Instituto Politécnico Nacional

Escuela Superior de Cómputo

AVANCES DEL TRABAJO TERMINAL RECOLECTOR Y CLASIFICADOR DE NOTICIAS

2018-B013

ALUMNOS:
CARLOS ANDRES HERNANDEZ GOMEZ
LUIS DANIEL MEZA MARTÍNEZ

DIRECTORES:

M. en C. JOEL OMAR JUÁREZ GAMBINO
Dra. CONSUELO VARINIA GARCÍA MENDOZA

Ciudad de México, 11 de octubre de 2019

Índice

1. Introdcucción

Este documento explica los avances realizados de la semana octava a la décima correspondiente al periodo 22 de septiembre − 12 de octubre(ver Figura 1) del año 2019. las actividades reportadas son:

1. Incremento del corpus
2. Selección de noticias
3. Entrenamiento de algoritmos
4. Validación cruzada
5. Aplicación web

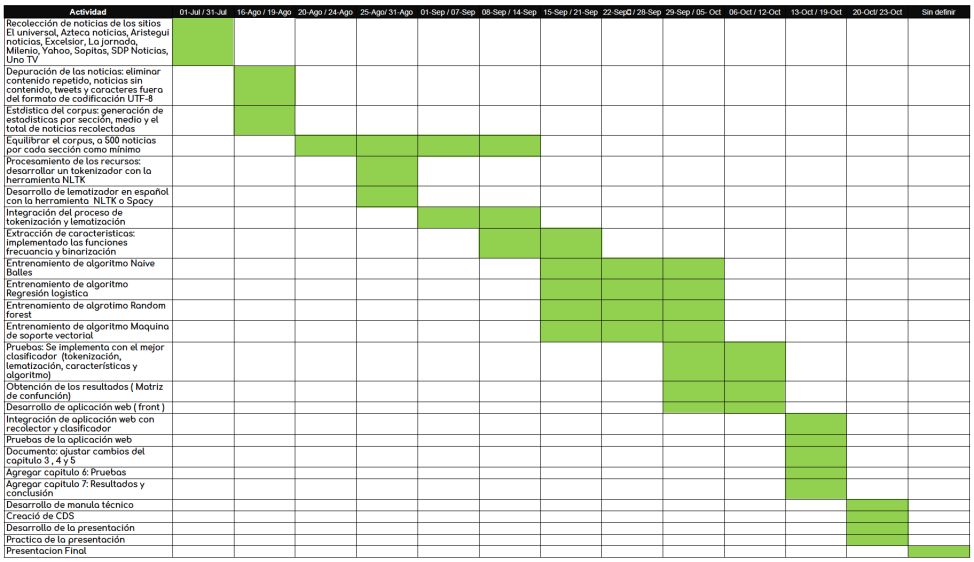

Figura 1: Cronograma correspondiente al trabajo terminal 2

2. Incremento del corpus

Como meta el número de noticias para formar el corpus se propuso de 500 noticias como mínimo por cada sección (las cuales son **ciencia y tecnología**, **deportes**, **política**, **cultura** y **economía**) sin embargo con vías de

alcanzar mejores resultados en el entrenamiento de los algoritmos, se incrementó el número de artículos a 700, es decir 3500 noticias en total.

Para este procedimiento se recolectaron noticias cada 3 días de los sitios web:

- **El Universal**: https://www.eluniversal.com.mx/
- **La Jornada**: https://www.jornada.com.mx/
- **Aristegui Noticias**: https://aristeguinoticias.com/
- **Sopitas**: https://www.sopitas.com/
- **TV Azteca**: https://www.aztecanoticias.com.mx/
- **El economista**: https://www.eleconomista.com.mx/

Se implementaron los *crawlers* desarrollos en el trabajo terminal 1. Además se especializaron algunos algoritmos para obtener mas noticias de la sección **cultura** específicamente del diario **El Economista**. El resultado de las noticias recolectadas se muestra en la Figura 2:

