Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Integrantes:

José Jesús Ramírez Cruz 202052478

Gabriel Reyes Leal 202053516

Alfonso Saldaña Campos 202056307

Periodo: Primavera 2024

Materia: Minería de Datos

Manual de Usuario

Nombre del Docente: María Beatriz Bernabé Loranca

Manual de Usuario para el Análisis de Sentimientos

INTRODUCCION

Este manual de usuario está diseñado para guiarte en el uso de un script de Python que realiza análisis de sentimientos en tweets utilizando técnicas de procesamiento de lenguaje natural (NLP) y machine learning. A través de este manual, aprenderás cómo preparar los datos, ejecutar el análisis y visualizar los resultados.

REQUISITOS PREVIOS

Antes de comenzar, asegúrate de tener instalados los siguientes requisitos:

- Python 3.x
- Librerías de Python: nltk, openpyxl, scikit-learn, joblib, unidecode, matplotlib, numpy, csv, re

Puedes instalar las librerías necesarias utilizando pip:

```
pip install nltk openpyxl scikit-learn joblib unidecode matplotlib numpy
```

PASO 1: PREPARACIÓN DEL ENTORNO

Descargar Recursos de nltk

Para que el script funcione correctamente, necesitarás descargar ciertos recursos de nltk. Ejecuta el siguiente código en Python para descargarlos:

```
import ssl
import nltk

ssl._create_default_https_context = ssl._create_unverified_context
nltk.download('wordnet')
nltk.download('stopwords')
```

Archivos de Datos

Asegúrate de tener los archivos de Excel con los tweets que deseas analizar. El script está diseñado para leer los datos de archivos de Excel (.xlsx).

PASO 2: ESTRUCTURA DEL SCRIPT

El script se compone de varias funciones organizadas de manera que permiten:

1. Preprocesar el texto: Limpieza y normalización de los tweets.
2. Clasificar sentimientos: Etiquetar los tweets como positivos, negativos o neutrales.
3. Entrenar un modelo Naive Bayes: Para clasificar automáticamente los sentimientos.
4. Visualizar los resultados: Mostrar los porcentajes de sentimientos en gráficos de barras.

Variables y Diccionarios

- diccionario_principal: Contiene las listas de palabras positivas y negativas.
- stop_words: Lista de palabras comunes que se eliminarán durante el preprocesamiento.

Funciones Clave

1. preprocess_text(text): Limpia y tokeniza el texto.
2. get_sentiment_labels(text): Determina el sentimiento basado en el diccionario.
3. read_tweets_from_excel(file_path): Lee los tweets de un archivo de Excel.
4. process_tweets_and_get_sentiments(tweets): Procesa los tweets y obtiene las etiquetas de sentimientos.
5. calculate_sentiment_percentages(sentiment_labels1, sentiment_labels2): Calcula los porcentajes de sentimientos.
6. plot_comparison(...): Visualiza la comparación de sentimientos.
7. train_naive_bayes_model(X, y): Entrena y guarda el modelo Naive Bayes.
8. clean_text(text): Limpia y normaliza el texto eliminando caracteres no alfabéticos.
9. extract_words_from_excel(input_file): Extrae palabras de un archivo de Excel.
10. write_top_words_to_csv(words, output_file): Escribe las palabras más comunes a un archivo CSV.

Paso 3: Ejecución del Script

Leer y Preprocesar los Tweets

Primero, lee los tweets desde los archivos de Excel:

```
file_path1 = "path/to/Tiempo1ClaudiaAnalisis.xlsx"
file_path2 = "path/to/Tiempo1XochitlAnalisis.xlsx"

tweets1 = read_tweets_from_excel(file_path1)
tweets2 = read_tweets_from_excel(file_path2)
```

Obtener Etiquetas de Sentimiento

Procesa los tweets y obtén las etiquetas de sentimiento:

```
sentiment_labels1 = process_tweets_and_get_sentiments(tweets1)
sentiment_labels2 = process_tweets_and_get_sentiments(tweets2)
```

Calcular Porcentajes de Sentimientos

Calcula los porcentajes de sentimientos para ambos conjuntos de tweets:

```
 resultados = calculate_sentiment_percentages(sentiment_labels1, sentiment_labels2)
```

