Mc Graw Hill

TEMA 2

Modelos avanzados de NLP. Enfoques con Deep learning

Deep learning aplicada:
NLP y visión artificial

ÍNDICE

1 Introducción a NLP

Fig. 1. Simulación de interacción entre un ordenador y el lenguaje humano natural (Imagen generada con IA)

INVESTIGA Y REFLEXIONA

¿Qué tareas implica que el NLP permita a los ordenadores «entender» y «hablar» el lenguaje humano?

El procesamiento de lenguaje natural (NLP por sus siglas en inglés, natural language processing) es una rama de la inteligencia artificial y la lingüística computacional que se centra en la interacción entre las computadoras y el **lenguaje humano natural,** como el inglés, español, francés, etc. Su objetivo principal es permitir a las máquinas **entender, interpretar, manipular y generar texto** o habla de manera similar a como lo hacen los humanos.

En términos simples, el NLP permite a las computadoras «entender» y «hablar» el lenguaje humano. Esto implica diversas tareas:

- **Comprensión del lenguaje:** implica la capacidad de las máquinas para entender y analizar el lenguaje humano en diferentes formas, como la identificación de palabras clave, el análisis sintáctico, la semántica y la extracción de información.

- **Generación del lenguaje:** es la capacidad de las máquinas para producir texto humano legible y coherente. Esto puede incluir tareas como la generación de resúmenes, las traducciones automáticas y la creación de diálogos.

Los **componentes fundamentales** del NLP son los siguientes:

- **Tokenización:** divide el texto en unidades más pequeñas, como palabras o frases.
- **Análisis sintáctico y semántico:** desglosa las estructuras gramaticales y comprende el significado del texto.
- **Extracción de entidades:** identifica y clasifica entidades relevantes, como nombres de personas, ubicaciones, organizaciones, etc.
- **Modelado del lenguaje:** representa el conocimiento lingüístico en forma de modelos estadísticos o de aprendizaje automático.

¿Cuáles son las aplicaciones prácticas del NLP?

- **Análisis de sentimientos:** evalúa la opinión o actitud expresada en el texto.
- **Asistentes virtuales y chatbots:** interactúan y responden a las consultas humanas en lenguaje natural.
- **Traducción automática:** convierte el texto de un idioma a otro.
- **Resumen automático de texto:** crea resúmenes breves y precisos de documentos largos.
- **Extracción de información:** recupera información específica de grandes conjuntos de datos textuales.

MÁS INFORMACIÓN

Accede al documento sobre Procesamiento del Lenguaje Natural para saber más sobre las aplicaciones prácticas del NLP.

1.1. Herramientas y bibliotecas principales en NLP

En el ámbito del procesamiento de lenguaje natural, hay varias herramientas y bibliotecas principales que facilitan el desarrollo de aplicaciones, investigación y análisis de texto. Estas bibliotecas proporcionan una serie de funcionalidades y algoritmos para realizar tareas como tokenización, análisis sintáctico, análisis semántico o modelado del lenguaje, entre otros. A continuación se muestran algunas de las herramientas y bibliotecas más populares en el campo del NLP:

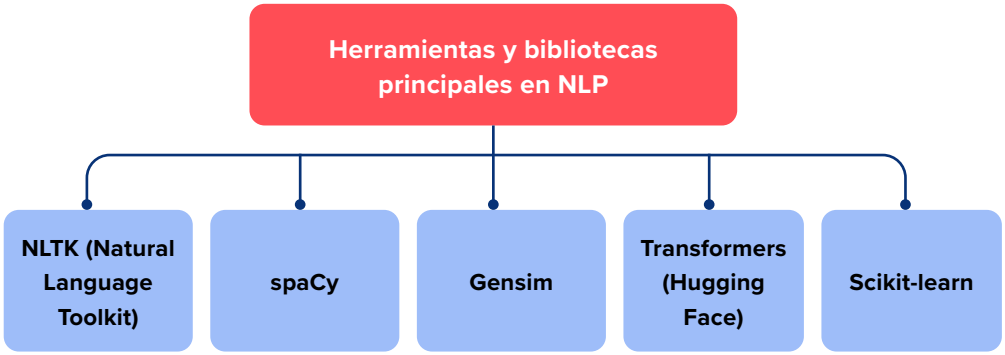

Figura 2. Herramientas y bibliotecas principales en procesamiento natural del lenguaje.

