Resumen Tema 2: PLN

El procesamiento del lenguaje natural (PLN) es un campo de la informática y la inteligencia artificial que se centra en cómo interactúan los ordenadores con las lenguas humanas. El PLN implica el estudio de cómo programar ordenadores para procesar y analizar grandes cantidades de datos de lenguaje natural. El objetivo del PLN es permitir a las máquinas entender, analizar y generar lenguaje humano de forma efectiva.

Algunas de las tareas que se realizan con el PLN son la clasificación de textos, la generación de resúmenes, la traducción de textos de un idioma a otro, el análisis de sentimiento, y la extracción de información.

Clasificación de texto La clasificación de texto es una tarea común del PLN que consiste en categorizar o predecir la clase de un documento de texto, a menudo con la ayuda del aprendizaje automático supervisado. Esta tarea tiene muchos usos prácticos en diferentes sectores, como la clasificación de spam, la categorización de noticias y blogs, la categorización de solicitudes de atención al cliente, y la detección de discursos de odio.

Existen dos tipos principales de sistemas de clasificación de textos: los sistemas basados en reglas y los sistemas basados en aprendizaje automático.

- **Sistemas basados en reglas**: Utilizan un conjunto de reglas lingüísticas construidas manualmente para clasificar el texto. Estas reglas indican al sistema cómo clasificar el texto en una categoría basándose en elementos textuales semánticamente relevantes. Los sistemas basados en reglas son comprensibles y pueden perfeccionarse con el tiempo, pero requieren de conocimientos profundos en la materia, consumen mucho tiempo y son difíciles de mantener y ampliar.
- **Sistemas basados en aprendizaje automático**: Aprenden el mapeo de los datos de entrada (texto en bruto) con las etiquetas (variables objetivo) utilizando algoritmos de aprendizaje automático supervisado. Estos sistemas tienen dos fases: entrenamiento y predicción. Durante la fase de entrenamiento, se entrena un algoritmo de aprendizaje automático supervisado en el conjunto de datos de entrada etiquetados. Al final de este proceso, se obtiene un modelo entrenado que se puede utilizar para obtener predicciones sobre datos nuevos y no vistos. En la fase de predicción, el modelo entrenado se usa para predecir etiquetas en datos nuevos y no vistos.

Preprocesamiento de textos El preprocesamiento de datos de texto es un paso importante en cualquier tarea de PLN. Los datos de texto son difíciles de procesar porque no están estructurados y contienen ruido. El objetivo del preprocesamiento de texto es limpiar y preparar los datos de texto para su posterior tratamiento o análisis. Este proceso puede incluir tareas como:

- **Tokenización:** Dividir un texto en unidades más pequeñas llamadas tokens.
- **Eliminación de palabras vacías (stop words):** Eliminar palabras comunes que no aportan significado relevante al texto.
- **Lematización y stemming:** Reducir las palabras a sus formas base o raíz.

Extracción de características Los métodos más comunes para extraer características del texto, es decir, convertir datos de texto en características numéricas para poder entrenar un modelo de aprendizaje automático, son:

- **Bolsa de palabras (Bag of Words):** Representar el texto como una colección de palabras, donde cada documento se convierte en un vector que contiene la frecuencia de aparición de cada palabra del vocabulario.
- **TF-IDF (Frecuencia de término-frecuencia inversa de documento):** Ponderar la importancia de cada palabra en un documento, considerando la frecuencia de la palabra en el documento y en todo el corpus.

Word embeddings Los *word embeddings* son una técnica avanzada para representar palabras en un espacio vectorial continuo. A diferencia de las representaciones tradicionales basadas en frecuencias, los *word embeddings* capturan el significado semántico de las palabras modelando sus contextos de uso. Palabras con significados similares tienden a estar cercanas en este espacio. Algunos modelos populares de *word embeddings* son Word2Vec, GloVe y FastText.

N-gramas Los N-gramas son secuencias contiguas de *n* elementos de un texto o discurso. Los elementos pueden ser caracteres, sílabas, palabras o incluso frases. Los N-gramas se usan para modelar el contexto y la estructura del texto. Algunos tipos de n-gramas son: unigramas, bigramas, trigramas.

Reconocimiento de entidades nombradas (NER) El Reconocimiento de Entidades Nombradas (NER) es una técnica que identifica y clasifica entidades mencionadas en un texto, como nombres de personas, organizaciones, lugares, fechas, cantidades y otras categorías predefinidas.

Análisis de sentimientos El análisis de sentimientos es una técnica que determina las emociones, opiniones y actitudes expresadas en un texto. El objetivo es clasificar las expresiones en categorías como positivas, negativas o neutras.

