TP4 - Redes neuronales - Pysentimiento

Definición de NLP

El Procesamiento de Lenguaje Natural son todas las tareas de la inteligencia artificial que tiene como entrada el lenguaje humano. Su *objetivo principal* es permitir a las máquinas comprender, interpretar y generar texto y voz en un formato que sea comprensible y útil para los seres humanos. El NLP permite:

1. Comunicación Máquina-Humano: Permite la comunicación efectiva entre las máquinas y los seres humanos a través del lenguaje natural.

2. Automatización de Tareas Lingüísticas: Facilita la automatización de tareas que requieren procesamiento y comprensión del lenguaje, como la clasificación de documentos, la extracción de información de texto no estructurado y la generación de resúmenes automáticos.

3. Análisis de Datos Textuales: Ayuda a las organizaciones a extraer información valiosa de grandes volúmenes de datos textuales, lo que es esencial para la toma de decisiones.

4. Traducción Automática: Facilita la traducción de texto y voz entre diferentes idiomas.

5. Accesibilidad: Ayuda a mejorar la accesibilidad al permitir que personas con discapacidades se comuniquen.

Aplicaciones de NLP en la vida cotidiana.

1. Asistentes Virtuales: Los asistentes virtuales como Siri (Apple), Alexa (Amazon) y Google Assistant (Google) utilizan NLP para comprender y responder a comandos de voz y preguntas en lenguaje natural.

2. Búsqueda en Internet: Los motores de búsqueda como Google utilizan algoritmos de NLP para comprender la intención del usuario y ofrecer resultados relevantes. Esto incluye la corrección de errores ortográficos y la interpretación de preguntas complejas.

3. Traducción automática: Servicios como Google Translate utilizan NLP para traducir texto de un idioma a otro.

4. Clasificación de Correos Electrónicos y Spam: Los filtros de spam de correo electrónico utilizan NLP para identificar correos no deseados, analizando el contenido y las características del mensaje.

5. Detección de Fraude: Los sistemas de detección de fraude en transacciones financieras utilizan NLP para analizar patrones de comportamiento y texto en tiempo real, identificando actividades sospechosas.

6. Chatbots y Atención al Cliente: Las empresas utilizan chatbots con capacidades de NLP para proporcionar respuestas automáticas a preguntas frecuentes y brindar asistencia al cliente en línea.

Fundamentos del Procesamiento de Lenguaje Natural

El procesamiento de lenguaje natural (NLP) implica el análisis y procesamiento de texto en lenguaje natural, pero antes de que las máquinas puedan comprender y trabajar con el texto, es necesario realizar una serie de pasos de tokenización y preprocesamiento.

Tokenización: es el proceso de dividir un texto en unidades más pequeñas llamadas "tokens". Estos tokens pueden ser palabras, frases, símbolos o incluso caracteres individuales, dependiendo del nivel de granularidad requerido para una tarea específica facilitando el análisis y la comprensión del contenido.

Ejemplo de Tokenización:

Frase de entrada: "Me voy a sacar un 10 en la presentación de IA.”
Tokens resultantes: ["Me”, “voy”, “a”, “sacar”, “un”, “10”, “en”, “la”, “presentación”, “de”, “IA”, “.”]

Preprocesamiento de Texto: es el conjunto de tareas que se realizan para limpiar y estandarizar el texto antes de su análisis. Estas tareas pueden incluir la eliminación de signos de puntuación, la conversión a minúsculas, la eliminación de palabras vacías, la lematización y la eliminación de caracteres especiales garantizando que el texto esté en un formato coherente y que se eliminen elementos que no aportan información relevante para la tarea en cuestión.

Ejemplo de Preprocesamiento de Texto:

Texto original: "La analítica de datos es una disciplina apasionante, con un gran potencial de crecimiento."

