**Unidad 3 – Procesamiento del lenguaje natural**

**Introducción**

El procesamiento del lenguaje natural es un campo que se encuentra en la intersección entre lingüística, inteligencia artificial y informática. Permite que los ordenadores sean capaces de interpretar, analizar y generar lenguaje humano.

El PLN surge en alrededor de 1950 con la prueba de Alan Turing que evaluaba si un ordenador pudriera utilizar lenguaje natural para hacer creer que era humana.

Existen multitud de aplicaciones a partir del PLN a parte de la generación de texto humano, como, por ejemplo: detección de spam, corrección ortográfica, accesibilidad, etc...

Un algoritmo de PLN se puede crear en cualquier lenguaje, pero Python dispone de una de las mejores librerías NLTK (Natual Language Toolkit)

**Flujo de trabajo en PLN**

¿Cómo se afronta la resolución de un problema mediante aprendizaje automático?

* Definición del problema (pregunta)
  + Exploración de datos (EPA)
* Aplicación de técnicas de análisis
* Comunicación de los resultados

**Pre-procesamiento de texto**

Limpiar y preparar el texto es crucial para el PLN, este suele ser el primer paso para realizar. A través de NLTK se puede:

* Eliminar el ruido (Noise removal): quitar etiquetas HTML
* Tokenizar: dividir el texto en palabras individuales.
* Normalización: limpiar el texto de otras formas:
  + Derivación (Streaming): identificar prefijos y sufijos de palabras
  + lematización: simplificar términos a su forma raíz (‘soy’ y ‘son’ se convierte a ‘ser’)
  + Otras tareas: pasar a minúsculas, eliminar palabras vacías (stopwords)

**Análisis (parsing) de texto**

Una vez el texto está limpio, es útil saber cómo se relacionan las palabras entre sí. El análisis de texto es un proceso de PLN relacionado con la segmentación de texto según su sintaxis. NLTK permite:

* Etiquetado de la parte del discurso (Part-of-speech tagging): identificar partes del discurso (verbos, sustantivos, etc.…)
* Reconocimientos de entidades nombradas (Named entity recognition): ayuda a identificar los nombres propios.
* Árboles de dependencia gramatical: ayuda a comprender las palabras de una oración.
* Análisis de expresiones regulares: reconocer patrones de forma flexible. Se puedes combinar con el etiquetado (POS) para encontrar direcciones, emails, etc.…

**Modelos de lenguaje**

**Bolsa de palabras (Bag-of-words)**

Los modelos lingüísticos son modelos probabilísticos del lenguaje, permiten calcular la probabilidad de que se use un sonido, letra, palabra o frase determinados.

Uno de los modelos de lenguaje mas comunes es el modelo monograma, mas conocido como ‘Bolsa de palabras’. Es un modelo estadístico que tiene un recuento de cada instancia para cada palabra (no tiene orden).

Este modelo es una buena manera de ver el lenguaje cuando se desea hacer predicciones sobre el tema o sentimiento de un texto. **Cuando la gramática y el orden de las palabras es irrelevante, probablemente este modelo es una buena opción.**

**N-Gramas y NLM**

El modelo n-grama considera una secuencia de n unidades y calcula la probabilidad de cada unidad dada la secuencia anterior de longitud n. Usualmente las cadenas mas largas suelen ser más útiles en este modelo. Ejemplo: un bigrama tiene n = 2.

Este modelo tiene algunos problemas

* Durante el entrenamiento se puede encontrar palabras que nunca ha visto antes por lo que le costara darle sentido.
* A medida que aumenta la longitud de la secuencia, se reduce el numero de ejemplo. Con pocos ejemplos es difícil hacer una predicion.

**Modelos temáticos (Topic modeling)**

El modelado temático es un área del PLN dedicado a descubrir temas latentes u ocultos en el cuerpo de un lenguaje. Una técnica más común es priorizar los términos usados con menor frecuencia, conocido como **frecuencia de términos – frecuencia inversa en documentos (tf-idf).**

Una vez determinada la frecuencia de palabras, el siguiente paso suele ser la asignación de Dirichlet latente (LDA). Este modelo determina que palabras siguen apareciendo juntas en los mismos contextos.

Word2vec es una técnica que mapea los resultados de un modelo temático para que las palabras usadas de manera similar estén más juntas.

**Similitud textual**

La similitud textual es un desafío dentre del procesamiento del lenguaje natural para la corrección ortográfica y la similitud de palabras.

La distancia de Levenshtein es la distancia de ediccion mínima entre dos palabras. Las tecinas de autocoreccion también consideran la distancia del teclado y similitud fonética.

Esta técnica también permite detectar plagios, utilizando la similitud léxica. O hacer recomendaciones a través de la similitud semántica (temas o significados similares)

**Predicción de idomas y generación de texto**

La predicción del lenguaje es una aplicación del PLN que busca predecir el texto dado el texto anterior. Para ello se pueden utilizar modelos como el bag of words, cadenas de markov o memoria a corto plazo a largo plazo (LSTM)

**Temas avanzados de PLN**

Existan un gran numero de aplicaciones avanzadas del PLN como:

* Clasificadores Naive Bayes: aprendizaje automático supervisado que aprovechan un teorema probabilístico para hacer predicciones y clasificaciones. Utilizado principalmente en el análisis de sentimientos
* Traducción automática.
* Accesibilidad del idioma para el reconociemtno de voz
* Detección de sesgos en la escritura o habla