Prueba tecnica

Jose Antonio Rodriguez Rivera

2023-06-23

Documentación: Modelo Básico de Preguntas y Respuestas

Este documento presenta la solución para el reto de construir un modelo básico de preguntas y respuestas desde cero. El código está escrito en Python y utiliza bibliotecas como pandas, scikit-learn y numpy.

Descripción del Problema

El objetivo de este ejercicio es desarrollar un modelo que pueda responder preguntas en función de un conjunto de datos de preguntas y respuestas. El modelo debe ser capaz de recibir una pregunta como entrada y proporcionar la respuesta correspondiente.

Datos de Entrada

El conjunto de datos utilizado en este ejercicio es un archivo CSV llamado "ejercicio2.csv". El archivo contiene dos columnas: "Pregunta" y "Respuesta". Cada fila representa una pregunta y su respuesta asociada. Los datos se dividen en conjuntos de entrenamiento y prueba para evaluar el rendimiento del modelo.

Procesamiento de Datos

Antes de entrenar el modelo, se realiza un proceso de preprocesamiento de datos que incluye las siguientes etapas:

1. Lectura de Datos: Se lee el archivo CSV utilizando la biblioteca pandas y se almacena en un DataFrame.
2. División de Datos: El conjunto de datos se divide en conjuntos de entrenamiento y prueba utilizando la función train_test_split de scikit-learn. Se utiliza una proporción de 80% para entrenamiento y 20% para prueba.
3. Vectorización de Preguntas: Se utiliza la técnica de TF-IDF (Term Frequency-Inverse Document Frequency) para convertir las preguntas en vectores numéricos. Se utiliza la clase TfidfVectorizer de scikit-learn para realizar la vectorización.

Entrenamiento del Modelo

El modelo utilizado en este ejercicio es una regresión logística. Después de vectorizar las preguntas, se entrena el modelo utilizando los datos de entrenamiento. Se utiliza la clase LogisticRegression de scikit-learn para entrenar el modelo.

Validación del Modelo

Una vez entrenado el modelo, se realiza la validación utilizando los datos de prueba. Se generan predicciones para las preguntas de prueba y se evalúa la precisión del modelo utilizando la métrica de exactitud (accuracy) proporcionada por la función accuracy_score de scikit-learn.

Pruebas del Modelo

Se proporciona una función adicional llamada "ask_question(question)" que permite realizar pruebas interactivas con el modelo entrenado. Esta función toma una pregunta como entrada, la vectoriza utilizando el vectorizador TF-IDF entrenado y predice la respuesta utilizando el modelo de regresión logística.

Mejoras Potenciales

Aunque este modelo básico cumple con los requisitos del ejercicio, existen varias áreas en las que se podría mejorar:

- Limpieza de Texto: Se podría implementar un proceso de limpieza de texto antes de la vectorización, eliminando caracteres especiales, convirtiendo el texto a minúsculas, etc.
- Evaluación Cruzada: En lugar de una única división de datos en conjuntos de entrenamiento y prueba, se podría implementar una validación cruzada para obtener una estimación más robusta del rendimiento del modelo.
- Aumento de Datos: Si el conjunto de datos es limitado, se podría aplicar técnicas de aumento de datos para generar más ejemplos y mejorar la capacidad de generalización del modelo.
- Explorar Otros Modelos: Además de la regresión logística, se podrían probar otros modelos como árboles de decisión, redes neuronales u otros enfoques avanzados de procesamiento de lenguaje natural.

Conclusiones

En este ejercicio, se ha construido un modelo básico de preguntas y respuestas desde cero utilizando Python y scikit-learn. El modelo utiliza regresión logística y TF-IDF para clasificar preguntas y generar respuestas. Se ha realizado el preprocesamiento de datos, entrenamiento, validación y pruebas del modelo. Se han discutido posibles mejoras y áreas de exploración adicional.