El PLN es un campo de investigación y desarrollo activo. No existe un único algoritmo que sea el mejor, y es importante experimentar para encontrar lo que funciona mejor para los datos y los objetivos de cada tarea.

El Procesamiento de Lenguaje Natural (PLN) es una disciplina que se originó en los años 60, estrechamente relacionada con la Inteligencia Artificial y la Computación Lingüística, así como con áreas como la Recuperación de Información, la Computación Afectiva y la Psicología.

Fases del PLN El análisis del lenguaje se divide en varias fases:

- **Preprocesamiento**: Implica la preparación del texto para su análisis, incluyendo la decodificación, identificación del idioma, eliminación de elementos no relevantes, limpieza y normalización, corrección de errores, lematización, stemming y segmentación. La segmentación, también conocida como tokenización, divide el texto en unidades básicas llamadas "tokens".
- **Análisis léxico**: Se enfoca en identificar la función de cada término dentro de una frase, utilizando etiquetas POS (Part of Speech) como sustantivo, verbo, adjetivo, etc..
- **Análisis sintáctico**: Analiza la estructura de las oraciones y las relaciones entre las palabras, utilizando herramientas como los analizadores sintácticos (parsers), que representan gráficamente las oraciones en forma de árboles sintácticos. También incluye el "chunking", que busca patrones o subfrases con una estructura específica.

- **Análisis semántico**: Se centra en comprender el significado de las palabras y las oraciones, utilizando recursos externos como diccionarios, tesauros y ontologías.
- **Análisis pragmático**: Se ocupa de la intencionalidad de las frases y cómo el contexto afecta su interpretación.

Preprocesamiento de Textos

El preprocesamiento de textos es una fase fundamental en el PLN que implica varias tareas:

- **Identificación del idioma**: Determina el idioma del texto para poder extraer información con mayor precisión.
- **Eliminación de elementos no relevantes**: Descarta secciones no textuales, como etiquetas HTML en páginas web.
- **Limpieza y normalización de términos**: Elimina acentos, guiones, signos de puntuación y normaliza las palabras (por ejemplo, convirtiendo a minúsculas).
- **Corrección de errores**: Utiliza herramientas de corrección ortográfica ("spell-checkers") para corregir palabras mal escritas.
- **Eliminación de palabras vacías (stop words)**: Remueve términos comunes que no aportan mucho significado, como artículos y preposiciones.
- **Lematización**: Reduce las palabras a su forma base o lema.
- **Stemming**: Reduce las palabras a su raíz común.
- **Segmentación (tokenización)**: Divide el texto en unidades básicas, como palabras o frases.

Aplicaciones del PLN

El PLN tiene diversas aplicaciones, incluyendo:

- Recuperación de información.
- Clasificación de textos.
- Generación automática de texto.
- Bots automáticos (como Alexa o Siri).
- Análisis de sentimientos.
- Traducción automática
- Extracción de información

Herramientas de PLN

Existen diversas herramientas y bibliotecas para el procesamiento del lenguaje natural:

- **NLTK (Natural Language Toolkit)**: Una biblioteca de Python para PLN.
- **Stanford NLP (Stanza)**: Una biblioteca de última generación que ofrece análisis de texto en múltiples idiomas.
- **spaCy**: Una biblioteca de Python rápida y eficiente, ideal para aplicaciones en producción, con modelos preentrenados para varios idiomas.
- **UDPipe**: Una herramienta multilingüe basada en el proyecto Universal Dependencies.
- **NLP-Cube**: Una herramienta para analizar grandes volúmenes de texto en varios idiomas usando redes neuronales.
- **Librerías específicas por idioma**: Como GermaNER para alemán, Tint para italiano, y MeCab, Kuromoji y GiNZA para japonés.

Corpus lingüísticos

Los corpus lingüísticos son colecciones de textos utilizados para el estudio y desarrollo de aplicaciones de PLN. Algunos corpus importantes incluyen:

- **British National Corpus (BNC)**: Un corpus de referencia del inglés británico.
- **Corpus of Contemporary German (DTA)**: Un corpus de alemán contemporáneo.
- **Corpus PAISÀ**: Un corpus de textos italianos contemporáneos de la web.
- **NINJAL-LWP**: Un corpus de referencia para el estudio del japonés.

En resumen, el PLN es un campo multidisciplinario que involucra diversas fases de análisis y utiliza herramientas y recursos específicos para comprender y procesar el lenguaje humano. Sus aplicaciones son amplias y abarcan desde la recuperación de información hasta la creación de sistemas inteligentes de interacción con humanos.

