# T3: Minería de Texto Básica

CIMAT, Procesamiento de Lenguaje Natural, Ciencias de la Computación

Profesor: Dr. Adrián Pastor López Monroy

Entregar: Viernes 23 de Febrero de 2024 antes de las 23:59:59

## 1 Instrucciones

Realiza los siguientes puntos en un notebook de Python *lo mejor organizado y claro posible*. Ponga su nombre completo al archivo de entrega (e.g., adrian_pastor_lopez_monroy.ipynb) y también en la primera celda del notebook junto con el número de Tarea. Al entregar la tarea, sube al classroom el notebook como un archivo (de ser posible evita el zip). El notebook deberá haber sido ejecutado en tú máquina y mostrar el resultado en las celdas. Puede usar libremente el código de la clase *"BoW"* y *"DOR"* para completar esta actividad. Puede usar su propio código también, pero no debe usar librerías para hacer BoW y derivados (e.g., sklearn, gensim, etc.).

- El numero de palabras para los puntos 1-5 puede fijarse hasta 5000 con algún corte (e.g., por frecuencia o por mayor tfidf). El número de palabras consideradas para el punto 6 debe ser las 10,000 más frecuentes de toda la colección.
- Sí una palabra del dataset no está en los recursos léxicos, diseñe algo básico para lidiar con ello, o podría simplemente ignorarla en esa representación.
- Para los últimos dos puntos puede ayudarse estudiando la sección 8.5 del libro de *Modern Information Retrieval. Ricardo Baeza Yates. Addison Wesley*. Se adjuntan copias.

### 1.1 Se vale pedir ayuda y "copiar"

Se vale pedir ayuda y/o copiar con atribución entre los miembros de la clase y apegándose estrictamente a los siguientes puntos:

1. Del total de actividades que se solicitan hacer (6 para esta tarea) solo puedes pedir ayuda/copiar en un total de **dos**.
2. Para los puntos dónde se pide ayuda, brevemente escribe en qué pediste ayuda y a quién.
3. Si tuviste que reusar alguna parte de código que no es tuyo, deja claro dos cosas: 1) brevemente porque tuviste dificultad para hacerlo, 2) cómo lo solucionó tu compañero.

### 1.2 Selección de Términos y DTRs

Importante: Para cada punto debe elaborar una pequeña discusión de al menos un pequeño parráfo.

1. Programa y visualiza TCOR. Puede hacer esto de forma similar a como el profesor lo hizo en la *Práctica 4* con DOR. El pesado puede ser el que el profesor sugirió en clase TCOR o PPMI como lo sugiere Dan Jurafsky; hacer al menos dos gráficas, la de constelación de palabras y subconjunto para ver algunas palabras con flechas.
2. Programa y visualiza alguna implementación de Random Indexing. Puedes hacer esto reusando parte del código del profesor en la *Práctica 4* con DOR. Hacer al menos dos gráficas, la de constelación de palabras y subconjunto para ver algunas palabras con flechas.
3. Use alguna de las DTRs anteriores por separado de alguna forma para clasificación de documentos (e.g., promedio de vectores de términos en cada documento para representar). Compárelas contra un BoW-TFIDF de 5000 palabras más frecuentes.
4. Bajo la representación TCOR de los términos, y asumiendo un vocabulario de 5000 palabras, muestre por orden de mayor similitud coseno (ver chp 6 del libro de Dan) los 10 pares de palabras más parecidas en toda la colección.
5. Bajo la representación BoW-TFIDF de los documentos, y asumiendo un vocabulario de 5000 palabras, muestre en por orden de más similitud coseno (ver chp 6 del libro de Dan) los 10 pares de documentos más parecidos en toda la colección. Muestre el texto que contienen y muestre la categoría de cada uno.
6. Implemente Ganancia de Información o Chi2 como lo sugiere Baeza-Yates (no función de sklearn ni similar) para descubrir el top 50 de las palabras más relevantes de TODA la colección. Haga una gráfica también con la herramienta de *word_cloud* dónde el tamaño de la palabra corresponda a su ganancia de información:
    - https://amueller.github.io/word_cloud/