TIC

TALENTO TECH

Misión 3

INTELIGENCIA ARTIFICIAL

Explorador

Tema 1: Procesamiento del Lenguaje Natural

UTraining Entrenamiento en Innovación y Tecnología | UNIVERSIDAD DE ANTIOQUIA | Universidad de Caldas | ubicua Technology

Campista, llegó el momento de retar tus conocimientos y que los pongas aprueba a través de los diferentes recursos que encontraras en este espacio como son: conceptos, ejemplos, herramientas, actividades prácticas y retos, los cuales te ayudaran alcanzar los objetivos trazados en el nivel explorador.

LIMPIEZA Y PREPROCESAMIENTO DE DATOS

Limpieza y preprocesamiento de datos

La limpieza y preprocesamiento de datos es una etapa esencial en el desarrollo de aplicaciones de aprendizaje de máquina, especialmente en el procesamiento de lenguaje natural (NLP). Esta fase se encarga de preparar los datos textuales para que los modelos puedan analizarlos y aprender de ellos de manera efectiva. A través de diversas técnicas como **la tokenización, normalización, eliminación de stopwords, lematización y stemming**, se transforma y simplifica el texto, eliminando ruido y estandarizando la información. Esto es fundamental para mejorar la precisión y eficiencia en tareas como la clasificación de texto, análisis de sentimientos, y la implementación de chatbots que interactúan de manera natural con los usuarios.

- **La Tokenización**

La tokenización es el proceso de dividir un texto en unidades más pequeñas, llamados tokens. Estos pueden ser palabras, frases o caracteres.

  - **Ejemplo:**

    Si un usuario escribe "Quiero hacer un reclamo sobre mi última compra", la tokenización separaría la oración en tokens como ["Quiero", "hacer", "un", "reclamo", "sobre", "mi", "última", "compra"]. Esto facilita al chatbot identificar que el mensaje trata de un "reclamo".

- **Normalización**

La normalización transforma las palabras a una forma estándar, como convertir todo el texto a minúsculas, eliminar puntuación o normalizar caracteres.

- **Ejemplo:**

  En el mensaje "QUIERO HACER UN RECLAMO!!!", la normalización convertiría el texto a "quiero hacer un reclamo", eliminando mayúsculas y signos de exclamación, para que el chatbot procese el texto de manera uniforme.

- **Eliminación de Stopwords**

El proceso de eliminación de stopwords consiste en remover palabras comunes que no añaden valor significativo al análisis, como "el", "y", "de".

- **Ejemplo:**

  En la frase "Quiero hacer un reclamo sobre el servicio", las stopwords como "quiero", "un", "sobre", "el" podrían eliminarse, dejando "hacer reclamo servicio". Esto ayuda al chatbot a centrarse en las palabras clave "reclamo" y "servicio".

- **Lematización**

La lematización reduce las palabras a su forma base o lema, teniendo en cuenta el contexto y la gramática..

- **Ejemplo:**

  Si el chatbot recibe la entrada "Estoy enviando una petición", la lematización podría convertir "enviando" en "enviar" y "petición" en "petición", ayudando a identificar la acción principal como "enviar" y la categoría como "petición".

- **Eliminación de Stemming**

El stemming reduce las palabras a su raíz eliminando prefijos y sufijos, sin considerar el contexto gramatical.

- **Ejemplo:**

  En la frase "Estoy solicitando información", el stemming podría convertir "solicitando" en "solicit", ayudando al chatbot a reconocer la intención principal del usuario, que es "solicitar".

APRENDIZAJES PRÁCTICOS

Estos ejercicios permitirán a los estudiantes aplicar los conceptos aprendidos en situaciones prácticas y específicas.

