Apéndices

Apéndice A: Matrices de confusión y resultados de precisión, exhaustividad y medida F

En este apéndice se presentan seis tablas con otros resultados obtenidos.

En cada una de las primeras cinco (tablas 1, 2, 3, 4 y 5), se encuentra la matriz de confusión, con sus valores promediados de verdadero positivo (tp), falso negativo (fn), falso positivo (fp), verdadero negativo (tn) y los números de casos, para los experimentos realizados. La sexta y última tabla (tabla 6) presenta los resultados promediados de precisión, exhaustividad y medida F.

	Clasificado como positivo	Clasificado como negativo	Total
Realmente positivo	435.8	164.1	599.9
Realmente negativo	164.7	435.3	600
Total	600.5	599.4	1200

Tabla 1: Matriz de confusión con los valores promedio (promedio de la validación cruzada con 10 pliegues) usando enunciados con adjetivos o adverbios. El total, N, se redondeó.

	Clasificado como positivo	Clasificado como negativo	Total
Realmente positivo	395.5	204.9	600.4
Realmente negativo	195.5	404.1	599.6
Total	591	609	1200

Tabla 2: Matriz de confusión con los valores promedio (promedio de la validación cruzada con 10 pliegues) usando todos los enunciados. El total, N, se redondeó.

	Clasificado como positivo	Clasificado como negativo	Total
Realmente positivo	415.1	185.8	600.9
Realmente negativo	183.8	419.3	603.1
Total	598.9	605.1	1200

Tabla 3: Matriz de confusión con los valores promedio (promedio de la validación cruzada con 10 pliegues) usando los enunciados agrupados con FNM. El total, N, se redondeó.

	Clasificado como positivo	Clasificado como negativo	Total
Realmente positivo	398.7	201.3	600
Realmente negativo	188	412	600
Total	586.7	613.3	1200

Tabla 4: Matriz de confusión con los valores promedio (promedio de la validación cruzada con 10 pliegues) usando enunciados con adjetivos o adverbios o disparadores de presuposición. El total, N, se redondeó.

	Clasificado como positivo	Clasificado como negativo	Total
Realmente positivo	67.2	37	104.2
Realmente negativo	38.3	65.7	104
Total	105.5	102.7	208

Tabla 5: Matriz de confusión con los valores promedio (promedio de la validación cruzada con 10 pliegues) usando enunciados con disparadores de presuposición. El total, N, se redondeó.

	Método	Precisión	Exhaustividad	Medida F
(1)	enunciados con adjetivos o adverbios	72.57	72.65	72.61
(2)	enunciados agrupados con FNM	69.31	69.08	69.19
(3)	enunciados con disparadores o adjetivos o adverbios	67.96	66.45	67.19
(4)	todos los enunciados	66.92	65.87	66.39
(5)	enunciados con disparadores de presuposición	63.70	64.50	64.10

Tabla 6: Resultados promediados de precisión, exhaustividad y medida F de la validación con 10 pliegues del clasificador creado.

Apéndice B: Descripción de los módulos del sistema

En este apéndice se describen brevemente los módulos de Python que componen al sistema. También se presenta el diagrama con las relaciones existentes entre estos módulos. En total son 14 módulos:

1. `analizador_info_imdb.py`: módulo principal. Encargado de comenzar el proceso, desde la recopilación de comentarios y datos hasta la validación del clasificador,

2. `infoIMDB.py`: módulo que extrae los nombres de las películas desde Wikipedia. También adquiere los datos generales desde IMDb,

3. `ObtenComentarios.py`: módulo que contiene al Web crawler, consigue los comentarios de las películas encontradas en el módulo anterior,

4. `matriz_documento_termino.py`: módulo que genera la matriz de datos, con los enunciados segmentados de las comentarios, para la aplicación del algoritmo de FNM,

5. `clusternmf.py`: módulo que recibe la matriz de datos y se encarga, mediante los dos módulos siguientes, de aplicar FNM e interpretar los resultados para generar la agrupación,

6. `gpnmf.py`: módulo que aplica el algoritmo de FNM a la matriz de datos recibida,