Texto preprocesado: "analítica datos disciplina apasionante gran potencial crecimiento"

Análisis Morfológico: se refiere al proceso de descomponer una palabra en sus partes constituyentes, conocidas como morfemas. Los morfemas son las unidades más pequeñas de significado en una palabra, y pueden ser prefijos, sufijos, raíces, entre otros, que ayudan a identificar las características gramaticales y semánticas de una palabra para comprender cómo las palabras se conjugan, flexionan o cambian en diferentes contextos gramaticales para una correcta interpretación.

Ejemplo de Análisis Morfológico:

Palabra: "corriendo"
Morfemas: ["correr" (raíz), "-iendo" (sufijo de gerundio)]

Lematización: es el proceso de reducir una palabra a su forma base, conocida como lema. El lema es la forma canónica o diccionario de una palabra que representa su significado fundamental. La lematización es útil para tratar diferentes formas flexionadas de una palabra como si fueran la misma palabra base. Esto sirve para agrupar palabras flexionadas bajo su forma lematizada, lo que facilita la comparación y el análisis de palabras relacionadas.

Ejemplo de Lematización:

Palabras: "corro", "corriendo", "correrá"
Lemas: "correr"

Representación vectorial de texto (Word Embeddings).

La representación vectorial de texto, comúnmente conocida como "Word Embeddings" o "incrustación de palabras", que consiste en asignar a cada palabra en un vocabulario una representación numérica (vector) en un espacio de características, de modo que las palabras con significados y contextos similares estén cerca unas de otras en ese espacio vectorial.

1. Similaridad Semántica:

La distancia entre vectores en el espacio vectorial refleja la similitud semántica entre las palabras. Las palabras que son sinónimos o que se utilizan en contextos similares tendrán vectores más cercanos en el espacio.

2. Significado y Contexto:

Las representaciones vectoriales de palabras se generan a partir del contexto en el que aparecen las palabras en grandes cuerpos de texto. Las palabras que suelen aparecer en contextos similares tendrán vectores similares, lo que captura el significado y las relaciones semánticas entre las palabras.

3. Dimensionalidad:

Cada palabra se representa en un espacio vectorial de alta dimensionalidad, actualmente los vectores generados por la segunda generación de modelos de openai (ada-002) poseen 1536 dimensiones.

4. Representación Distribuida:

En lugar de representar palabras como índices o etiquetas, las representaciones vectoriales asignan a cada palabra un vector numérico denso. Esto significa que cada elemento del vector contiene información relevante sobre la palabra y su contexto.

5. ¿Cómo se hace la comparación?

Por ejemplo, supongamos que tenemos el vector guardado en una base de datos, lo que se suele hacer es calcular la distancia entre los vectores, calcular el producto punto entre los vectores o calcular la similitud de coseno entre los vectores.

En un espacio vectorial, las representaciones de las palabras "gato" y "perro" pueden estar más cerca entre sí que las palabras "gato" y "coche", ya que los dos primeros comparten un contexto semántico similar.

Modelos de Lenguaje: son sistemas que tienen como objetivo calcular la probabilidad de que una secuencia de palabras ocurra en un cierto contexto, lo que implica entender la estructura y el significado del lenguaje.

Los modelos de lenguaje tradicionales, como los basados en n-gramas o cadenas de Markov, tenían limitaciones en su capacidad para capturar la semántica y las relaciones de largo alcance en el texto.

Transformers: son una arquitectura de red neuronal basada en un mecanismo de atención que permite que el modelo considere todas las palabras en una secuencia simultáneamente, en lugar de procesarlas secuencialmente.

Los Transformers han sido fundamentales para el desarrollo de modelos de lenguaje preentrenados a gran escala, como BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer) y otros.

Modelos de Lenguaje basados en Transformers: una característica clave de los modelos de lenguaje preentrenados basados en Transformers es su capacidad de transferir conocimientos. Estos modelos se entrenan en grandes cuerpos de texto y luego se ajustan a tareas específicas con datos adicionales (fine-tuning).