Práctica: Limpieza, Preprocesamiento y Representación de Datos para Análisis de Sentimientos

Objetivo:

En esta práctica, trabajarán con un conjunto de datos de texto para realizar limpieza, preprocesamiento y representación necesarios para un análisis de sentimientos. Aplicarán técnicas clave como tokenización, normalización, eliminación de stopwords, lematización, stemming, y representarán el texto usando Bolsa de Palabras y TF-IDF.

INSTRUCCIONES

1. Carga del Conjunto de Datos:

- Descarga el conjunto de datos de comentarios de productos o reseñas de clientes desde una plataforma recomendada (por ejemplo, Kaggle).
- Carga el conjunto de datos en un entorno de trabajo, como Jupyter Notebook.

2. Tokenización:

- Divide el texto en tokens (palabras o frases individuales).
- Utiliza bibliotecas como nltk o spaCy en Python para realizar la tokenización.

3. Normalización:

- Convierte todo el texto a minúsculas para asegurar la consistencia.
- Elimina caracteres especiales, números y puntuación innecesaria utilizando expresiones regulares.

4. Eliminación de Stopwords:

- Identifica y elimina las palabras comunes que no aportan valor semántico (stopwords) del texto.
- Utiliza listas de stopwords disponibles en bibliotecas como nltk o spaCy.

5. Lematización:

- Reduce las palabras a su forma base o raíz (lemmas) utilizando herramientas como WordNetLemmatizer en nltk o el componente de lematización en spaCy.

6. Stemming:

- Alternativamente, realiza stemming para reducir las palabras a sus raíces (stem) utilizando el PorterStemmer o SnowballStemmer en nltk.

7. Representación de Texto:

- Bolsa de Palabras (Bag of Words):
  - Utiliza CountVectorizer de sklearn para convertir el texto preprocesado en una representación de Bolsa de Palabras.
- TF-IDF (Term Frequency-Inverse Document Frequency):
  - Aplica TfidfVectorizer de sklearn para representar el texto con TF-IDF, que evalúa la importancia de una palabra en relación con los documentos del conjunto de datos.

8. Aplicación Práctica:

- Aplica todas las técnicas de preprocesamiento al conjunto de datos.
- Compara las representaciones de Bolsa de Palabras y TF-IDF y discute sus diferencias y aplicaciones en el análisis de sentimientos.

9. Evaluación de Resultados:

- Muestra ejemplos de texto antes y después del preprocesamiento.
- Compara y analiza cómo las representaciones de Bolsa de Palabras y TF-IDF afectan la calidad del análisis de sentimientos.

10. Informe:

- Elabora un informe que incluya:
    - Una descripción del conjunto de datos utilizado.
    - Los pasos seguidos en el preprocesamiento y representación del texto.
    - Resultados y ejemplos de cómo cada técnica afecta el texto y el análisis de sentimientos.
    - Reflexiones sobre la eficacia de Bolsa de Palabras frente a TF-IDF.

Entrega:

Carga el Jupyter Notebook con el código y resultados, junto con el informe en formato PDF, en la plataforma de entrega indicada por el profesor.

Campista, en este espacio encontraras ayudas en diferentes formatos que pueden potenciar tu proceso de aprendizaje.

- *Intro al Natural Language Processing (NLP) #1 - ¡De PALABRAS a VECTORES!*
  https://www.youtube.com/watch?v=Tg1MjMIVArc

- *Words: Types, Tokens, & Tokenization | TTIC 31190 (NLP) - Fall 2020*
  https://www.youtube.com/watch?v=sXbDXE_uD2s

- *Natural Language Processing - Tokenization (NLP Zero to Hero - Part 1)*
  https://youtu.be/fNxaJsNG3-s?si=IGPD757UD3xr0sYK

REPRESENTANCIÓN DE TEXTO

La representación de texto es crucial en el procesamiento de lenguaje natural (NLP), ya que convierte datos textuales en formas que los algoritmos de machine learning pueden analizar y procesar. Dos enfoques comunes para la representación de texto son **Bolsa de Palabras (Bag of Words)** y **TF-IDF (Term Frequency-Inverse Document Frequency)**. A continuación, se explica cada uno de estos enfoques con un paso a paso detallado para calcular los datos.