WEB

Accede a este artículo del Ministerio para la transformación digital y de la función pública donde encontrarás 10 librerías poulares de procesamiento del lenguaje natural.

INVESTIGA Y REFLEXIONA

Además de las bibliotecas expuestas en este epígrafe: ¿Qué otras bibliotecas conoces que favorezcan el procesamiento del lenguaje natural?

1. NLTK (Natural Language Toolkit)

NLTK es una **biblioteca de Python** muy utilizada para el procesamiento de texto y NLP.Proporciona herramientas y recursos para trabajar con texto, como tokenización, stemming, lematización, análisis sintáctico y semántico, y modelos de lenguaje preentrenados.

- Ofrece una **amplia gama de módulos** para tareas de NLP, así como acceso a corpus lingüísticos y herramientas para procesamiento de texto.

2. spaCy

spaCy es otra **biblioteca de procesamiento de texto** en Python. Se centra en la eficiencia, la velocidad y la facilidad de uso. Proporciona herramientas para tokenización, etiquetado gramatical, análisis de dependencias y reconocimiento de entidades nombradas (NER), entre otras tareas.

- Ofrece **modelos preentrenados para varios idioma**s y permite una integración sencilla con otras herramientas de aprendizaje automático.

3. Gensim

Gensim es una biblioteca de Python para **modelado de temas y vectores de palabras** (word vectors). Se utiliza principalmente para el análisis semántico de texto, incluyendo tareas como la creación de modelos de temas (topic modelling) y la generación de representaciones vectoriales de palabras.

➔ Ofrece **algoritmos eficientes** para la creación de modelos de tópicos, así como la implementación de técnicas de embedding de palabras como Word2Vec.

4. Transformers (Hugging Face)

Transformers es una biblioteca desarrollada por Hugging Face que proporciona acceso a **modelos de lenguaje preentrenados** de vanguardia, como BERT, GPT y otros. Ofrece una API fácil de usar para utilizar estos modelos en diversas tareas de NLP.

➔ Permite la adaptación y el ajuste fino (fine-tuning) para tareas específicas.

MÁS INFORMACIÓN

Para entender mejor en qué consiste y cuándo se utiliza el NLP, accede al siguiente documento: Definición y contexto de NLP.

5. Scikit-learn

Aunque Scikit-learn es principalmente conocida por el aprendizaje automático, también proporciona herramientas para tareas de **procesamiento de texto** y NLP.

➔ Ofrece funcionalidades para vectorización de texto, clasificación, clustering y otras tareas relacionadas con NLP. Además, incluye algoritmos simples pero eficaces para tareas de procesamiento de texto y aprendizaje automático.

Estas son solo algunas de las herramientas y bibliotecas populares en el campo del procesamiento de lenguaje natural. Cada una tiene sus propias fortalezas y se utiliza para diferentes tareas según las necesidades del proyecto.

VIDEO

Para consultar el siguiente vídeo sobre **Comparación de rostros con AWS Rekognition,** escanea el código QR o pulsa aquí.

2 Técnicas de pre procesamiento de texto

El preprocesamiento de texto es una etapa fundamental en el procesamiento de lenguaje natural. Consiste en una serie de **pasos que se aplican al texto sin procesar** antes de realizar análisis o tareas de modelado. Estos pasos ayudan a convertir el texto en una forma **más estructurada y significativa** para que los algoritmos de NLP puedan trabajar de manera más efectiva.