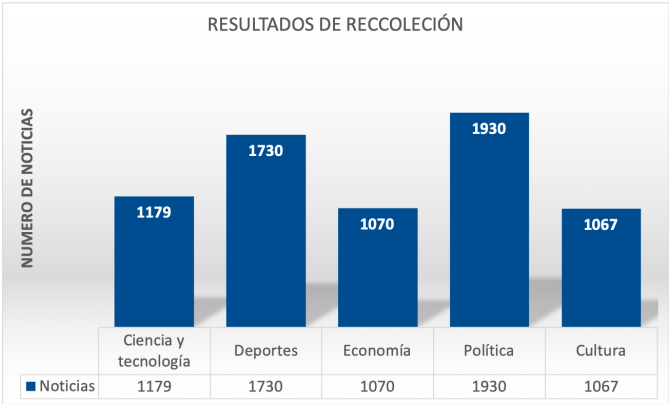

Figura 2: Gráfica de recolección de noticias

3. Selección de noticias

Una vez obtenidas las noticias se limpiaron y se seleccionaron los artículos aptos para ser usadas en el proceso de entrenamiento, mediante la aplicación de un filtro. Un texto válido debe contener como mínimo 180 palabras. La meta consistió en obtener 700 noticias validas por sección, sin embargo hay secciones que obtuvieron mas de dicho número y otras un poco menos. Cabe destacar que las secciones se acotaron a 700 noticias, los resultados obtenidos se muestra en la figura 3.

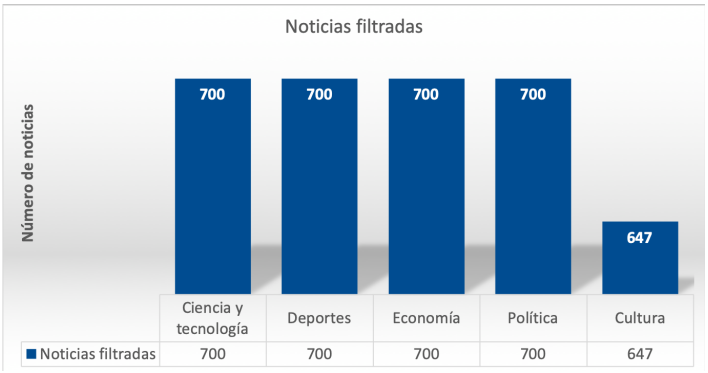

Figura 3: Gráfica de noticias obtenidas con mas de 180 palabras

El total de noticias obtenidas es de 3447.

4. Entrenamiento de algoritmos

Se entrenaron los algoritmos propuestos en trabajo terminal 1 los cuales son: **Naive bayes**, **Random forest**, **Maquina de soporte vectorial** y **Regresión logística**.

El proceso de entrenamiento se muestra en la Figura 4.

1. **Lectura del corpus**:Consiste en leer las noticias almacenadas en un archivo tipo CSV, con la librería **Pandas** de **python**. Se obtiene el título y la redacción del artículo.

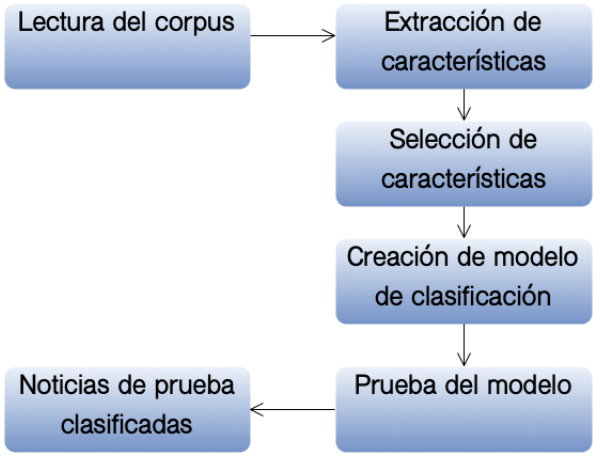

Figura 4: Proceso de entrenamiento

2. **Selección de características**: Se identifican las palabras que son comunes para cada sección por ejemplo, fútbol, jugador y gol son palabras que aparecen en el tópico de deportes.