- **Bolsa de Palabras (Bag of Words)**

1. **Recolección de Datos:**
   - Supongamos que tenemos tres mensajes de un chatbot:
     - *Mensaje 1:* "Hola, necesito ayuda con mi pedido."
     - *Mensaje 2:* "¿Cómo puedo presentar una queja sobre el servicio?"
     - *Mensaje 3:* "Gracias por tu asistencia."

2. **Construcción del Vocabulario:**
   - Identifica todas las palabras únicas en el conjunto de documentos. Ignora la puntuación y convierte todas las palabras a minúsculas.

- *Vocabulario:* ["hola", "necesito", "ayuda", "con", "mi", "pedido", "cómo", "puedo", "presentar", "una", "queja", "sobre", "el", "servicio", "gracias", "por", "tu", "asistencia"]

3. **Creación de la Matriz de Bolsa de Palabras:**
   - Para cada documento, cuenta la frecuencia de cada palabra del vocabulario. Luego, representa cada documento como un vector en el que cada dimensión corresponde a la frecuencia de una palabra del vocabulario.

4. **Uso del Vector:**
   - Cada vector resultante se puede utilizar como entrada para algoritmos de machine learning para tareas como clasificación o clustering.

--> Matriz de bolsa de palabras:

Mensaje 1	1	1	1	1	1	1	0	0	0
Mensaje 2	0	0	0	0	0	0	1	1	1
Mensaje 3	0	0	0	0	0	0	0	0	0

- **TF-IDF (Term Frequency-Inverse Document Frequency) --> PENDIENTE**

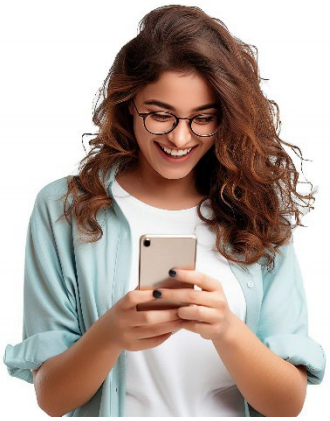
1. **Recolección de Datos:**
   - Usa el mismo conjunto de documentos:
     - *Documento 1:* "Hola, necesito ayuda con mi pedido."
     - *Documento 2:* "¿Cómo puedo presentar una queja sobre el servicio?"
     - *Documento 3:* "Gracias por tu asistencia."
2. **Cálculo de la Frecuencia de Término (TF):**
   - Recolección de Datos:
     - Usa el mismo conjunto de documentos:
       - Documento 1: "Hola, necesito ayuda con mi pedido."

- Documento 2: "¿Cómo puedo presentar una queja sobre el servicio?"
- Documento 3: "Gracias por tu asistencia."

- Cálculo de la Frecuencia de Término (TF):
  - Calcula la frecuencia de cada palabra en cada documento. La fórmula para TF de una palabra ttt en un documento ddd es:
- Número total de palabras en Documento 1: 6
- Frecuencia de "hola" en Documento 1: 1
- TF("hola", Documento 1) = 1 / 6 ≈ 0.167

$$\text{TF}(t,d) = \frac{\text{Número de veces que } t \text{ aparece en } d}{\text{Número total de palabras en } d}$$

ANÁLISIS DE SENTIMIENTOS UTILIZANDO MODELOS DE APRENDIZAJES DE MÁQUINA



El análisis de sentimientos es una tarea fundamental en el procesamiento de lenguaje natural (NLP) que implica clasificar textos en categorías de sentimientos como positivo, negativo o neutral. Los modelos de aprendizaje de máquina, como la **Regresión Logística y Naive Bayes**, son ampliamente utilizados para esta tarea. A continuación, se presenta una teoría general sobre cómo estos modelos pueden ser aplicados para clasificar tipos de PQRs (saludos, quejas, reclamos, peticiones y despedidas) basándose en el análisis de sentimientos.

