Procesamiento de Lenguaje Natural (NLP)

Clase 1: Introducción y Preprocesamiento de Texto

Prof. D.Sc. BARSEKH-ONJI Aboud

Faculty of Engineering
Universidad Anáhuac México Sur

17 de octubre de 2025

Agenda de la Clase

1. Introducción al NLP
2. El Desafío del Lenguaje
3. Preprocesamiento de Texto: La Limpieza de Datos
4. Ejemplo Práctico en MATLAB
5. Conclusiones y Próximos Pasos

Agenda

1. Introducción al NLP
2. El Desafío del Lenguaje
3. Preprocesamiento de Texto: La Limpieza de Datos
4. Ejemplo Práctico en MATLAB
5. Conclusiones y Próximos Pasos

¿Qué es el Procesamiento de Lenguaje Natural?

Definición

Es un campo de la Inteligencia Artificial (IA) que permite a las computadoras **comprender, interpretar y manipular el lenguaje humano**.

- Busca cerrar la brecha entre la comunicación humana y la comprensión de las computadoras.
- No se trata solo de palabras, sino de entender el **contexto**, la **intención** y el **sentimiento**.

Lenguaje Humano $\Longleftrightarrow$ Lenguaje de Máquina

Aplicaciones en el Mundo Real

- **Traductores Automáticos:**
  Google Translate, DeepL.
- **Asistentes Virtuales:**
  Siri, Alexa, Google Assistant.
- **Análisis de Sentimientos:**
  ¿Qué opinan los clientes de un producto en redes sociales?
- **Chatbots:**
  Atención al cliente automatizada.
- **Clasificación de Texto:**
  Organizar correos en spam/no spam.
- **Resumen de Textos:**
  Crear un resumen de un largo artículo de noticias.

Agenda

¿Por qué es tan difícil para una máquina?

El lenguaje humano es inherentemente complejo y ambiguo.

El gran problema: La Ambigüedad

Consideremos la frase: *"Vi a un hombre en la colina con un telescopio."*

- ¿Quién tiene el telescopio?
  - ¿Lo estaba usando yo para ver al hombre?
  - ¿El hombre en la colina tenía el telescopio?

- Los humanos usamos el **contexto** para resolver esto. Las máquinas necesitan ser entrenadas para hacerlo.

Otros desafíos incluyen la ironía, el sarcasmo, las jergas y la evolución constante del lenguaje.

Agenda

1. Introducción al NLP
2. El Desafío del Lenguaje
3. Preprocesamiento de Texto: La Limpieza de Datos
4. Ejemplo Práctico en MATLAB
5. Conclusiones y Próximos Pasos

La Base de Todo Proyecto de NLP

Analogía

El preprocesamiento de texto es como la *mise en place* en la cocina: antes de poder cocinar, debes lavar, pelar y cortar los ingredientes.

- El texto del mundo real es "sucio": contiene mayúsculas, puntuación, palabras irrelevantes, etc.
- Nuestro objetivo es **estandarizar y limpiar** el texto para que los algoritmos de IA puedan procesarlo de manera eficiente.

Texto Crudo $\rightarrow$ **Limpieza** $\rightarrow$ **Texto Procesado**

Pasos Clave del Preprocesamiento

1. **Tokenización:** Dividir el texto en unidades más pequeñas (palabras o "tokens").

Pasos Clave del Preprocesamiento

1. **Tokenización:** Dividir el texto en unidades más pequeñas (palabras o "tokens").
2. **Normalización:** Convertir todo a minúsculas.

Pasos Clave del Preprocesamiento

1. **Tokenización:** Dividir el texto en unidades más pequeñas (palabras o "tokens").
2. **Normalización:** Convertir todo a minúsculas.
3. **Eliminación de *Stop Words*:** Quitar palabras comunes sin significado (ej. "el", "y", "un").

Pasos Clave del Preprocesamiento

1. **Tokenización:** Dividir el texto en unidades más pequeñas (palabras o "tokens").
2. **Normalización:** Convertir todo a minúsculas.
3. **Eliminación de *Stop Words*:** Quitar palabras comunes sin significado (ej. "el", "y", "un").
4. ***Stemming* (Derivación):** Reducir las palabras a su raíz.

Veremos cada uno de estos pasos en detalle y luego los aplicaremos en MATLAB.

1. Tokenización

Objetivo

Segmentar un texto en sus componentes básicos, llamados **tokens**. Generalmente, un token es una palabra.

Ejemplo:

"El NLP es fascinante." ↓ [El] [NLP] [es] [fascinante] [.]

En MATLAB..

La función `'tokenizedDocument'` se encarga de esto y mucho más.

2. Normalización (a Minúsculas)

Objetivo

Asegurar que la misma palabra se trate de igual forma, sin importar si está en mayúsculas o minúsculas.

Ejemplo:

- El algoritmo vería "Hola", "hola" y "HOLA" como tres palabras diferentes.
- Al convertir todo a minúsculas, las tres se convierten en **"hola"**.