7. `datos_entrenamiento.py`: módulo que interpreta las dos matrices resultantes de la aplicación de FNM y entrega los enunciados agrupados,

8. `enunciados_adjetivos_adverbios.py`: módulo que entrega una lista con las oraciones que contienen adjetivos o adverbios,

9. `oraciones_con_pres_triggers.py`: módulo que entrega una lista con las oraciones que contienen disparadores de presuposición,

10. `oraciones_con_triggers_y_adjetivos.py`: módulo que entrega una lista con las oraciones que contienen adjetivos o adverbios o disparadores de presuposición,

11. `oraciones_normales.py`: módulo que entrega una lista con todas las oraciones,

12. `validacion.py`: módulo que aplica la validación cruzada con k pliegues. Entrena y prueba el clasificador, mediante el siguiente módulo, y calcula las medidas de desempeño,

13. `bayes_ingenuo.py`: módulo que entrena el clasificador bayesiano ingenuo y lo prueba con los pliegues adecuados correspondientes,

14. `prueba_bopang.py`: módulo que lleva a cabo las pruebas con los datos encontrados en [47].

En la figura 1 se presentan las llamadas que hacen los módulos entre ellos.

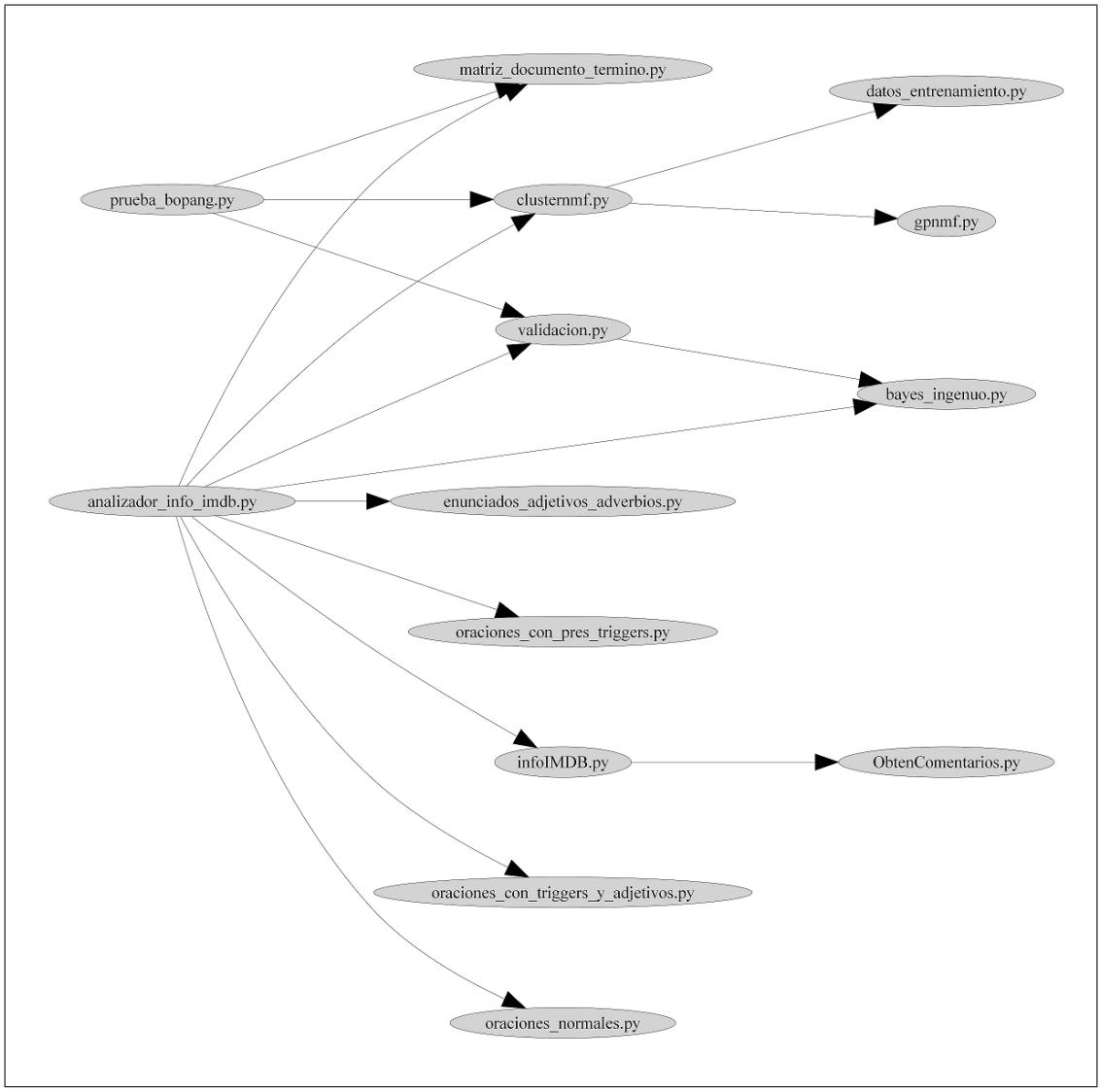

Figura 1: Relación de llamadas entre los módulos - La figura muestra las llamadas hechas entre cada uno de los módulos creados. Por ejemplo, el módulo `analizador_info_imdb.py` llama al módulo `infoIMDB.py`, y este a su vez llama al módulo `ObtenComentarios.py`

Apéndice C: Descripción del módulo de agrupamiento automático con FNM

El módulo `clusterFNM.py` contiene la función para agrupar automáticamente (por similitud) textos de acuerdo a una de dos tipos de métricas, por distancia Euclidiana o por el valor del coseno entre los documentos.

Parámetros de entrada:

- `archivo_documentos:` cadena con la ubicación (*path*) de un archivo de texto plano con un documento por línea.
- `r:` entero que indica el número de grupos deseados.
- `tol:` doble que indica la máxima diferencia entre la norma de la matriz de datos y la norma del producto de las dos matrices resultantes de la factorización.