Objetivos análisis de Sentimientos

Determinar la actitud expresada en un texto (por ejemplo, positiva, negativa o neutral). Esto es útil para clasificar tipos de PQRs y entender el sentimiento general hacia un servicio o producto.

- **Regresión Logística**

La regresión logística es un modelo de clasificación binaria que puede extenderse para manejar múltiples clases (por ejemplo, positivo, negativo, neutral). Utiliza una función logística para predecir la probabilidad de una clase específica dada una entrada. En el contexto de análisis de sentimientos, se ajusta a las características del texto para predecir el sentimiento.

Proceso de Clasificación:

1. Extracción de Características:

- Representación de Texto: Primero, se transforman los textos en vectores numéricos usando técnicas como Bolsa de Palabras o TF-IDF.
- Características: Cada vector representa la frecuencia o importancia de las palabras en el texto.

2. Entrenamiento del Modelo:

- Se entrena el modelo de regresión logística utilizando un conjunto de datos etiquetado con ejemplos de textos y sus sentimientos asociados.
- La regresión logística ajusta los pesos de las características para minimizar el error en la clasificación.

3. Predicción:

- Se utiliza el modelo entrenado para predecir el sentimiento de nuevos textos. El modelo calcula la probabilidad de cada clase y asigna el sentimiento con la probabilidad más alta.

Ejemplo de Aplicación:

- Saludos: Los mensajes como "Hola, ¿cómo estás?" podrían ser clasificados como positivos.

- *Quejas y Reclamos:* Los textos como "El servicio fue muy malo" podrían ser clasificados como negativos.
- *Peticiones:* Mensajes como "Necesito más información sobre el producto" podrían ser clasificados como neutrales.

- **Naive Bayes**

Naive Bayes es un modelo de clasificación probabilístico basado en el teorema de Bayes, que asume que las características (palabras) son independientes entre sí. Aunque la suposición de independencia rara vez se cumple en la práctica, el modelo sigue siendo eficaz para muchos problemas de clasificación de texto.

Proceso de Clasificación:

1. Extracción de Características:

- Representación de Texto: Se convierten los textos en vectores usando Bolsa de Palabras o TF-IDF.

2. Entrenamiento del Modelo:

- Se entrena el modelo Naive Bayes utilizando un conjunto de datos etiquetado. Calcula las probabilidades a priori de cada clase y las probabilidades condicionales de cada palabra dada una clase.

3. Predicción:

- Para clasificar un nuevo texto, el modelo calcula la probabilidad posterior para cada clase usando el teorema de Bayes y selecciona la clase con la probabilidad más alta.

Ejemplo de Aplicación:

- Saludos: Mensajes como "¡Hola! ¿Cómo puedo ayudarte?" podrían tener alta probabilidad de ser clasificados como positivos.

- *Quejas y Reclamos:* Textos como "Tu respuesta fue insatisfactoria" podrían tener alta probabilidad de ser clasificados como negativos.
- *Peticiones:* Mensajes como "Quisiera hacer una solicitud de servicio" podrían tener alta probabilidad de ser clasificados como neutrales

Implementación en Clasificación de PQRs

1. Preprocesamiento:

- Tokenización, Normalización, Eliminación de Stopwords, Lematización, y Stemming: Prepara los datos textuales eliminando el ruido y estandarizando el texto.

2. Representación del Texto:

- *Bolsa de Palabras o TF-IDF:* Transforma el texto en vectores numéricos.

3. Entrenamiento y Evaluación:

- Regresión Logística y Naive Bayes: Entrena y evalúa los modelos usando un conjunto de datos etiquetado que clasifique los textos en las categorías de PQRs.

