Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Inteligencia Artificial Avanzada:

Proyecto Final:

Clasificador de textos.

Saúl Sosa Díaz
(alu0101404141@ull.edu.es)

La Laguna, martes 2 de Abril de 2023

Índice:

Preprocesado.

Para preprocesar las noticias se han seguido los siguientes pasos.

- Eliminación de números y símbolos de puntuación.

- Eliminación de las StopWords inglesas: se han eliminado las StopWords de la lengua inglesa que se encuentran en la librería nltk.

- Eliminación de marcas HTML: se han borrado todas las marcas HTML utilizando la librería BeautifulSoup.

- Conversión a minúsculas: todas las letras de las noticias han sido convertidas a minúsculas para evitar problemas de sensibilidad a las mayúsculas.

- Corrección ortográfica: se ha utilizado la librería TextBlob para corregir cualquier error de ortografía en el texto.
  Para mejorar el tiempo de ejecución del programa, se ha implementado una estrategia que consiste en verificar si la palabra existe en un diccionario de la librería de nltk antes de llevar a cabo la corrección ortográfica. Si la palabra está en el diccionario, no se realiza ninguna corrección y se procede directamente a la lematización. De esta manera, se ha logrado reducir el tiempo que se dedica a la corrección ortográfica, que es la tarea que más tiempo consumía en el programa.

- Lematización: se ha utilizado la librería spacy para llevar a cabo la lematización, lo que significa que se han transformado las palabras de las noticias a su forma base, eliminando inflexiones gramaticales y simplificando el análisis semántico.

- Comprobar que la palabra resultante se encuentre en un diccionario.
  El diccionario seleccionado para esta tarea es el aportado por nltk.

Librerías utilizadas.

Para ejecutar este programa son necesarias las siguientes librerías:

- Pandas: Se utiliza para manipular y analizar datos.

```
Python
python3 pip install pandas
```

- NLTK: Se utiliza para el procesamiento del lenguaje natural.

```
Python
python3 pip install nltk
```

- BeautifulSoup 4: Para analizar y extraer datos de documentos HTML y XML.

```
Python
python3 pip install beautifulsoup4
```

- TextBlob: Se utiliza para realizar tareas de análisis de texto como etiquetado de partes del discurso, análisis de sentimientos, análisis de sujetos y extracción de frases clave.

```
Python
python3 pip install textblob
python3 -m textblob.download_corpora
```

- Spacy: Se utiliza para realizar tareas avanzadas de procesamiento de texto, como etiquetado de partes del discurso, análisis sintáctico, reconocimiento de entidades nombradas, análisis semántico y más.

```
Python
python3 pip install spacy
python3 -m spacy download en_core_web_sm
```

Estructura de directorios.

.
├── src ← Contiene el código fuente del programa.
├── data ← Contiene el fichero de entrenamiento, el vocabulario y el fichero de prueba.
├── corpus ← Contiene los corpus necesarios para la creación de los modelos.
├── solutions ← Contiene las soluciones al problema actual.
└── models ← Contiene los modelos generados por el programa.

Instrucciones de uso.

Para ejecutar el programa hay que situarse en la carpeta principal del proyecto y lanzar el siguiente comando:

```python
python3 ./src/main.py
```

El programa admite varios parámetros:

Parámetro	Utilidad
-v, --vocab	Crea el vocabulario de las noticias de entrenamiento.
-c, --corpus	Crea los corpus y los guarda en la carpeta corpus.
-m, --models	Crea los modelos.
-t, --text	Preprocesa el texto a clasificar

Para garantizar que el programa se ejecute correctamente, se ha implementado una funcionalidad que obliga a proporcionar los parámetros necesarios en caso de que falte algún archivo.

Implementación del programa.

El programa se divide en múltiples ficheros.

- aciertos.py: Utilizado para estimar los errores.

- categorizeText.py:
  Utilizado para categorizar las noticias, producirá en la carpeta *solutions* dos ficheros, *clasificacion_alu0101404141.csv* y *resumen_alu0101404141.csv*.
  En este fichero se compara las probabilidades de que una noticia pertenezca a una clase u a otra y le asigna la mayor.

- createCorpusTrain.py:
  En este fichero se crean los corpus.

- createModels.py:
  En este fichero se crean los modelos.

- createVocab.py:
  En este fichero se crea el vocabulario.

- main.py:
  Este es el fichero principal del programa y gestiona si es necesario crear algún fichero antes de categorizar la noticia.

- preprocessText.py:
  Preprocesa las noticias a categorizar y crea un nuevo fichero en la carpeta *data*. Se hace de esta manera para no tener que preprocesar las noticias todas las veces que se ejecuta la aplicación y evitar los extenuantes tiempos de ejecución.

- Los siguientes ficheros contienen las funciones que hacen que funcionen los ficheros anteriores.
  - utilsCorpus.py
  - utilsModels.py
  - utilsVocab.py
  - utilsText.py

Estimación de errores.

Se ha observado que, al utilizar los mismos datos para el entrenamiento y la validación del modelo, se obtiene un porcentaje de acierto del 89.8%.

Por otro lado, al dividir los datos en dos conjuntos, uno para el entrenamiento y otro para la prueba, se ha obtenido un porcentaje de acierto del 77.8%.