PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Procesamiento de Lenguaje Natural

Manuel Espinoza Quintero

IIC2613 — Inteligencia Artificial
Profesora Jocelyn Dunstan
Julio de 2024

1. Introducción

El Procesamiento de Lenguaje Natural (NLP) es una subdisciplina de la inteligencia artificial que se enfoca en la interacción entre los computadores y las personas a través del lenguaje natural. Una de las tareas más comunes y útiles en NLP es la clasificación de texto, que involucra asignar categorías predefinidas a fragmentos de texto.

En este proyecto, se abordará la tarea de **clasificación de emails de spam**. El objetivo es desarrollar un modelo que pueda distinguir automáticamente entre correos electrónicos legítimos y correos electrónicos no deseados (spam). Esta tarea es de gran relevancia y utilidad por varias razones:

- **Seguridad y Privacidad:** Los emails de spam pueden contener enlaces maliciosos, intentos de phishing y otros contenidos dañinos que ponen en riesgo la seguridad y privacidad de los usuarios.
- **Eficiencia:** Un sistema de clasificación de spam eficiente reduce el tiempo que los usuarios necesitan para revisar y eliminar correos electrónicos no deseados, mejorando así la productividad.
- **Protección de Recursos:** Filtrar correos electrónicos de spam ayuda a proteger los recursos de la red y el almacenamiento, evitando el uso innecesario de ancho de banda y espacio en los servidores de correo.
- **Experiencia del Usuario:** Un sistema efectivo de detección de spam mejora la experiencia del usuario al mantener la bandeja de entrada limpia y relevante.

2. Datos y Métodos de Preprocesamiento

Utilizamos un dataset de correos electrónicos que contiene mensajes etiquetados como 'spam' o 'ham'. El dataset incluye información textual de los correos y su respectiva categoría. Proporcionado por Team AI en Kaggle [1].

2.1. Desbalance de Clases

La gráfica en el anexo muestra la distribución de las clases en el dataset, indicando un claro desbalance entre las categorías 'ham' (correos electrónicos legítimos) y 'spam' (correos electrónicos no deseados).

Este desbalance puede llevar a que los modelos de aprendizaje automático se vuelvan sesgados hacia la clase mayoritaria ('ham'), por lo que se deben considerar técnicas para mitigar este efecto en el futuro.

2.2. Análisis Cualitativo y Datos Nulos

Se realizó un preprocesamiento de los datos que incluyó la conversión de textos a minúsculas, eliminación de puntuación y stopwords, y tokenización. No se encontraron datos nulos en el dataset, lo cual facilitó el proceso de preprocesamiento.

3. Implementación Realizada

Para resolver la tarea de clasificación de emails de spam, implementamos dos modelos diferentes: Naive Bayes y SVM. A continuación, se describe el flujo de trabajo seguido para cada modelo:

- **Carga y Exploración de Datos:** Se cargaron los datos y se realizó un análisis exploratorio inicial para entender la distribución y características del dataset.
- **Preprocesamiento de Datos:** Incluyó la conversión a minúsculas, eliminación de puntuación, eliminación de stopwords y tokenización.
- **Vectorización:** Se utilizó TF-IDF para convertir los textos preprocesados en vectores numéricos.
- **Entrenamiento del Modelo:** Se entrenaron dos modelos diferentes (Naive Bayes y SVM) utilizando los datos vectorizados.
- **Evaluación del Modelo:** Se evaluó el desempeño de cada modelo utilizando métricas como precisión, recall y f1-score.
- **Análisis de Resultados:** Se compararon los resultados obtenidos para determinar el mejor modelo.

3.1. Diagrama de Bloques

El diagrama de bloques en el anexo ilustra el flujo de trabajo seguido en el proyecto.

4. Resultados Obtenidos

A continuación, se presenta una tabla con los resultados obtenidos para cada modelo:

Modelo	Precisión	Recall	F1-Score
Naive Bayes	0.97	0.88	0.92
SVM	0.99	0.96	0.98

Cuadro 1: Resultados de los Modelos

5. Conclusiones

Los resultados obtenidos muestran que el modelo SVM proporciona el mejor rendimiento para la clasificación de emails de spam y luego por el modelo Naive Bayes. A continuación, se resumen los aspectos logrados y no logrados, así como posibles mejoras y trabajo futuro:

5.1. Aspectos Logrados

- Alta precisión en la clasificación de correos de spam utilizando el modelo SVM.
- Implementación exitosa de dos modelos diferentes para comparación.
- Análisis detallado de la distribución de datos y preprocesamiento efectivo.

5.2. Aspectos No Logrados

- El recall del modelo Naive Bayes para la clase 'spam' es relativamente bajo.