3. **Extracción de características**: Se crea un espacio vectorial por cada noticia donde cada elemento del vector representa la presencia o ausencia de una característica(palabra). Cabe mencionar que las características son extraídas de 2 formas, binario(donde 1 representa la presencia de la característica y 0 la ausencia) y por frecuencia (donde se cuenta la cantidad de veces que cada característica aparece). Se ha implementado **CountVectorizer** de la librería *scikit learn* para este proceso.

4. **Creación del modelo**: Los modelos creados son entrenados de forma supervisada, se brinda un conjunto de noticias (es decir el espacio vectorial de cada artículo) y el resultado de la clasificación para crear el modelo. Se ha implementado los algoritmos de la librería **sckit learn**.

5. **Prueba del modelo**: El corpus se divide en 2 partes, donde la primera es llamada **conjunto de entrenamiento** (el cual consiste en el 93 % las noticias) y la segunda es llamada **conjunto de prueba** (el cual consiste en 7 % del corpus).

6. **Noticias de prueba clasificadas**: Se obtiene un vector numérico de

resultados el cual muestra la clasificación de las noticias de prueba. donde **0**:Deportes, **1**:Economía, **2**:Política, **3**:Cultura y **4**:Ciencia y tecnología.

5. Validación cruzada

Para medir la eficiencia de cada algoritmo se genera un conjunto de noticias de entrenamiento y otro de prueba sin embargo, la selección de estas noticias puede ser manipulados para obtener mejores resultados. Para evitar esta situación se generan las pruebas con un método llamado validación cruzada el cual consiste en partir el corpus en pliegues y generar las pruebas mas de una vez (ver Figura 5) y calcular la eficiencia promedio del clasificador, de esta forma se obtienen resultados mas robustos y confiables.

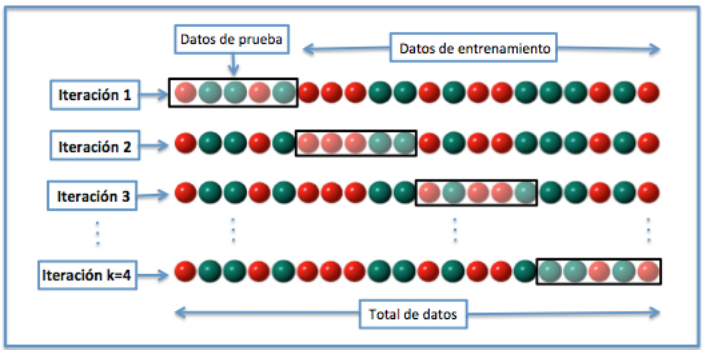

Figura 5: Validación cruzada

5.1. Métricas de evaluación

Utilizando la validación cruzada se debe medir la eficiencia del clasificador, para esto se hace uso de la matriz de confusión.

Matriz de confusión

Una matriz de confusión es una representación de la información de los resultados obtenidos por un clasificador, dicha matriz suele ser de tamaño n x

n, donde n es es el número de clases diferentes con las que se están trabajando.

		Valor de predicción	
		Positivos	**Negativos**
Valor real	**Positivos**	Verdadero Positivo (VP)	Falso Negativo (FN)
	Negativos	Falso Positivos (FP)	Verdadero Negativo (VN)

Figura 6: Matriz de confusión

La Figura 6 muestra un ejemplo de matriz de confusión con dos clases, la cual ejemplifica de manera adecuada las diferentes entradas de la misma.

Gracias a la matriz de confusión, es posible obtener ciertas métricas que nos ayudan a evaluar el modelo de aprendizaje, las cuales son:

Exactitud: es la proporción del número total de predicciones que son correctas respecto al total. Se determina utilizando la ecuación:

$$Exactitud = \frac{VP + VN}{VP + VN + FN + FP} \tag{1}$$

Recall: Es la proporción de predicciones positivas que fueron correctamente clasificadas. Se determina utilizando la ecuación:

$$Recall = \frac{VP}{VP + FP} \tag{2}$$

Precisión: Es la proporción de predicciones positivas que se clasificaron correctamente. Se determina con la siguiente ecuación:

$$Precision = \frac{VP}{VP + FN} \tag{3}$$

F-Measure (F1): Se interpreta como la media armónica entre Precisión y Recall. Se determina con la siguiente ecuación:

$$F - Measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4}$$

5.2. Resultados

Para las pruebas se han generado 3 pliegues en la validación cruzada, la Figura 7 muestra los resultados obtenidos al extraer las características por frecuencia y la Figura 8 muestra los resultado obtenidos al extraer las características de forma binaria.

