PRÁCTICA 3 y 4

Procesadores del lenguaje

Jerónimo Boza
Celia Escudero

UNIVERSIDAD
NEBRIJA

PRÁCTICA 3. DOCUMENTACIÓN

¿Qué diferencias encuentras entre las librerías? ¿Cuáles son las fortalezas y limitaciones de cada una?

Las librerías NLTK, SpaCy, TextBlob y Gensim tienen enfoques y fortalezas diferentes en el procesamiento de lenguaje natural.

NLTK es una librería completa y flexible, ideal para aprender y experimentar con diversas técnicas de NLP, como tokenización, etiquetado gramatical, y análisis sintáctico. Su fortaleza está en su amplitud y alcance, pero su limitación es que puede ser más lenta en comparación con otras librerías, especialmente cuando se trabaja con grandes volúmenes de texto.

SpaCy, por otro lado, está optimizada para la velocidad y la eficiencia. Es ideal para aplicaciones de NLP en producción, ya que es rápida y precisa en tareas como la lematización, el etiquetado de partes del discurso y el análisis de dependencias. Sin embargo, su limitación es que no tiene tanta flexibilidad en cuanto a herramientas y recursos como NLTK.

TextBlob es una librería más sencilla y fácil de usar, especialmente buena para tareas de análisis de sentimientos, traducción y corrección ortográfica. Es adecuada para proyectos pequeños o aplicaciones que no requieran complejas personalizaciones, pero su rendimiento y capacidad son limitados en comparación con NLTK y SpaCy.

Gensim está especializado en el modelado de temas y la creación de representaciones vectoriales de texto, como la técnica de Word2Vec. Su fortaleza radica en el análisis de grandes volúmenes de datos y en la modelización semántica, pero no está diseñado para tareas generales de NLP, lo que lo hace menos versátil en comparación con las otras librerías.

Cada librería tiene su lugar dependiendo de la tarea y los requerimientos del proyecto, desde la flexibilidad de NLTK hasta la eficiencia y especialización de SpaCy y Gensim.

¿Cómo afectó la normalización a los resultados obtenidos en las diferentes técnicas de NLP? (BoW, análisis de sentimiento, POS tagging, etc)

La normalización del texto tiene un impacto fundamental en los resultados de diversas técnicas de procesamiento de lenguaje natural (NLP) como el modelo Bag of Words (BoW), el análisis de sentimiento y el POS tagging.

En el caso de BoW, la normalización, que incluye la conversión a minúsculas, la eliminación de puntuación y stop words, y la lematización, ayuda a reducir el ruido y a mejorar la calidad del modelo. Esto asegura que palabras similares se traten como una sola, mejorando la eficiencia y precisión del modelo. Por ejemplo, "Perro" y "perro" serían representadas como una única palabra, evitando la duplicación innecesaria.

En el análisis de sentimiento, la normalización también juega un papel crucial. Eliminar las stop words y los errores ortográficos permite que el modelo se concentre en las palabras clave que aportan el mayor valor al sentimiento general del texto. Al lematizar las palabras,

por ejemplo, "felices" y "felicidad" pueden ser reconocidas como una misma idea positiva, lo que mejora la precisión en la clasificación de los sentimientos. Esto evita que las variaciones ortográficas o morfológicas de una palabra sean malinterpretadas.

En el POS tagging (etiquetado de partes del discurso), la normalización, y especialmente la lematización, facilita una asignación más precisa de las etiquetas gramaticales, ya que las palabras se reducen a su forma base. Sin embargo, es importante considerar que la eliminación de números o stop words podría afectar el análisis en contextos donde estos elementos son esenciales para el significado del texto.

La normalización mejora la consistencia y la relevancia de los datos, lo que lleva a resultados más precisos en todas estas técnicas, aunque debe aplicarse con cuidado para no perder información crucial.

PRÁCTICA 4. REFLEXIÓN

1. Cual es la diferencia entre que analice tu código un compilador y modelo de lenguaje.

Un compilador está diseñado específicamente para el procesamiento de lenguajes de programación en un formato ejecutable por una máquina. Analiza el código de manera técnica y verifica si cumple con las reglas sintácticas y semánticas del lenguaje. El objetivo principal es que sea ejecutable y, a su vez, eficiente, aunque en este caso no nos ofrece explicaciones ni consejos más allá de detectar los errores correspondientes.

Por otro lado en un modelo de lenguaje como Chat GPT, se utiliza el análisis del código desde un punto de vista contextual y lingüístico identificando errores basado en ejemplos aprendidos. Este además puede ofrecer soluciones creativas y explicaciones completas, mejorando su respuesta y la comprensión del usuario.

2. Implicaciones éticas y prácticas del uso de modelos de NLP entrenados para código.

El gran riesgo del uso de modelos de lenguaje y/o generar código es algo a debatir. Por un lado, estas herramientas pueden reproducir partes de código protegidas por derechos de autor si fueron entrenadas con ese material, lo que podría ocasionar problemas de plagio o conflictos de propiedad.

Además, existe la posibilidad de que los programadores dependan demasiado de estas herramientas, dejando de lado el aprendizaje de conceptos fundamentales de programación. Otro tema preocupante es que los modelos pueden arrastrar sesgos presentes en los datos de entrenamiento, lo que podría generar malas prácticas o limitar las posibles soluciones. Para evitar estos problemas, es clave comprobar los datos, establecer controles en la generación de código y usarlos de forma responsable como apoyo, no como un sustituto.