicación**
  - Usa SpaCy para realizar reconocimiento de entidades nombradas (NER) y contar los comentarios que contienen al menos una entidad (usando len(nlp(c).ents) > 0).
  - Luego, calcula el porcentaje de comentarios con entidades respecto al total de cada grupo.
  - Este análisis mide la prevalencia de información estructurada en los textos.

- **Código ejemplo**
  - porcentaje_entidades_odio = (sum(1 for c in corpus_odio if len(nlp(c).ents) > 0) / len(corpus_odio)) * 100
  - porcentaje_entidades_no_odio = (sum(1 for c in corpus_no_odio if len(nlp(c).ents) > 0) / len(corpus_no_odio)) * 100

# Pregunta 7

**Considerando dos grupos de comentarios (odio y no odio), ¿cuál es el porcentaje de comentarios que contienen entidades NER de tipo PERSON en cada grupo?**

- **Explicación**
  - Usa SpaCy para identificar entidades nombradas de tipo PERSON en los comentarios.
  - Luego, calcula el porcentaje de comentarios que contienen este tipo de entidad respecto al total de comentarios en cada grupo.
  - Este análisis te ayuda a explorar si los comentarios con odio tienden a referirse a personas específicas.

- **Código ejemplo**
  - porcentaje_personas_odio = (sum(1 for c in corpus_odio if any(ent.label_ == 'PER' for ent in nlp(c).ents)) / len(corpus_odio)) * 100
  - porcentaje_personas_no_odio = (sum(1 for c in corpus_no_odio if any(ent.label_ == 'PER' for ent in nlp(c).ents)) / len(corpus_no_odio)) * 100

# Pregunta 8

**Considerando dos grupos de comentarios (odio y no odio), ¿cuál es el porcentaje de palabras en cada combinación posible de género y número (p.ej., masculino singular) en cada grupo?**

- **Explicación**
  - Usa SpaCy para etiquetar gramaticalmente las palabras (POS tagging).
  - Analiza las características de género (Gender) y número (Number) en las palabras de cada grupo.
  - Calcula la distribución porcentual de estas combinaciones dentro de cada grupo, incluyendo palabras que no tengan género o número.
  - Este análisis permite explorar posibles sesgos lingüísticos en los comentarios.

- **Código ejemplo**

```python
from collections import defaultdict

def genero_numero(comentarios):
    contador = defaultdict(int)
    total_palabras = 0

    for comentario in comentarios:
        for token in comentario:
            if token.is_alpha:
                total_palabras += 1
                genero = token.morph.get("Gender", "_")
                numero = token.morph.get("Number", "_")
                contador[f"{genero}/{numero}"] += 1

    porcentaje_genero_numero = {}
    for clave, count in contador.items():
        porcentaje_genero_numero[clave] = (count / total_palabras) * 100 if total_palabras else 0

    return porcentaje_genero_numero
```

# Pregunta 9

**Considerando dos grupos de comentarios (odio y no odio), indica cuántas entidades de cada tipo posible se reconocen en cada uno de los grupos.**

- **Explicación**
  - Usa SpaCy para etiquetar gramaticalmente las palabras (POS tagging).
  - Analiza las características de género (Gender) y número (Number) en las palabras de cada grupo.
  - Calcula la distribución porcentual de estas combinaciones dentro de cada grupo, incluyendo también aquellas palabras que no tienen género ni número.
  - Este análisis permite explorar posibles sesgos lingüísticos en los comentarios.

- **Código ejemplo**

```
def contar_entidades_por_tipo(comentarios):
    tipos_entidades = Counter()
    for comentario in comentarios:
        for ent in nlp(comentario).ents:
            tipos_entidades[ent.label_] += 1
    return tipos_entidades

entidades_odio = contar_entidades_por_tipo(corpus_odio)
entidades_no_odio = contar_entidades_por_tipo(corpus_no_odio)
```

# Pregunta 10

**Considerando dos grupos de comentarios (odio y no odio), extrae y muestra los 100 lemas más repetidos en los comentarios de cada grupo.**

- **Explicación**
  - Usa SpaCy para lematizar las palabras (reducirlas a su forma base o diccionario) y cuenta la frecuencia de cada lema.
  - Excluye las stopwords y signos de puntuación, ya que no aportan significado relevante.
  - Luego, ordena los resultados en cada grupo y muestra los 100 más frecuentes. Este análisis identifica temas o patrones recurrentes en cada grupo.

- **Código ejemplo**

```
def extraer_lemas_frecuentes(comentarios, n=100):
    lemas = Counter()
    for comentario in comentarios:
        for token in nlp(comentario):
            if not token.is_stop and not token.is_punct:
                lemas[token.lemma_] += 1
    return lemas.most_common(n)

lemas_frecuentes_odio = extraer_lemas_frecuentes(corpus_odio)
lemas_frecuentes_no_odio = extraer_lemas_frecuentes(corpus_no_odio)
```

# Pregunta 11

**¿Es posible utilizar alguna de las características extraídas en las preguntas anteriores para determinar si un mensaje contiene odio? Justifica tu respuesta con el análisis estadístico que consideres necesario.**

- **Explicación**
  - Esta pregunta busca que apliques análisis estadístico o técnicas de machine learning para evaluar si características como número de palabras, porcentaje de entidades NER o distribución de género y número son útiles para clasificar mensajes.
  - Justifica si estas características son discriminativas y por qué, realizando un análisis crítico y detallado.

- **Ejemplo**
  - Es significativo el número de oraciones que contienen odio, (1.57 oraciones), frente a más de dos que no contienen odio (2.27)
  - El lema más repetido en los comentarios sin odio es el cuarto más repetido en los comentarios que contienen odio (Gobierno), pero que un comentario incluya uno de los tres primeros lemas más repetido en los comentarios que contienen odio si podría dignificar que dicho comentario contiene odio.
    - {mierda; puta; asco}

| | | Masculino | Femenino |
|---|---|---|---|
| No contienen odio | Singular | 40.55 | 32.57 |
| | Plural | 16.07 | 10.81 |
| Contienen odio | Singular | 40.49 | 32.99 |
| | Plural | 18.24 | 8.28 |


www.unir.net