Algoritmo	Noticias clasificadas correctamente	Noticias totales	Accuracy Promedio	Fmeasure Promedio	Recall Promedio	Precision Promedio
Naive Bayes	2877	3447	83%	84%	84%	84%
Maquina de soporte vectorial	2684	3447	78%	78%	79%	78%
Random Forest	2878	3447	83%	83%	84%	84%
Regresión logística	2983	3447	86%	86%	86%	86%

Figura 7: Resultados de entrenamiento modo frecuencia

Algoritmo	Noticias clasificadas correctamente	Noticias totales	Accuracy Promedio	Fmeasure Promedio	Recall Promedio	Precision Promedio
Naive Bayes	2844	3447	83%	83%	84%	83%
Maquina de soporte vectorial	2947	3447	85%	86%	86%	86%
Random forest	2898	3447	84%	84%	84%	84%
Regresión logística	3001	3447	87%	87%	87%	87%

Figura 8: Resultados de entrenamiento en modo binario

Se puede observar que el mejor resultado es dado por el clasificador **Regresión logística**, al extraer las características de forma binaria con un *Accuracy, Fmeasure, Recall y Precision* de un 87 %. Cabe mencionar que esta no son las pruebas finales de los clasificadores el siguiente paso es variar los parámetros usados en cada clasificador para obtener el mejor clasificador con los mejores parámetros.

6. Aplicación Web

El objetivo final del trabajo terminal es brindar una plataforma donde los usuarios puedan consultar las noticias clasificadas por cada sección, para ello se desarrollará una aplicación web. Esta etapa se divide en 2 vertientes principales: **Frontend** y **Backend**.

6.1. Frontend

Las interfaces de usuario se están desarrollando en el *Framework* **Java Server Faces**. La Figura 9 muestra la vista preliminar que tendrá el usuario al ingresar a la aplicación web.

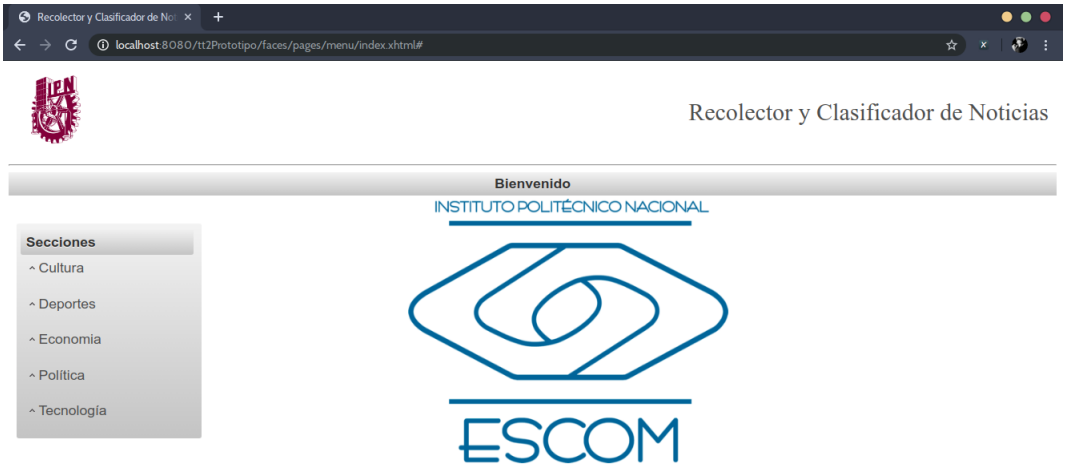

Figura 9: Pantalla de Inicio.

Una vez ingresando al sistema el usuario tendrá la posibilidad de seleccionar la sección deseada, las cuales son: **Ciencia y tecnología**, **Cultura**, **Deportes**, **Economía** y **Política**.

Después de dar clic en una sección se lleva acabado el proceso recolección de los artículos en los sitios definidos y la clasificación. El sistema muestra el mensaje **Proceso de recolección y clasificación** como se visualiza en la Figura 10.

