Ejercicio 3: Clasificación de Texto

Comparación de Modelos

1 Introducción

En las últimas sesiones hemos visto algunos modelos de para clasificar texto.

El propósito de este ejercicio es comparar el desempeño de los modelos, así cómo la fácilidad de uso de los mismos.

La tarea consiste en tomar un corpus de artículos de noticias públicados en el 2014 y entrenar un modelo para que pueda decidir a qué categoría pertenece el artículo.

2 El Corpus

Vamos a utlizar el corpus “News-Aggregation” publicado originalmente en el repositorio de la UCI[1].

Este repositorio contiene la siguiente información:

- **ID**: Identificador numérico del artículo
- **TITLE**: El título del artículo
- **URL**: URL dónde se publicó el artículo
- **PUBLISHER**: Quién publicó el artículo
- **CATEGORY**: la categoría a la que pertenece el artículo, las categorías existentes son:
  - *b* : Business
  - *f* : Science and Technology
  - *e* : Entertaiment
  - *m* : Health and Medicine
- **STORY**: Id alfanumérica de para referenciar al contenido del artículo
- **HOSTNAME**: Dominio donde fue publicado el artículo
- **TIMESTAMP**: marca el tiempo aproximado de la publicación

Para la tarea de clasificación solamente vamos a usar las columnas de TITLE y CATEGORY, sin embargo, con las otras columnas podemos hacer alge de EDA para entender mejor nuestro corpus/dataset

1 Gasparetti,Fabio. (2016). News Aggregator. UCI Machine Learning Repository. https://doi.org/10.24432/C5F61C.

3 Actividad 1: EDA sobre nuestro corpus

Para conocer mejor el corpus con el que estamos trabajando vamos a realizar algunas tareas de EDA. En nuestro caso, el EDA consistirá en construir gráficos de barras que muestren cómo se distribuyen los artículos en el dataset de acuerdo a:

- Mes en el que fueron publicados (el mes se extrae de la columna TIMESTAMP)
- Categorías
  - Muestra también el % de artículos del dataset que pertenecen a cada categoría.
- Publishers

Utiliza la librería "wordcloud"[2] para generar nubes de palabras por categoría.

4 Clasificación del texto

Vamos a comparar el desempeño de los siguientes modelos:

- Logistic Regression[3] [4]
  - Utilizar BoW[5] y TD-IDF[6]
- Naive Bayes[7]
  - Utilizar BoW y TD-IDF
- CNN
- LSTM

Para lleva a cabo la comparación hay que seguir estos pasos:

1. Preprocesamiento y normalización del texto. Eliminar las STOPWORDS, convertir a minúsculas, quitar símbolos no alfabéticos, etc.
2. Construir y entrenar el modelo
   1. Generar métricas de desempeño (accuracy al final del entrenamiento)
   2. Realizar predicciones de prueba con nuevos títulos (ver más abajo) y generar métricas del desepeño (acuracy con las nuevas predicciones)
3. Construir una tabla para mostrar el desempeño de los modelos

2 https://amueller.github.io/word_cloud/

3 https://scikit-learn.org/stable/user_guide.html

4 https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

5 https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

6 https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

7 https://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes

5 Entregables

Al final de este ejercicio tendrás uno o varios Notebooks con los scripts necesarios para lograr las actividades descritas.

Empaqueta los scripts/notebooks en un archivo "*ejercicio3_tuNombre.zip*" y súbelo en el espacio de Canvas designado para tal efecto.