5.3. Posibles Mejoras y Trabajo Futuro

- Implementar técnicas de balanceo de clases para mejorar el recall del modelo Naive Bayes.
- Afinar los hiperparámetros de los modelos para mejorar su precisión.
- Explorar otros modelos avanzados como Random Forest o Gradient Boosting para comparar su desempeño.
- Continuar recolectando datos de spam para enriquecer el dataset y mejorar la capacidad de generalización de los modelos.

5.4. Desafíos y Aprendizajes

Durante el desarrollo de la tarea, uno de los principales desafíos fue manejar el proceso de vectorización de los textos.

- **Vectorización de Textos:** La vectorización es un paso muy importante en el procesamiento de lenguaje natural, ya que convierte los textos en una representación numérica que los modelos de aprendizaje automático pueden entender. Elegir la técnica adecuada de vectorización (como Bag of Words, TF-IDF, o embeddings) y ajustar sus parámetros requiere una comprensión profunda del impacto de estas decisiones en el rendimiento del modelo.
- **Aprendizajes:** A través de este proyecto, se aprendió que:
  - La técnica de vectorización debe ser seleccionada en función de las características del dataset y del problema específico que se desea resolver. En este caso, TF-IDF se demostró eficaz para la tarea de clasificación de spam.
  - La normalización y el ajuste de parámetros como la frecuencia mínima de palabras y el tamaño del vocabulario pueden tener un impacto significativo en el rendimiento del modelo.
  - Es fundamental experimentar con diferentes técnicas de vectorización y comparar su rendimiento para encontrar la mejor opción para el problema específico.

6. Anexos

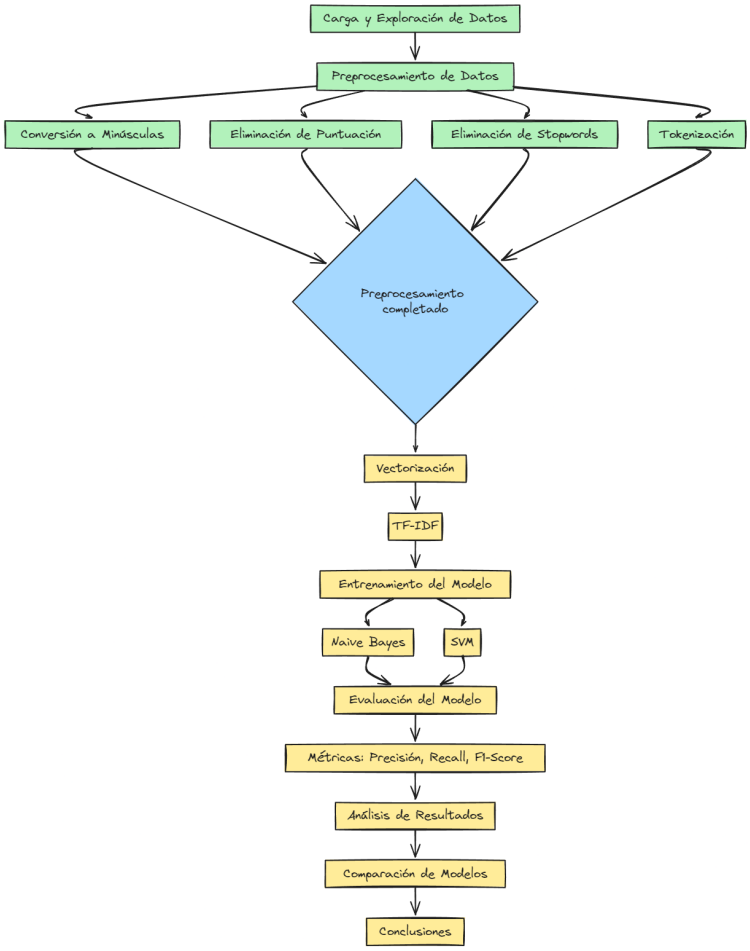

Figura 1: Diagrama de Bloques del Flujo de Trabajo

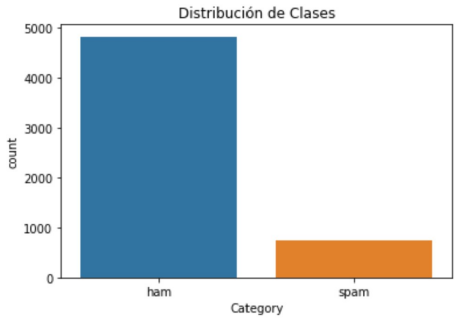

Figura 2: Distribución de Clases

Referencias

[1] Team AI. (n.d.). Spam Text Message Classification. Kaggle. Retrieved from `https://www.kaggle.com/datasets/team-ai/spam-text-message-classification`