III. Aplicaciones de NLP

Análisis de sentimientos

El análisis de sentimientos es la tarea de determinar si un fragmento de texto (como un comentario, una reseña o un tweet) tiene una polaridad positiva, negativa o neutral. En otras palabras, busca identificar si el autor del texto está expresando una opinión favorable, desfavorable o neutra sobre un tema o entidad específica.

Importancia:

El análisis de sentimientos es ampliamente utilizado en la industria para comprender cómo se sienten los clientes acerca de productos, servicios o experiencias. Las organizaciones pueden utilizarlo para monitorear la reputación en línea, mejorar la toma de decisiones y la satisfacción del cliente, y adaptar sus estrategias en función de los comentarios de los usuarios.

PySentimiento

¿Qué es y por qué es relevante?

PySentimiento es una biblioteca de análisis de sentimientos en Python que permite determinar la polaridad de un fragmento de texto, es decir, si el texto expresa un sentimiento positivo, negativo o neutral. La biblioteca utiliza modelos de lenguaje preentrenados y técnicas de aprendizaje automático para llevar a cabo esta tarea.

Relevancia de PySentimiento:

Facilidad de Uso: se pueden analizar sentimientos con solo unas pocas líneas de código, lo que facilita su implementación en aplicaciones y proyectos.

Open Source: PySentimiento es una biblioteca de código abierto, lo que significa que los desarrolladores pueden acceder al código fuente y contribuir a su desarrollo. Esto fomenta la colaboración y permite a la comunidad mejorar y ampliar la funcionalidad de la biblioteca.

Ejemplos de uso de PySentimiento

```
pip install pysentimiento
```

```
from pysentimiento import create_analyzer
analyzer = create_analyzer(task="sentiment", lang="es")

analyzer.predict("Qué gran jugador es Messi")
# returns AnalyzerOutput(output=POS, probas={POS: 0.998, NEG: 0.002,
NEU: 0.000})

emotion_analyzer = create_analyzer(task="emotion", lang="en")

emotion_analyzer.predict("fuck off")
# returns AnalyzerOutput(output=anger, probas={anger: 0.798, surprise:
0.055, fear: 0.040, disgust: 0.036, joy: 0.028, others: 0.023, sadness:
0.019})
```

Cómo funciona PySentimiento por detrás

Modelos de Lenguaje Preentrenados:

La base de PySentimiento son los modelos de lenguaje preentrenados. Estos modelos, como BERT y RoBERTa, se entrenan en grandes conjuntos de datos en diversos idiomas y capturan el conocimiento sobre el lenguaje natural y las relaciones semánticas.

Tareas de Clasificación de Texto:

Está diseñado para tareas de clasificación de texto, donde el objetivo es determinar la polaridad de un fragmento de texto (positiva, negativa o neutral). La biblioteca utiliza un enfoque de "relleno" para completar una frase o contexto inicial y luego realiza la clasificación basada en el modelo de lenguaje.

Tokenización y Procesamiento de Texto:

Convierte el texto de entrada en tokens (unidades más pequeñas) y realiza el preprocesamiento necesario para alimentar los modelos de lenguaje preentrenados.

Clasificación de Sentimientos:

Una vez que el texto se ha procesado y tokenizado, utiliza el modelo de lenguaje preentrenado para realizar la clasificación de sentimientos. El modelo asigna una etiqueta de sentimiento (positivo, negativo o neutral) al texto y proporciona probabilidades asociadas a cada etiqueta.

Salida del Análisis de Sentimientos:

La salida del análisis de sentimientos incluye la polaridad (positiva, negativa o neutral) y las probabilidades asociadas a cada categoría.

Configuración Personalizada:

PySentimiento permite la configuración personalizada, lo que significa que los usuarios pueden especificar el modelo de lenguaje preentrenado, así como los umbrales de polaridad (positivo y negativo) para adaptar el análisis de sentimientos según sus necesidades.