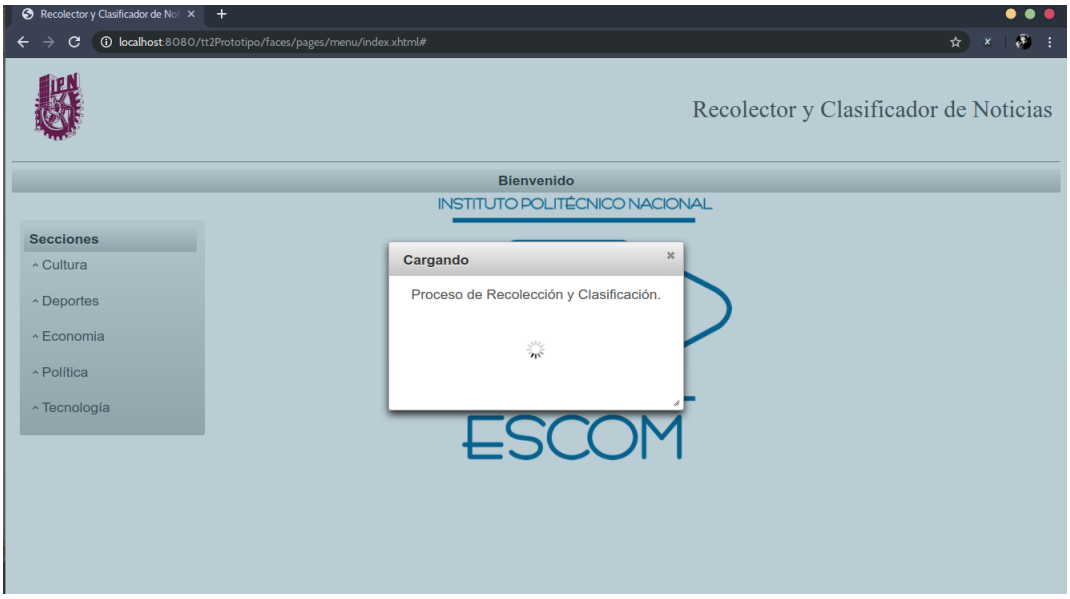

Figura 10: Mensaje de aplicación

Al concluir el proceso se muestra un mensaje para notificar al usuario que han sido clasificadas las noticias como se visualiza en la Figura 11 y el usuario tiene la opción de continuar con el proceso o cancelarlo.

Si el usuario ha decidido continuar con el proceso de recolección y clasificación, la aplicación muestra las noticias de la sección seleccionada. Además se genera una menú en la parte superior de la pantalla en el cual el usuario puede filtrar las noticias en un periodo de tres días, como se visualiza en la Figura 12.