Algunas de las técnicas comunes de preprocesamiento de texto incluyen la tokenización, la normalización y la limpieza del texto. Las vemos a continuación:

2.1. Tokenización

La tokenización es el proceso de **dividir el texto en unidades más pequeñas** llamadas tokens. Estos tokens pueden ser palabras individuales, frases o incluso caracteres, dependiendo del nivel de granularidad requerido.

➔ **Ejemplo:** si se tiene la frase «El gato está durmiendo», la tokenización podría dividirla en tokens individuales como [«El», «gato», «está», «durmiendo»].

2.2. Normalización

La normalización es la estandarización del texto, lo que implica la reducción de las variaciones de palabras a una forma común. Esto puede incluir la conversión de texto a minúsculas, la eliminación de puntuaciones o caracteres especiales, y la corrección de errores ortográficos.

➔ **Ejemplo:** convertir «Caminar» y «caminando» a su forma base «caminar».

INVESTIGA Y REFLEXIONA

¿Sabrías identificar la utilidad de las diferentes técnicas de procesamiento de texto? Aparte de las aquí expuestas, ¿qué otras técnias conoces?

2.3. Limpieza de texto

La limpieza de texto implica la eliminación de elementos no deseados, como etiquetas HTML, números, signos de puntuación, enlaces, emojis o cualquier otro tipo de ruido que no sea relevante para el análisis.

➔ **Ejemplo:** eliminar números, signos de puntuación y caracteres especiales.

2.4. Ejemplo de preprocesamiento de texto

A continuación, se muestra un ejemplo simple de preprocesamiento de texto utilizando la biblioteca NLTK en Python, que incluye tokenización, normalización y limpieza de texto:

```
import nltk
from nltk.tokenize import word_tokenize
import string
from nltk.corpus import stopwords
# Ejemplo de texto
texto = «¡Hola! Este es un ejemplo de texto. Contiene
signos de puntuación, mayúsculas y minúsculas.»
# Tokenización
tokens = word_tokenize(texto.lower()) # Convertir
texto a minúsculas y tokenizar
# Normalización
tokens = [token for token in tokens if token.
isalpha()]
# Eliminar tokens que no son palabras # Limpieza de
texto
stop_words =     set(stopwords.words('spanish'))
# Obtener stopwords en español
tokens = [token for token in tokens if token not in
stop_words]
# Eliminar stopwords
print(tokens)
```

MÁS INFORMACIÓN

Para entender mejor en qué consiste y el preprocesameinto de texto y cómo utilizarlo en RR.SS, accede al siguiente documento: Agentes Inteligentes y Web Semántica: Preprocesamiento de Texto de Redes Sociales.

Este ejemplo muestra una serie de pasos de preprocesamiento de texto básicos utilizando NLTK en Python. Incluye tokenización, normalización (convirtiendo a minúsculas y eliminando tokens que no son palabras) y limpieza (eliminación de stopwords). Estos pasos son fundamentales para preparar el texto antes de realizar análisis más avanzados en tareas de NLP.

INVESTIGA Y REFLEXIONA

¿Sabrías ordenar los pasos esenciales para preparar el texto antes de realizar análisis avanzados en tareas NLP? ¿Cuántos pasos debes realizar? ¿En qué orden?

