Documento fundamentación de Evaluación Cognitiva.

Nota importante: Para ejecutar los siguientes scripts es necesario tener instalados Python 3 y R 4.0.

Códigos Python y R:

* Clasificador.py

Fundamentación.

Este código tiene la intención de clasificar un conjunto de palabras en un csv en verbos, adjetivos, sustantivos y adverbios y separarlos en diferentes dataframes donde se contarán el total de estos y así tener el resultado de frecuencia de cada clasificación.

Insertar código…

¿Cómo usar?

Para utilizar este código se necesita:

1. Tener un csv con palabras del idioma **español**, estas palabras debieron haber sido limpiadas y corregidas previamente para que pueda ejecutarse correctamente el clasificador.
2. Declarar en la variable **ruta\_origen** la ruta junto con el archivo de las palabras por analizar.
3. Ejecutar en la línea de comandos (estando en la carpeta donde se encuentre este script): py Clasificador.py
4. Esperar unos minutos hasta que finalice la ejecución del programa.

* conceptoSelector.py

Fundamentación.

Insertar código…

¿Cómo usar?

* conteo\_clasif\_palabras.py
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?
* conteo\_palabras.py

Fundamentación.

Este código analiza cada celda de un csv, busca si existe una falta de ortografía o gramática en la celda actual del recorrido y da diferentes sugerencias para corregir esa falta utilizando el wrapper de la librería Hunspell (CyHunspell). Una vez corregidase almacena en un diccionario y en caso

de que exista una falla parecida a esa la corregirá automáticamente y avanzará a la siguiente celda hasta que no haya más datos por procesar.

Insertar código…

¿Cómo usar?

1. Tener instaladas las librerías mostradas al principio del código. Nota: La librería Hunspell para Python se denomina CyHunspell.
2. Contar con un csv con palabras del idioma **español** que se encuentren con faltas de ortografía y gramática y celdas vacías llenadas con cualquiera de los siguientes datos: “NS”, “0”, “NN”, “WW”, “-”.
3. Declarar en la variable **ruta\_origen** la ruta donde se encuentra el documento junto con el nombre del archivo y en **ruta\_destino** la ruta y el nombre en donde se creará el archivo limpio.
4. Declarar en la variable **cols** las columnas que se quieren analizar.
5. Finalmente, durante la ejecución del programa seleccionar con el teclado del equipo la corrección correcta dependiendo del contexto y las palabras sugeridas.

* csvCleaner.py
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?
* formatter.py
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?
* quickPlotter.py
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?
* similitud.py
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?
* trainer.py
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?
* multi\_testing.R
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?
* testing 2.0v.R
  + Explicación.
  + Fundamentación.
  + Código.
  + ¿Cómo usar?

Jupyter notebooks: