UNIVERSIDAD DEL VALLE DE GUATEMALA
Data Science
Sección 30

LABORATORIO 5

José Emilio Reyes Paniagua, 22674
Michelle Angel de María Mejía Villela, 22596
Silvia Alejandra Illescas Fernández, 22376

Guatemala, 26 de agosto del 2025

Repositorio: https://github.com/JEmilioRey1021/Laboratorio-5.-Clasificaci-n-de-Tweets-usando-Miner-a-de-Texto.git

Resumen

Este informe presenta el análisis de un conjunto de datos de tweets relacionado con desastres naturales utilizando técnicas de minería de texto. El objetivo principal es desarrollar un modelo de clasificación que pueda predecir si un tweet está relacionado con un desastre (target = 1) o no (target = 0). Para ello, se realizaron diversas etapas de preprocesamiento de los datos, análisis exploratorio, generación de n-gramas y entrenamiento de modelos de clasificación.

1. Introducción

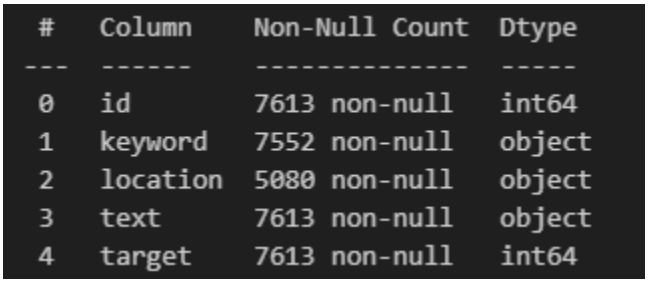
En este laboratorio se utilizó el conjunto de datos Natural Language Processing with Disaster Tweets proporcionado por Kaggle. El conjunto de datos contiene más de 7,613 tweets con etiquetas que indican si el tweet está relacionado con un desastre o no. El objetivo de este trabajo fue construir un modelo que pueda clasificar correctamente estos tweets.

2. Descripción del Conjunto de Datos

El conjunto de datos contiene las siguientes columnas:

1. id: Identificador único del tweet.
2. keyword: Palabra clave asociada al tweet (puede estar vacía).
3. location: Ubicación desde la cual se publicó el tweet.
4. text: Texto completo del tweet.
5. target: Etiqueta binaria que indica si el tweet es sobre un desastre (1) o no (0).

El conjunto de datos está dividido en dos archivos: train.csv para el entrenamiento del modelo y test.csv para probar su desempeño.



#	Column	Non-Null Count	Dtype
0	id	7613 non-null	int64
1	keyword	7552 non-null	object
2	location	5080 non-null	object
3	text	7613 non-null	object
4	target	7613 non-null	int64

Figura 1. Columnas del dataset

3. Preprocesamiento de los Datos

3.1. Limpieza de Datos

Antes de aplicar cualquier modelo, se realizó un exhaustivo proceso de limpieza de los datos. Las tareas de preprocesamiento incluyeron:

1. Conversión a minúsculas: Para evitar que las palabras en mayúsculas y minúsculas se trataran como diferentes.
2. Eliminación de caracteres especiales: Se eliminaron símbolos como `#`, `@`, y apóstrofes.
3. Eliminación de URLs: Se eliminaron las URLs presentes en los tweets.
4. Emoticones y emojis: Se eliminaron emoticones y emojis, ya que no contribuyen a la clasificación.
5. Eliminación de signos de puntuación: Se eliminaron signos de puntuación como `.,!?`.
6. Eliminación de stopwords: Se eliminaron palabras comunes sin valor semántico como artículos y preposiciones.

```
    text = RE_URL.sub(' ', text)

# 3) Quitar menciones (o al menos el @)
if remove_mentions:
    text = RE_MENTION.sub(' ', text)

# 4) Quitar símbolo '#' pero conservar la palabra del hashtag
if strip_hashtag_symbol:
    text = RE_HASHTAG.sub('', text)

# 5) Normalizar/quitar apóstrofes (don't -> dont)
text = RE_APOS.sub('', text)

# 6) Quitar emoticones/emojis
if remove_emoticons_emojis:
    text = RE_EMOTICONS.sub(' ', text)
    text = RE_EMOJI.sub(' ', text)

# 7) Quitar puntuación
if remove_punct:
    # translate es muy rápido para puntuación ASCII
    text = text.translate(str.maketrans('', '', string.punctuation))
    # limpiar cualquier resto no alfanumérico
    text = RE_NONALPHANUM.sub(' ', text)

# 8) Quitar números (con opción de conservar el token '911')
if remove_numbers:
    if keep_911:
        # Marcamos '911' para conservarlo
        text = re.sub(r'\b911\b', ' keepnineoneone ', text)
    text = re.sub(r'\d+', ' ', text)
```

Figura 2. Transformaciones de limpieza aplicadas

3.2. Análisis de Texto

Se aplicaron las siguientes técnicas para analizar el contenido de los tweets:

1. Frecuencia de palabras: Se analizaron las palabras más frecuentes en los tweets de desastres y no desastres.

Figura 3. Distribución de clases, accidentes y no accidentes

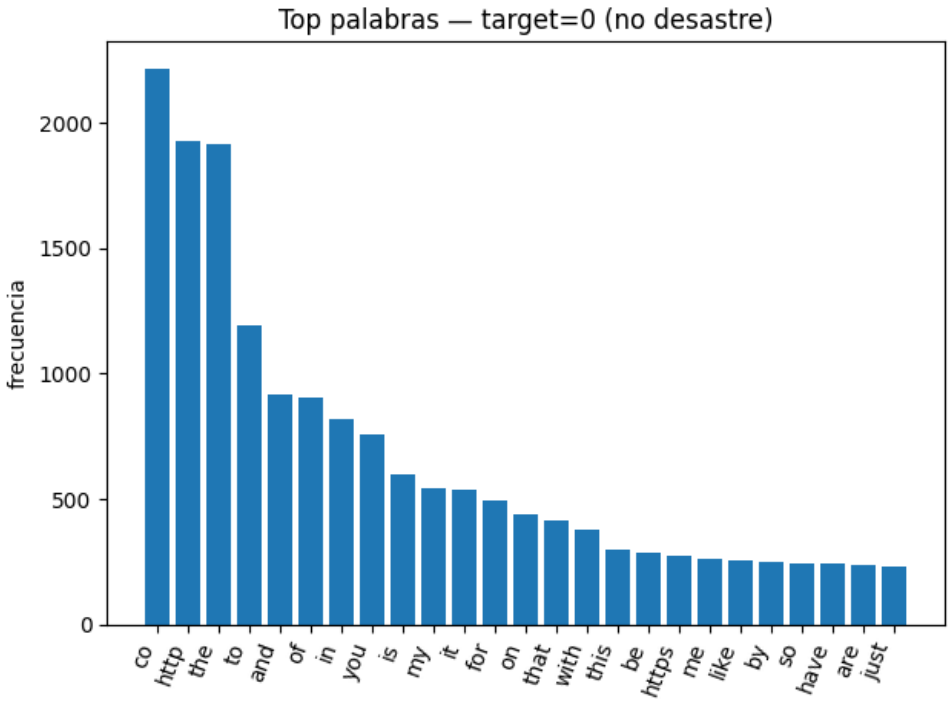


Figura 4. Top palabras no desastre (antes de limpieza)

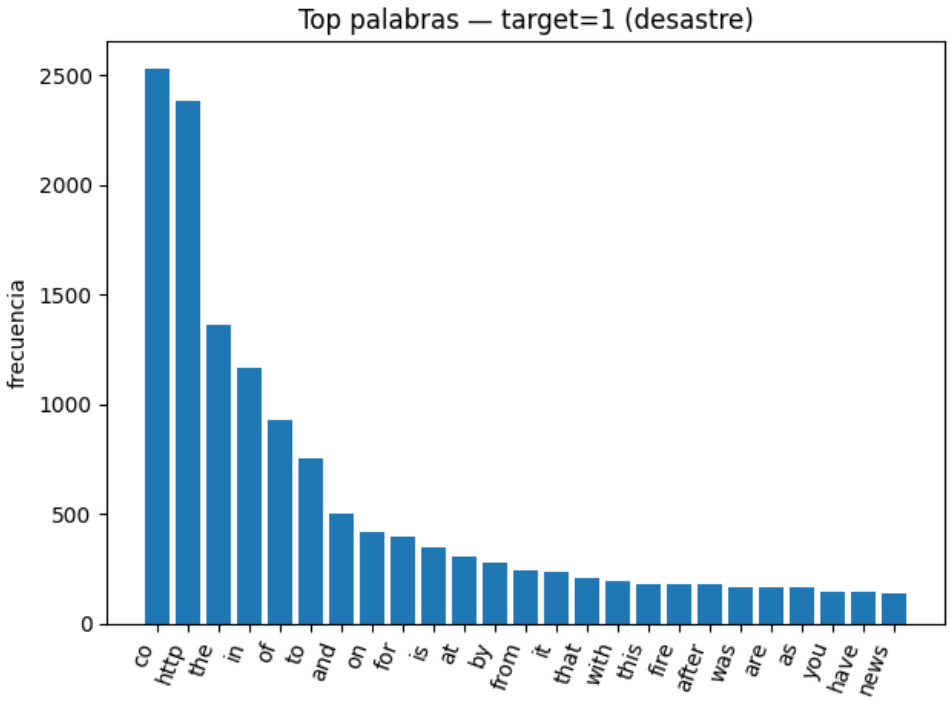


Figura 5. Top palabras desastre (antes de limpieza)

2. Nubes de palabras: Se generaron nubes de palabras para visualizar las palabras más comunes en cada clase. Los resultados se guardaron en las imágenes `wordcloud_target0.png` y `wordcloud_target1.png`, que muestran la frecuencia de

las palabras en los tweets no relacionados con desastres (target=0) y en los relacionados con desastres (target=1).

4. Análisis Exploratorio de los Datos

4.1. Frecuencia de Palabras

Se realizó un análisis de las palabras más frecuentes en los tweets. A continuación, se muestran los resultados para las dos clases de tweets (luego de la limpieza).

1. Tweets no relacionados con desastres (target = 0): Las palabras más comunes fueron "new", “one”, “dont”, “video”, “body”, “people”, etc.

2. Tweets relacionados con desastres (target = 1): Las palabras más frecuentes incluyeron "new", “via”, “fire”, “disaster”, “police”.

4.2. Nubes de Palabras

Las nubes de palabras generadas muestran visualmente las palabras que más se repiten en cada clase:
- Tweets no relacionados con desastres
- Tweets relacionados con desastres

Estas nubes permiten observar claramente qué términos son predominantes en cada categoría.

Figura 5. Nube de palabras generada para no desastres

Figura 6. Nube de palabras generada para desastres

4.3. Análisis de N-Gramas

Se generaron unigramas, bigramas y trigramas para identificar patrones contextuales que pudieran mejorar la clasificación. Los bigramas proporcionaron información valiosa sobre las combinaciones de palabras que ocurren con frecuencia en los tweets de desastres, como "suicide bomber", "nothern california", etc. Lo cual nos proporcionaba sobre ubicaciones comunes o más mencionadas de desastres y tipos de desastres. En cuanto a los no desastres, encontramos bigramas como: “crossed body”, “liked video”, etc. Que nos hablan de temáticas comunes o trends. Los trigramas por su parte nos dieron ideas aún más completas, como “northern california wildfire” lo que nos dice que hubo un incendio grave en esa localidad.

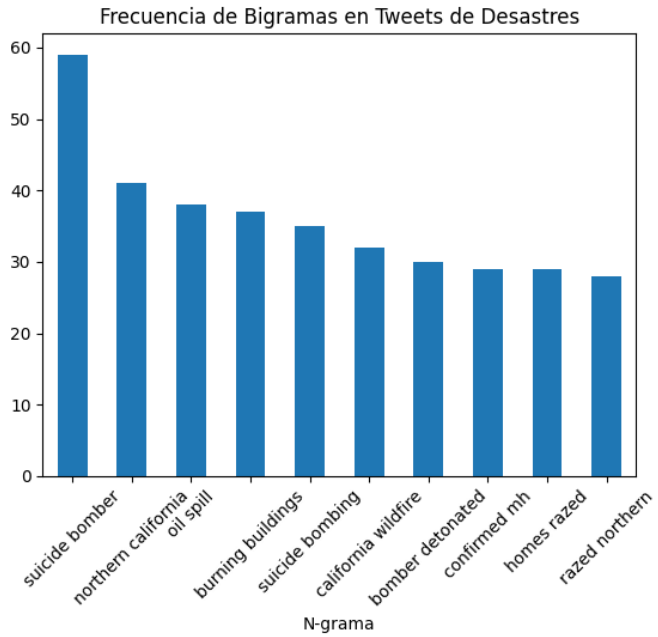


Figura 7. Distribución del bigrama generada para desastres

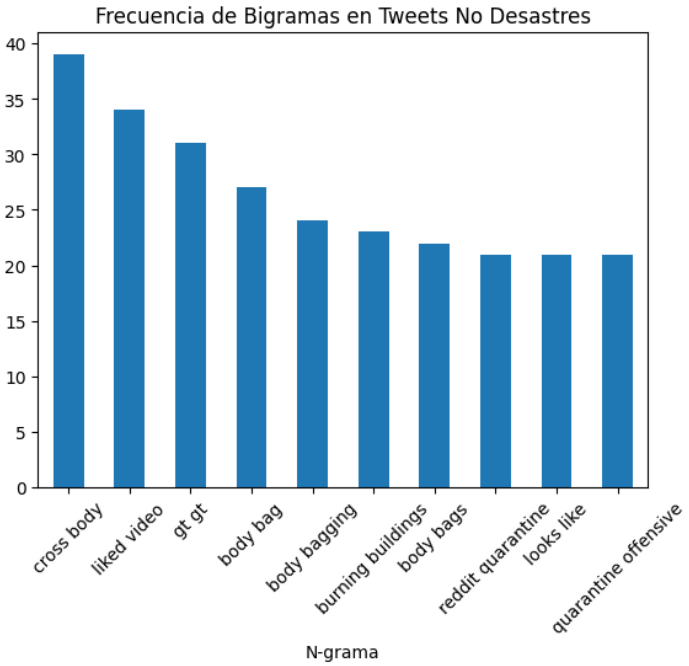


Figura 8. Distribución del bigrama generada para no desastres

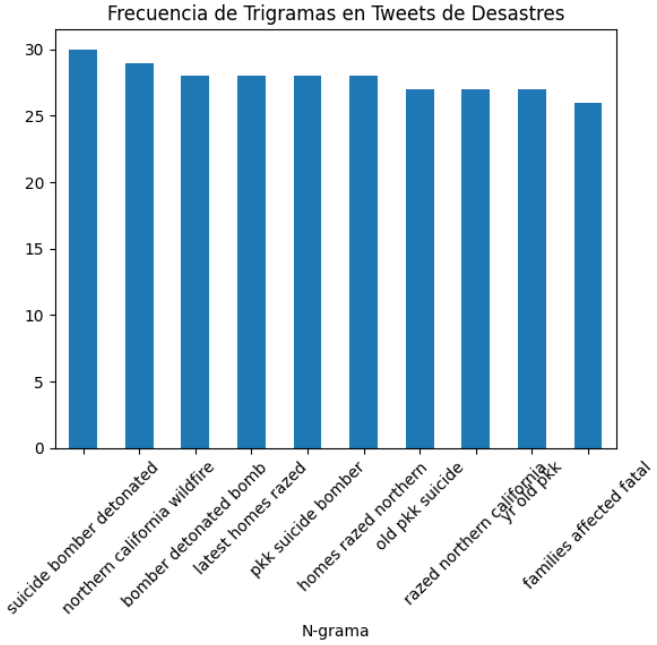


Figura 9. Distribución del trigrama generada para desastres

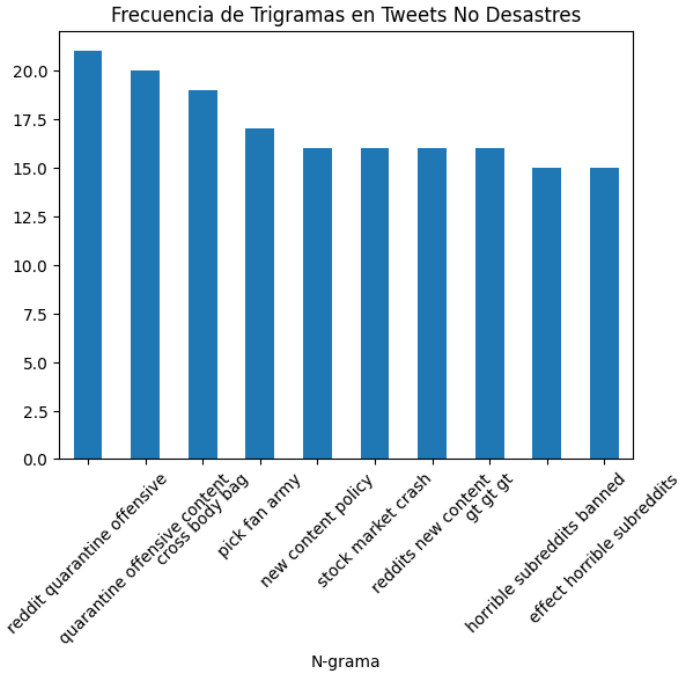


Figura 10. Distribución del trigrama generada para no desastres

5. Entrenamiento del Modelo

5.1. Modelo de Clasificación

Se entrenaron dos modelos de clasificación para predecir si un tweet está relacionado con un desastre (target = 1) o no (target = 0):

Naive Bayes Multinomial

Este modelo probabilístico es ampliamente utilizado para clasificación de texto debido a su simplicidad y buen desempeño en problemas de alta dimensionalidad como los TF-IDF de los tweets.

Se utilizó un pipeline que combinó la vectorización TF-IDF del texto con el clasificador Naive Bayes.

Support Vector Machine (SVM) con kernel lineal

Este modelo busca un hiperplano que separe las dos clases en el espacio de características.

Se utilizó un pipeline similar al de Naive Bayes, con vectorización TF-IDF y un clasificador SVM lineal.

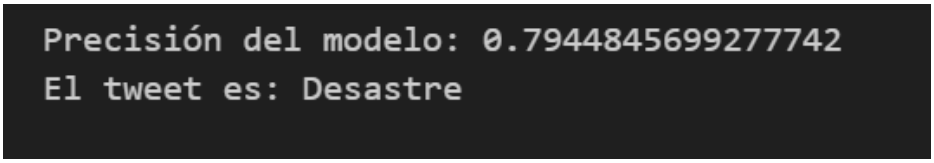
5.2. Evaluación del Modelo

Los modelos fueron evaluados utilizando el conjunto de prueba (20% de los datos), obteniendo las siguientes precisiones:

Modelo	Precisión
Naive Bayes	0.7945
SVM (lineal)	0.7892

- Ambos modelos alcanzan una precisión cercana al 79%, lo que indica un desempeño aceptable para un problema de clasificación de texto.
- Naive Bayes muestra un ligero margen de ventaja sobre SVM en este conjunto de datos.
- La diferencia entre ambos modelos es mínima, lo que sugiere que tanto enfoques probabilísticos como basados en márgenes pueden ser efectivos para esta tarea de clasificación.
- La función de clasificación desarrollada permite ingresar un nuevo tweet y obtener una predicción inmediata de la categoría.

Earthquake just hit the city, buildings are collapsing and people are trapped! Please stay safe and help if you can. #Earthquake #Disaster



Just finished a great workout at the gym! Feeling strong and motivated to keep going. #Fitness #Motivation

7. Conclusiones

1. La limpieza de los tweets permitió resaltar las palabras más relevantes, facilitando la comprensión de las características de cada clase.

2. Unigramas, bigramas y trigramas ayudaron a capturar patrones contextuales. Por ejemplo, “northern california wildfire” indica un evento específico de desastre, mientras que bigramas de no desastres reflejan tendencias comunes como “liked video”.

3. Los tweets de desastres destacaron palabras como “fire”, “disaster” y “police”, mientras que los no desastres mostraron términos como “video”, “body” y “people”.

4. Tanto Naive Bayes como SVM lineal alcanzaron precisiones cercanas al 79%, demostrando que los métodos de minería de texto son efectivos para esta tarea.

8. Recomendaciones

- ✓ Optimización del Modelo: Para mejorar aún más el modelo, se pueden explorar otras técnicas como redes neuronales o modelos más complejos.

- ✓ Aumento de Datos: Si el desempeño del modelo disminuye, se podría considerar aumentar el conjunto de datos con más ejemplos de tweets relacionados con desastres.

9. Referencias

Jurafsky, D., & Martin, J. H. (2014). -Speech and Language Processing-.

Feinerer, I., Hornik, K., & Meyer, D. (2008). Text Mining Infrastructure in R. -Journal of Statistical Software-, 25(5), 1–54.

NLTK Documentation: https://www.nltk.org/

Wordcloud Documentation: http://amueller.github.io/word_cloud/