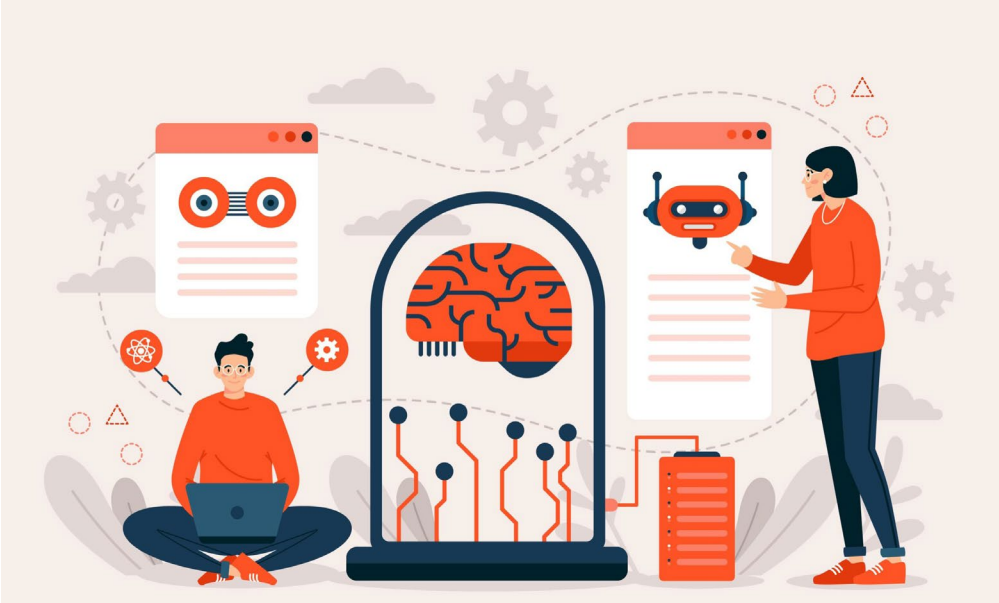

Fig. 3. El preprocesamiento de texto es una etapa fundamental en el procesamiento de lenguaje natural.

VIDEO

Para consultar el siguiente vídeo sobre el **Uso de Auto ML con AWS Sagemarker Experiments,** escanea el código QR o pulsa aquí.

Caso de Estudio

Fundamentos de machine learning. Entrenar con AWS SageMaker

En este caso de estudio, aprenderás a utilizar Amazon SageMaker para crear, entrenar e implementar un modelo de machine learning (ML) por medio del algoritmo de ML XGBoost. Amazon SageMaker es un servicio completamente administrado que brinda a todos los científicos de datos y desarrolladores la capacidad de crear, entrenar e implementar modelos de machine learning rápidamente.

En general, trasladar los modelos de machine learning desde la etapa de conceptualización hasta la etapa de producción es sumamente complejo e implica mucho tiempo. Debes administrar grandes cantidades de datos para entrenar el modelo, elegir el mejor algoritmo posible para entrenarlo, administrar la capacidad de cómputo mientras se lo entrena y, luego, implementar el modelo en un entorno de producción. Amazon SageMaker reduce esta complejidad y facilita de manera significativa la creación y la implementación de modelos de machine learning. Una vez que elige los algoritmos y los marcos adecuados de la amplia gama de opciones disponibles, SageMaker administra toda la infraestructura subyacente para entrenar su modelo a escala de petabytes e implementarlo en la producción.

En este caso práctico, asumirás el rol de un desarrollador de machine learning que trabaja en un banco. Se te solicita que desarrolles un modelo de machine learning para predecir si los clientes se inscribirán para un certificado de depósito.

Pasos a realizar:

1. Crea una instancia de cuaderno de Amazon SageMaker.
2. Prepara los datos.
3. Entrena el modelo ML.
4. Implementa el modelo.
5. Evalúa el rendimiento del modelo.
6. Elimina los recursos.

Reflexiona y responde a las siguientes cuestiones*:

1. **¿Cuál es el principal objetivo de Amazon SageMaker?**
2. **¿Qué ventaja ofrece el algoritmo XGBoost en el contexto de predicción de inscripciones a certificados de depósito?**
3. **¿Cuál es el objetivo del modelo de machine learning a desarrollar?**
4. **¿Qué ventaja principal ofrece Amazon SageMaker en el proceso de desarrollo de modelos de ML?**
5. **Al utilizar Amazon SageMaker para este proyecto, ¿qué aspecto del desarrollo de modelos NO necesita gestionar directamente el desarrollador?**

** Responde a estas preguntas de manera interactiva en el contenido digital.*


Mc
Graw
Hill