Análisis de Sentimientos y Modelos de Lenguaje

- Inicialmente, el análisis de sentimientos se realizaba mediante la **polaridad de las palabras**, asignando puntuaciones a palabras positivas y negativas. Sin embargo, este método era **limitado con la ironía**.
- En 2012, con el auge de las redes neuronales, se empezaron a utilizar **redes neuronales recurrentes** para analizar texto. Estas redes aprendían la relación entre secuencias de palabras y su sentimiento, requiriendo **datasets etiquetados manualmente**.
- Entrenar una red neuronal para una tarea diferente requería un nuevo dataset y entrenamiento. En 2017, se adoptó la idea de usar **modelos pre-entrenados**, ajustándolos con menos datos para tareas específicas, similar a lo que se hacía en visión por ordenador.
- Con la llegada de los **Transformers**, se entrenaron modelos para **entender el lenguaje** a través de tareas como rellenar huecos en frases o completar textos. Esto cambió el paradigma del aprendizaje supervisado al **aprendizaje auto-supervisado**, donde las etiquetas se generaban automáticamente.
- A partir de entonces, los **modelos de lenguaje** se entrenaron con grandes cantidades de texto de internet, lo que les permitió aprender a entender cómo nos comunicamos. Estos modelos se comparten gratuitamente en plataformas como **Hugging Face**.
- Los modelos pre-entrenados se pueden usar directamente para tareas como análisis de sentimiento o se pueden **re-entrenar con datos etiquetados** para un mejor rendimiento.
- En 2020, el entrenamiento de modelos de lenguaje basados en **Transformers** escaló a tamaños enormes, como **GPT-3** con 175 mil millones de parámetros. Estos modelos, entrenados para autocompletar texto, también aprendieron a realizar otras tareas como la traducción.
- Modelos como GPT-3 pueden realizar tareas para las que no fueron entrenados explícitamente, como el análisis de sentimientos, simplemente ajustando la dinámica del problema.

Transformers y Mecanismos de Atención

- Los Transformers surgieron en 2017, transformando la concepción de la inteligencia artificial.
- Antes de los Transformers, se utilizaban **redes neuronales recurrentes** para procesar secuencias de texto, procesando cada palabra secuencialmente y agregando la información procesada a la siguiente palabra.
- Las redes recurrentes tienen problemas para recordar las primeras palabras de una frase debido a que su peso disminuye durante el entrenamiento, lo que dificulta encontrar relaciones entre palabras distanciadas.
- Los **mecanismos de atención** surgieron como una alternativa para solucionar el problema de la falta de memoria en las redes recurrentes.
- Estos mecanismos se basan en que cada palabra se representa como un **vector numérico**, y las redes neuronales aprenden a generar dos vectores distintos: uno para identificar las propiedades interesantes de la palabra (**vector "key"**) y otro para describir las propiedades que la palabra está buscando (**vector "query"**).
- La relación entre palabras se calcula mediante el **producto escalar** entre los vectores "query" y "key". Esto genera un **vector de atención** que indica la importancia que el modelo le da a cada palabra para dar contexto a la palabra actual.
- La matriz de atención muestra la importancia que cada palabra asigna al resto de la frase. Esto permite a la IA encontrar relaciones entre palabras en diferentes idiomas, incluso cuando el orden no se preserva.

- Para contextualizar las palabras, se utiliza un **vector de valor**, que se combina con las atenciones calculadas mediante una **suma ponderada**. Esto da como resultado vectores de palabras que recogen el contexto de la frase.
- Los Transformers utilizan los mecanismos de atención como base, sin la necesidad de redes recurrentes. El paper "Attention Is All You Need" introdujo esta arquitectura.

Embeddings

- Las redes neuronales solo procesan números, por lo que el texto debe ser vectorizado. El **one-hot encoding** convierte cada palabra en un vector, donde todas las componentes son cero excepto la que corresponde a la palabra.
- Aunque el one-hot encoding es un buen punto de partida, no refleja las relaciones semánticas entre palabras. Los **embeddings** buscan capturar estas relaciones.
- Un embedding es una representación compacta y ordenada de los datos que una red neuronal aprende para resolver una tarea específica.
- El embedding resultante depende de la tarea que la red deba resolver.
- Los **embeddings pre-entrenados** pueden mejorar el rendimiento de las redes, ya que transfieren conocimiento de una tarea a otra. **Word2Vec** es uno de los sistemas de embeddings pre-entrenados más utilizados.
- Los embeddings se entrenan con mucho texto y tareas genéricas, lo que permite que sean universales y aplicables a diversas tareas. Esto permite construir una estructura del vocabulario donde palabras conceptualmente similares están representadas de manera próxima en un espacio vectorial.