- `timelimit:` entero que indica el límite de tiempo de ejecución, en segundos.
- `maxiter:` entero que indica el número máximo de iteraciones.

Regresa:

Una lista con `r` objetos tipo `Grupo`. Cada `Grupo` contiene los documentos que pertenecen a cada uno de los grupos: también incluye su nivel de pertenencia al grupo.

Guarda en disco:

- `clusterFNM.txt:` archivo de texto plano con los grupos encontrados y los documentos que pertenecen a cada uno de ellos.
- `matriz_datos.txt:` archivo de texto plano con la matriz de datos generada para el algoritmo de agrupamiento.

Apéndice D: Descripción del módulo de clasificación binaria mediante Bayes ingenuo

El módulo `clasificacionBayes.py` contiene la función para predecir dos tipos diferentes de clases de documentos mediante el algoritmo de Bayes ingenuo.

Parámetros de entrada:

- `archivo_documentos_clase1`: cadena con la ubicación (*path*) de un archivo de texto plano con un documento por línea. La primera línea del archivo deberá contener el nombre de la clase. Estos documentos pertenecen a la primera clase.
- `archivo_documentos_clase2`: cadena con la ubicación de un archivo de texto plano con un documento por línea. La primera línea del archivo deberá contener el nombre de la clase. Estos documentos pertenecen a la segunda clase.
- `archivo_documentos_a_clasificar`: cadena con la ubicación de un archivo de texto plano con un documento por línea. Estos documentos son los que se van a clasificar.
- `frecuencia_absoluta_minima`: entero que indica la frecuencia absoluta mínima con la que deben aparecer los tokens. Si no se indica, se usan todos los tokens.

Regresa:

Dos listas, la primera con los documentos de `archivo_documentos_a_clasificar` que pertenecen a la primera clase, la segunda con los documentos que pertenecen a la segunda clase.

Guarda en disco:

- `documentos_clase_`*nombre de la primera clase*`.txt`: archivo de texto plano con los documentos a los que se les asignó la primera clase.
- `documentos_clase_`*nombre de la segunda clase*`.txt`: archivo de texto plano con los documentos a los que se les asignó la segunda clase.