Hola mundo $\rightarrow$ **hola** mundo

En MATLAB..

usamos la función `'lower()'`

3. Eliminación de *Stop Words*

Objetivo

Eliminar palabras extremadamente comunes que no aportan un significado semántico relevante al texto.

- Ejemplos en español: *el*, *la*, *y*, *a*, *un*, *pero*, *por*, *para*...
- Estas palabras son útiles para la gramática humana, pero a menudo son "ruido" para los modelos de IA.

Ejemplo:

"el perro **y** el gato corren **en** el parque" $\downarrow$ "perro gato corren parque"

En MATLAB..

Usamos la función `'removeWords()'`.

4. *Stemming* (Derivación)

Objetivo

Reducir una palabra a su raíz o "stem", aunque el resultado no sea siempre una palabra real. Es un método rápido y heurístico.

- Ayuda a agrupar diferentes formas de una misma palabra.
- Por ejemplo, queremos que el modelo entienda que "correr", "corriendo" y "corrió" se refieren al mismo concepto.

Ejemplo:

aprendiendo $\rightarrow$ **aprend**
aprender $\rightarrow$ **aprend**

En MATLAB..

La función `'normalizeWords'` con el estilo `'stem'` realiza esta tarea.

4. *Stemming* (Derivación)

1. Stemming (Derivación)

- **Cómo funciona:** Corta el final de las palabras usando reglas. Es rápido y eficiente.
- **Resultado:** La raíz"no siempre es una palabra real.
- **Ejemplo:**
  - ‘aprendizaje‘ → ‘aprendizaj‘
  - ‘aprendiendo‘ → ‘aprend‘
- **Uso:** `normalizeWords(doc, 'Style', 'stem')`

2. Lemmatization (Lematización)

- **Cómo funciona:** Usa un diccionario para encontrar la palabra raíz (lema). Es más preciso pero más lento.
- **Resultado:** Siempre es una palabra real.
- **Ejemplo:**
  - ‘mejores‘ → ‘bueno‘
  - ‘fui‘ → ‘ir‘
- **Uso:** `normalizeWords(doc, 'Style', 'lemma')`

4. *Stemming* (Derivación)

Nuestra Elección para la Clase

Usaremos **Stemming** porque es ideal para tareas de clasificación: es muy rápido y efectivo para agrupar las palabras clave de un texto.

Agenda

1. Introducción al NLP
2. El Desafío del Lenguaje
3. Preprocesamiento de Texto: La Limpieza de Datos
4. Ejemplo Práctico en MATLAB
5. Conclusiones y Próximos Pasos

Limpiando un Comentario de Principio a Fin

Objetivo

Aplicar todos los pasos de preprocesamiento a una oración real usando el Text Analytics Toolbox™ de MATLAB.

```
textoOriginal = 'Los alumnos estan aprendiendo sobre el Procesamiento de
    Lenguaje Natural y están muy emocionados.';

doc = tokenizedDocument(lower(textoOriginal));

listaStopWords = ["y", "el", "la", "los", "las", "un", "una", "de", "sobre",
     "estan", "están", "."];
documento = removeWords(doc, listaStopWords);

documentoProcesado = normalizeWords(documento, 'Style', 'stem');

palabrasFinales = string(documentoProcesado.tokenDetails.Token);
```

Análisis del Resultado

Texto Original:

- "Los alumnos estan aprendiendo sobre el Procesamiento de Lenguaje Natural y estan muy emocionados."

Tokens Finales (Salida del Código)

```
'alumn'
'aprend'
'procesamiento'
'lenguaj'
'natural'
'muy'
'emocion'
```

Análisis del Resultado

Observaciones

- El texto se ha reducido a sus palabras más significativas.
- Las palabras están en su forma raíz, lo que facilita el análisis posterior.
- ¡Este texto "limpio" ya está listo para ser convertido en números para un modelo de IA!

Agenda

Resumen y Conclusiones

- **¿Qué es el NLP?** Es el campo que enseña a las máquinas a entender el lenguaje humano.

- **El Preprocesamiento es Clave:** No podemos trabajar con texto "crudo". Limpiar y estandarizar los datos es el primer paso y el más importante.

- **Nuestra Caja de Herramientas:** Hemos aprendido 4 técnicas fundamentales:
  1. Tokenización
  2. Normalización a minúsculas
  3. Eliminación de Stop Words
  4. Stemming

Próximos Pasos

La Próxima Frontera: De Palabras a Números

Hemos limpiado nuestras palabras, pero los modelos de Machine Learning no entienden "alumn" o "aprend". Entienden números.

En la próxima clase, aprenderemos a convertir nuestro texto procesado en **vectores numéricos** usando técnicas como **Bag-of-Words** y **TF-IDF**.

¡Será el puente definitivo entre el lenguaje y las matemáticas!