4. Clasificación:

- Aplica el modelo entrenado a nuevos textos para clasificar saludos, quejas, reclamos, peticiones y despedidas según el sentimiento detectado.

APRENDIZAJES PRÁCTICOS

Estos ejercicios permitirán a los estudiantes aplicar los conceptos aprendidos en situaciones prácticas y específicas.

Práctica: Limpieza, Preprocesamiento y Representación de Datos para Análisis de Sentimientos

Objetivo:

En el campo del procesamiento de lenguaje natural (NLP), la capacidad para limpiar y preprocesar datos textuales es fundamental para construir modelos precisos y efectivos. Esta práctica tiene como objetivo familiarizarte con las técnicas esenciales para preparar datos de texto para un análisis de sentimientos, abordando tanto la limpieza y preprocesamiento como la representación del texto.

INSTRUCCIONES

Paso 1: Definir el Conjunto de Datos

El profesor definirá un conjunto de datos con frases de ejemplo clasificadas en cada categoría:

- ***Saludo:*** "Hola", "Buenos días", "Buenas tardes", "¿Cómo estás?"
- ***Queja:*** "Quisiera hacer una queja", "No estoy satisfecho con el servicio", "El producto llegó dañado"
- ***Reclamo:*** "Necesito reclamar por un error", "Quiero hacer un reclamo", "Esto es inaceptable, quiero una solución"
- ***Petición:*** "Me gustaría pedir información", "Necesito saber más detalles sobre el producto", "Por favor, envíenme más información"
- ***Despedida:*** "Adiós", "Hasta luego", "Nos vemos", "Chao"

Cada frase se asociará con una etiqueta que indique su categoría.

Paso 2: Crear un Diccionario de Palabras y Calcular Frecuencias

El profesor creará un diccionario que cuente cuántas veces aparece cada palabra en cada categoría:

- ***Contar palabras por categoría:*** Por cada frase, se dividirán las palabras y se contarán las apariciones de cada palabra en su respectiva categoría.
- ***Calcular la frecuencia total de palabras:*** Se sumarán todas las apariciones de palabras en cada categoría para obtener la frecuencia total de palabras por categoría.
- ***Calcular el número total de frases:*** Se contará el total de frases en el conjunto de datos.

Paso 3: Implementar el Clasificador Naive Bayes

Para clasificar una nueva frase, el profesor seguirá estos pasos:

- **Dividir la frase en palabras:** Se convertirán todas las palabras a minúsculas y se dividirán.
- **Calcular las probabilidades por categoría:**
    - Para cada categoría, se empezará con la probabilidad de la categoría (cuántas frases pertenecen a esa categoría dividido por el total de frases).
    - Para cada palabra en la nueva frase, se calculará la probabilidad de que esa palabra aparezca en la categoría utilizando el conteo de palabras y aplicando la suavización de Laplace (sumar 1 al contador para evitar divisiones por cero).
    - Se sumarán los logaritmos de estas probabilidades para obtener el puntaje total para cada categoría.
- Elegir la categoría con la mayor probabilidad: La categoría con el puntaje más alto será la predicción del clasificador para la nueva frase.

Paso 4: Probar el Clasificador

Para asegurarse de que el clasificador funcione correctamente, el profesor probará algunas frases nuevas, como:

- "Hola, ¿cómo estás?"
- "Quisiera hacer una queja sobre el servicio"
- "Esto es inaceptable"
- "Necesito información sobre el producto"
- "Hasta luego, nos vemos"

MATERIAL COMPLEMENTARIO

Campista, en este espacio encontraras ayudas en diferentes formatos que pueden potenciar tu proceso de aprendizaje.

- Andrew Ng Naive Bayes Text Clasification
  https://www.youtube.com/watch?v=NFd0ZQk5bR4

- Tutorial 49- How To Apply Naive Bayes' Classifier On Text Data (NLP)- Machine Learning
  https://www.youtube.com/watch?v=temQ8mHpe3k