Visualizar Resultados

Visualiza la comparación de sentimientos:

```
(positive_percent1, negative_percent1, neutral_percent1,
 positive_percent2, negative_percent2, neutral_percent2,
 total_tweets1, total_tweets2,
 positive_tweets1, positive_tweets2,
 negative_tweets1, negative_tweets2,
 neutral_tweets1, neutral_tweets2) = resultados

plot_comparison(positive_percent1, negative_percent1, neutral_percent1,
                positive_percent2, negative_percent2, neutral_percent2,
                total_tweets1, total_tweets2,
                positive_tweets1, positive_tweets2,
                negative_tweets1, negative_tweets2,
                neutral_tweets1, neutral_tweets2)
```

Entrenar el Modelo Naive Bayes

Entrena el modelo Naive Bayes con los datos de los tweets:

```
X = tweets1 + tweets2
y = sentiment_labels1 + sentiment_labels2

train_naive_bayes_model(X, y)
```

Analizar Nuevos Tweets (Tiempo 2)

Lee los nuevos tweets y preprocesa:

```
file_path3 = "path/to/Tiempo2ClaudiaAnalisis.xlsx"
file_path4 = "path/to/Tiempo2XochitlAnalisis.xlsx"

tweets3 = read_tweets_from_excel(file_path3)
tweets4 = read_tweets_from_excel(file_path4)

preprocessed_tweets3 = [preprocess_text(str(tweet)) for tweet in tweets3]
preprocessed_tweets4 = [preprocess_text(str(tweet)) for tweet in tweets4]
```

Carga el modelo y el vectorizador, y realiza las predicciones:

```
model = joblib.load('naive_bayes_model.pkl')
vectorizer = joblib.load('count_vectorizer.pkl')

X_vectorized3 = vectorizer.transform(preprocessed_tweets3)
X_vectorized4 = vectorizer.transform(preprocessed_tweets4)

sentiment_labels3 = model.predict(X_vectorized3)
sentiment_labels4 = model.predict(X_vectorized4)
```

Calcula y visualiza los sentimientos para el nuevo conjunto de datos:

```
resultados_tiempo2 = calculate_sentiment_percentages(sentiment_labels3, sentiment_

(positive_percent3, negative_percent3, neutral_percent3,
 positive_percent4, negative_percent4, neutral_percent4,
 total_tweets3, total_tweets4,
 positive_tweets3, positive_tweets4,
 negative_tweets3, negative_tweets4,
 neutral_tweets3, neutral_tweets4) = resultados_tiempo2

plot_comparison(positive_percent3, negative_percent3, neutral_percent3,
                positive_percent4, negative_percent4, neutral_percent4,
                total_tweets3, total_tweets4,
                positive_tweets3, positive_tweets4,
                negative_tweets3, negative_tweets4,
                neutral_tweets3, neutral_tweets4)
```

PASO 4: ANÁLISIS DE PALABRAS COMUNES

Extrae las palabras más comunes y escribe a CSV:

```
input_xlsx = 'path/to/Tiempo1ClaudiaAnalisis.xlsx'
input_xlsx2 = 'path/to/Tiempo1XochitlAnalisis.xlsx'
output_csv2 = 'path/to/conteoPreClaudia.csv'
output_csv = 'path/to/conteoPreXochitl.csv'

words = extract_words_from_excel(input_xlsx)
words2 = extract_words_from_excel(input_xlsx2)

write_top_words_to_csv(words2, output_csv2)
write_top_words_to_csv(words, output_csv)
```

Repite para el segundo conjunto de datos:

```
input_xlsx = 'path/to/Tiempo2ClaudiaAnalisis.xlsx'
input_xlsx2 = 'path/to/Tiempo2XochitlAnalisis.xlsx'
output_csv2 = 'path/to/conteoPostClaudia.csv'
output_csv = 'path/to/conteoPostXochitl.csv'

words = extract_words_from_excel(input_xlsx)
words2 = extract_words_from_excel(input_xlsx2)

write_top_words_to_csv(words2, output_csv2)
write_top_words_to_csv(words, output_csv)
```