Extensibilidad:

Entrenamiento inicial: Los modelos de lenguaje preentrenados se entrenan en grandes cuerpos de texto sin etiquetas, específicamente, pysentimiento utiliza un dataset de más de 600 millones de tweets. Durante este entrenamiento, los modelos intentan aprender la estructura del lenguaje natural, las relaciones semánticas y la gramática.

Representaciones vectoriales: Los modelos de lenguaje preentrenados generan representaciones vectoriales de las palabras, donde cada palabra se representa como un vector numérico en un espacio multidimensional. Estos vectores capturan información sobre el significado y el contexto de las palabras.

Afinamiento en tareas específicas: Después del entrenamiento inicial, los modelos de lenguaje preentrenados se afinan o ajustan en tareas específicas, como el análisis de sentimientos. Esto implica proporcionar datos etiquetados (textos con etiquetas de sentimiento) y adaptar el modelo para que sea capaz de realizar tareas de clasificación de texto.

Uso de PySentimiento

En PySentimiento, se utilizan modelos de lenguaje preentrenados para llevar a cabo el análisis de sentimientos. Esto se hace de la siguiente manera:

Selección del Modelo: PySentimiento permite a los usuarios seleccionar un modelo de lenguaje preentrenado específico para la tarea de análisis de sentimientos. Ejemplos comunes de modelos incluyen BERT, RoBERTa, GPT-2 y otros.

Configuración del Modelo: Se puede configurar PySentimiento para utilizar un modelo específico proporcionando el nombre o la ruta del modelo preentrenado deseado.

Análisis de Sentimientos: Una vez configurado, PySentimiento utiliza el modelo preentrenado para analizar el sentimiento en un fragmento de texto. Por ejemplo:

El resultado incluirá la polaridad del sentimiento (positivo, negativo o neutral) y las probabilidades asociadas a cada categoría.

Problemas Comunes

Ambigüedad y Sarcasmo: El lenguaje humano es inherentemente ambiguo y a menudo se utiliza sarcasmo o ironía. Esto puede dificultar la clasificación precisa de sentimientos, ya que un comentario sarcástico podría ser interpretado incorrectamente.

Matices del Lenguaje: El análisis de sentimientos a menudo no captura matices en el lenguaje. Por ejemplo, un comentario neutral que menciona una tragedia podría ser malinterpretado como negativo.

Desequilibrio de Clases: En conjuntos de datos, a menudo hay un desequilibrio entre las clases de sentimiento, con una predominancia de ejemplos neutrales. Esto puede afectar la capacidad del modelo para clasificar correctamente sentimientos positivos o negativos.

Generalización Limitada: Los modelos preentrenados a menudo se entrenan en un conjunto diverso de datos, pero pueden no generalizar bien a dominios o idiomas específicos. Se requiere afinamiento y adaptación para mejorar la precisión.

Sesgo en los Datos de Entrenamiento: Los datos de entrenamiento utilizados para modelos de lenguaje pueden contener sesgos culturales, de género, étnicos u otros. Estos sesgos se reflejan en las predicciones del modelo y pueden llevar a decisiones sesgadas.

Sesgo en el Lenguaje Natural: El lenguaje natural en sí mismo puede ser sesgado y reflejar prejuicios y estereotipos. Los modelos de lenguaje preentrenados pueden aprender y propagar estos sesgos.

Privacidad de los Usuarios: El procesamiento de texto a menudo involucra el análisis de datos personales o privados. Es fundamental garantizar la privacidad de los usuarios y cumplir con las regulaciones de protección de datos.

Consentimiento y Transparencia: Es importante obtener el consentimiento de los usuarios antes de analizar sus datos de texto y ser transparente sobre cómo se utilizarán sus datos.

Responsabilidad Ética: Los desarrolladores y empresas que utilizan el análisis de sentimientos deben considerar las implicaciones éticas de sus aplicaciones. Esto incluye evitar la discriminación y el sesgo, así como utilizar la tecnología de manera responsable.