Práctica curso 24-25

Procesamiento del Lenguaje Natural

Grado en Ciencia de Datos e Inteligencia Artificial

MyRedditAssistant

El objetivo de esta práctica es crear un sistema de análisis y clasificación de posts de Reddit relacionados con Data Science. Se deberán implementar 6 módulos que se describen en este documento. Para ello, se proporciona un dataset en el fichero *dataframe_reddit_datascience.csv*.

Descarga del dataset: https://drive.upm.es/s/4usxJO42z2y2QnR

La práctica se realizará en parejas, que deberán estar dadas de alta en Moodle.

Módulos a implementar

1. **Normalización**. Implementa una función para limpiar y normalizar los textos de los *posts* (texto en la columna "*post")*. La versión normalizada se grabará en una nueva columna "*clean_post"*. Deberá realizarse una limpieza y normalización de los datos en función de las características que observes en los textos. Describe con detalle cada paso seguido durante el preprocesado y normalización y añade la función ***preprocess_post(text: str)*** al fichero *core.py* con la implementación final de este módulo.

2. **Sistema de clasificación de subreddit.** Se deberá implementar una función ***classify_subreddit(text)*** que clasifique un texto de entrada en una de las siguientes categorías: MachineLearning, datascience, statistics, learnmachinelearning, computerscience, AskStatistics, artificial, analytics, datasets, deeplearning, rstats, computervision, DataScienceJobs, MLQuestions, dataengineering, data, dataanalysis, datascienceproject, Kaggle.

   Para ello, se proporciona un dataset sobre el que se podrán entrenar distintos algoritmos de clasificación. La etiqueta del subreddit correspondiente se encuentra en la columna "subreddit". Se deberán probar, al menos, los siguientes 3 métodos:
   - Un método basado en TF-IDF + algoritmo de clasificación de machine learning
   - Un método basado en entidades reconocidas (Named-Entity Recognition) + algoritmo declasificación de machine learning
   - Un método basado en Word Embeddings + algoritmo de clasificación de machine learning

   Para evaluar cada método, se utilizará la métrica *f1 score*, y se utilizará un 70% de los datos del dataset para entrenamiento y un 30% para test (realizando un sampling aleatorio previo).
   En el notebook ***implementacion_modulo_2.ipynb*** deberás documentar todos los pasos seguidos y resultados obtenidos, así como explicar las diferencias entre los métodos probados. En el fichero ***core.py*** deberás incluir una función ***classify_subreddit(text)*** que devuelva un string con la etiqueta resultante de la clasificación.

3. **Extracción de información.** Deberás implementar varias funciones que, recibiendo como entrada un string (un post), devuelvan una lista con los resultados obtenidos.

- **find_subreddit_mentions(text: str)**: → Permitirá extraer los subreddits mencionados en un post. Por ejemplo: "*I\'m cross posting this from **/r/cyberlaw**, hopefully you guys find it as interesting*". Se debe extraer en este caso /r/cyberlaw. En caso de que haya más de uno, se deberán extraer todos y guardarlos en una lista. Para ello, se deberá utiliza una única expresión regular.
- **url_extraction(text: str)** → Permitirá extraer todas las URLs en un post mediante una única expresión regular
- **phone_number_extracion(text: str):** → Permitirá la extracción de números de teléfono mediante una única expresión regular
- **dates_extraction(text: str):** → Permitirá la extracción de todas las fechas contenidas en un post.
- **code_extraction(text:str)**: Extracción de código de programación o HTML incluido en un post. Permitirá la extracción de todo el código que se incluya en un post.

Todas las funciones estarán explicadas y detalladas en el notebook correspondiente, incluyendo su implementación final en 5 nuevas funciones en el archivo ***core.py***

4. **Análisis de sentimiento.** Implementa un módulo de detección de sentimiento de los posts. Para entrenar y evaluar diferentes métodos, dispones de la columna "sentiment" en el dataset. Al igual que en el módulo de clasificación de subreddits, Para evaluar cada método, se utilizará la métrica *f1 score*, y se utilizará un 70% de los datos del dataset para entrenamiento y un 30% para test (realizando un sampling aleatorio previo). Deberás probar y evaluar los siguientes métodos:
    - Un método basado en lexicons
    - Un método basado en palabras únicas en textos de cada tipo de sentimiento
    - Un método basado en Word embeddings + algoritmo de machine learning de clasificación

   La función de este módulo tendrá como nombre *sentiment_analysis(text: str)*

5. **Generación de resúmenes.** Implementa un método de resumen extractivo de posts basado en frecuencias y evalúa el resultado. Deberás realizar distintas pruebas para demostrar que el método es adecuado para el tipo de textos, realizando los ajustes necesarios para su correcto funcionamiento. La función se denominará *post_summarisation(text: str)*
6. **Distancias entre textos.** El último módulo permitirá, dados dos textos, calcular su distancia semántica. Para ello, evalúa diferentes alternativas y justifica la elección final tomada. La función se denominará *texts_distance(text1: str, text2: str)*

Entrega

Se debe entregar un fichero comprimido .zip con nombre el siguiente nombre:

APELLIDO1_APELLIDO2_GRUPO_XX.zip

donde APELLIDO1 y APELLIDO2 sean los primeros apellidos de los dos integrantes de la práctica, y XX el número asignado al grupo de prácticas en Moodle.

- ***implementacion_modulo_<número_módulo>.ipynb:*** Se deberá implementar un Notebook de Python que muestre todos los pasos realizados para resolver cada uno de los módulos. Se sustituirá <número_módulo> por el nº correspondiente. Por ejemplo, para el módulo de normalización se entregará un fichero *implementacion_modulo_1.ipynb*
- ***processed_dataset.csv.zip:*** Dataset ya preprocesado y que será el que se use en la implementación de los módulos 2-7. Se deberá entregar comprimido en un fichero .zip.
- ***core.py:*** Incluirá una función por cada uno de los módulos para incluir su funcionalidad. En aquellos casos en los que en los que se solicite que se prueben distintos métodos (por ejemplo, en el módulo 2), se deberá implementar la función con aquel que mejores resultados haya arrojado.
- ***testing.ipynb:*** Un notebook en el que se invocarán a las funciones implementadas en ***core.py*** y se comprobará el correcto funcionamiento del mismo con distintos ejemplos.

Importante

Todos los notebooks y código deberán estar correctamente documentos utilizando celdas de markdown, explicando todos los pasos realizados, los resultados obtenidos y toda decisión tomada. Una explicación escasa o no suficientemente detallada conllevará el suspenso en la práctica.