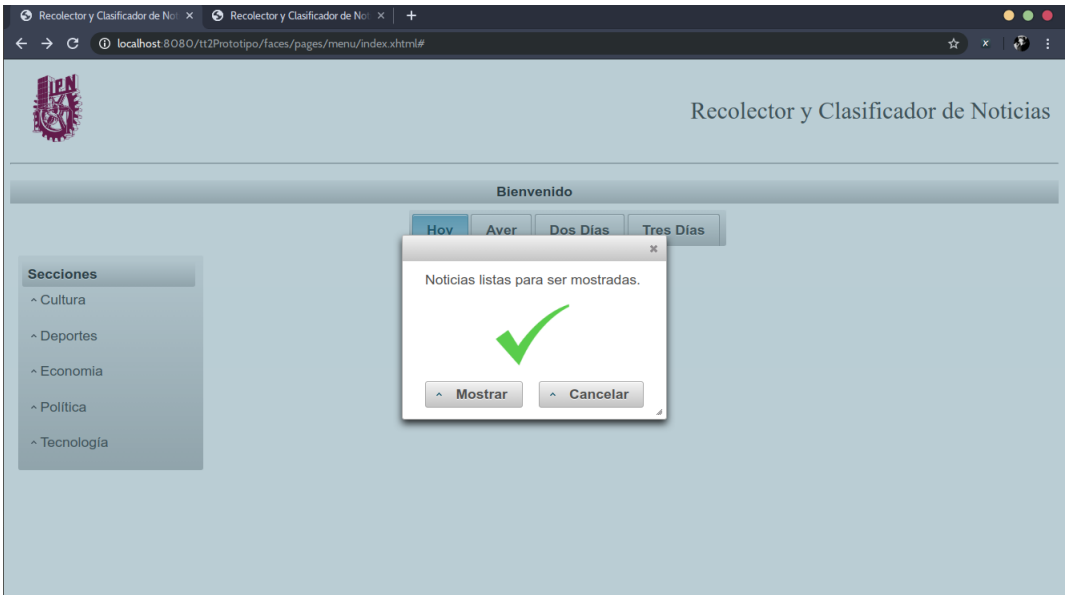

Figura 11: Mensaje de aplicación

6.2. Backend

Para el desarrollo de la aplicación web se ocupa un repositorio de *Maven* el cual permite controlar las versiones de las dependencias así como de las librerías utilizadas.

La implementación del **Backend** se está programando en **Python** y **Javs server faces**.

- **Python**: En este lenguaje de programación se está desarrollando el clasificador con la librería *Scikit learn* y los *Crawlers* con la librería *Scrapy*.
- **Java server faces**: La interacción del usuario con la aplicación se va desarrollar en este *framework*, y es en este punto donde se va programar la unión del proceso de clasificación y recolección con la aplicación web. Cabe destacar que esto aun no se ha iniciado debido a que los clasificadores siguen en pruebas.

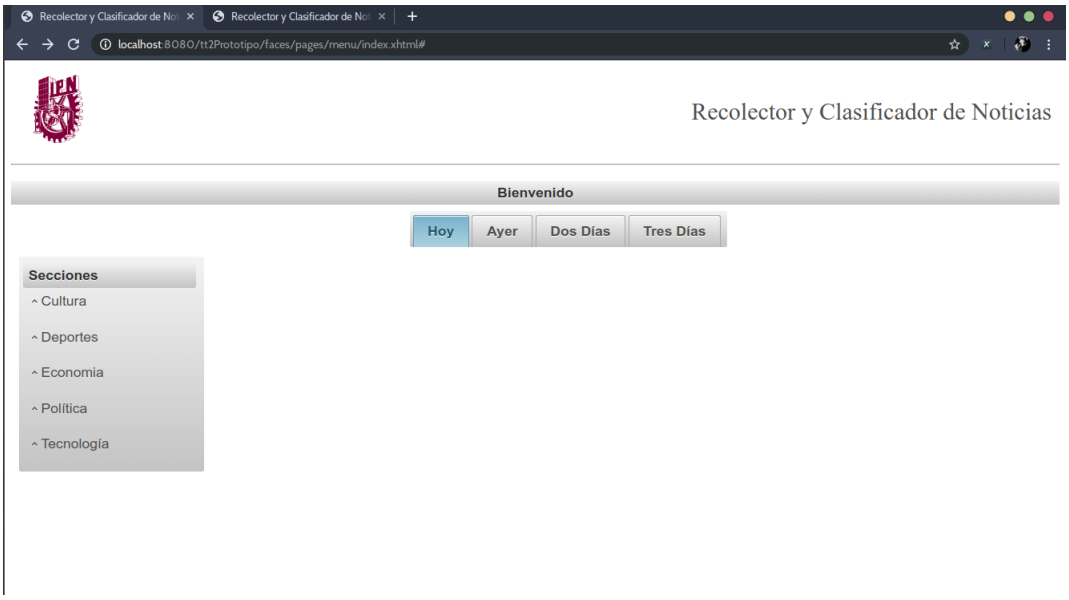

Figura 12: Noticias clasificadas

7. Conclusión

Como resumen de los avances, las tareas concluidas se muestran a continuación:

- Incremento el corpus de 500 a 700 noticias por sección
- Selección de noticias aptas para ser usadas en el entrenamiento
- Diseño de los clasificadores
- Resultados de validación cruzada

Actividades en desarrollo:

- Maqueta de la aplicación web(**frontend**)
- Unión del recolector y clasificador con la aplicación web