Instituto Politécnico Nacional

Escuela Superior de Cómputo

Unidad de Aprendizaje:

Procesamiento del Lenguaje Natural

Práctica 5:

Text Classification

Profesor: Joel Omar Juárez Gambino

Equipo 3:
Loeza Saldaña Sebastián

Fecha de entrega: 19 de noviembre 2024

Tarea a resolver:

En esta práctica, se quiere hacer un modelo que clasifique noticias en diferentes secciones como deportes o economía. Para lograr esto, se implementarán diversos algoritmos de clasificación

utilizando el corpus de noticias recopilado en la práctica 2. El proceso incluirá la normalización del texto mediante técnicas como lematización, eliminación de *stopwords* y limpieza de caracteres no deseados. Posteriormente, el texto será representado en distintos formatos, como binario, frecuencia y TF-IDF, con el objetivo de analizar su impacto en el desempeño de los algoritmos. Finalmente, se evaluarán los resultados para identificar las configuraciones y métodos de clasificación más efectivos, optimizando el rendimiento según las métricas seleccionadas (F1-SCORE). Este enfoque permitirá comparar y seleccionar los modelos más adecuados para el análisis del corpus.

Proceso de la normalización de texto:

La normalización es un paso crucial en el procesamiento del lenguaje natural (PLN), ya que permite reducir la variabilidad del texto y facilitar su interpretación por parte de los modelos de aprendizaje automático. Para la normalización del texto en esta práctica, se aplicaron diversas técnicas con el objetivo de preparar el corpus para su análisis y mejorar la precisión de los algoritmos de clasificación. Entre estas técnicas se incluyen la lematización, eliminación de stopwords, limpieza de caracteres no deseados y tokenización.

La primer técnica de la que se hbalará es la lematización, el cuál es el proceso de convertir las palabras a su forma base o lema. A diferencia de la stemming, que simplemente recorta las palabras a su raíz, la lematización considera el contexto y la gramática de la palabra para devolver su forma canónica. Por ejemplo, las palabras "correr" y "corriendo" se normalizan a "correr". Este proceso es esencial para reducir la dimensionalidad del corpus y mejorar la precisión del modelo, ya que asegura que diferentes formas de la misma palabra se traten de manera uniforme.

Otra técnica es la eliminación de stopwords que implica remover palabras comunes y poco informativas que no añaden valor semántico al análisis, como "y", "el", "de", "en", etc. Estas palabras se consideran ruido en el texto y pueden afectar negativamente el rendimiento del modelo si se dejan en el corpus. Al eliminar estas stopwords, se logra un texto más limpio y concentrado en términos que realmente aportan significado, lo cual es fundamental para la clasificación efectiva.

La tercer técnica es la limpieza de caracteres no deseados que se refiere a la eliminación de símbolos, números, y caracteres especiales que no son relevantes para el análisis del texto. Esta técnica es particularmente importante en textos que pueden contener errores tipográficos,

emoticonos o caracteres extraños que podrían interferir con la correcta interpretación del contenido. Por ejemplo, en un corpus de reseñas, es posible que se encuentren caracteres especiales que no aportan información significativa y que, por lo tanto, deben ser eliminados.
Finalmente, la tokenización es el proceso de dividir el texto en unidades más pequeñas llamadas "tokens", que pueden ser palabras, frases o incluso caracteres. Este paso es fundamental para la mayoría de los algoritmos de clasificación, ya que permite transformar el texto en una forma estructurada que puede ser fácilmente analizada. Dependiendo de los objetivos del análisis, se puede optar por tokenizar a nivel de palabra o de frase, lo que influirá en cómo se interpretan las relaciones entre las palabras en el contexto.

Además, se experimentó con diferentes combinaciones de estas técnicas para generar múltiples versiones del corpus, optimizando así el proceso de clasificación. Por ejemplo, en una configuración se aplicó lematización, tokenización y eliminación de *stopwords*, pero no se realizó limpieza adicional del texto. En otro caso, se eliminaron las *stopwords* y se tokenizó el texto, pero se omitió la lematización. Estas variaciones permitieron evaluar cómo cada técnica y su combinación influyen en el desempeño de los modelos, proporcionando un enfoque más robusto para seleccionar las estrategias de normalización más adecuadas según las características del corpus y los objetivos del proyecto.

Representaciones de texto:

Para poder entrenar el modelo, se optó por utilizar tres diferentes representaciones las cuáles son TF-IDF, Binario y Frecuencia. Cada una de estas representaciones aborda el problema de la clasificación de texto desde una perspectiva única, permitiendo al modelo captar diferentes aspectos del contenido textual.
En la representación TF-IDF es una técnica comúnmente utilizada en el procesamiento de lenguaje natural. Se basa en dos conceptos fundamentales: la frecuencia de un término en un documento (TF) y la frecuencia inversa de documentos que contienen ese término (IDF). La idea es que las palabras que son frecuentes en un documento pero raras en otros documentos son más relevantes para el contenido de ese documento. Este enfoque ayuda a reducir el peso de términos comunes que no aportan información significativa, como "y", "el", "en", etc. De este modo, TF-IDF

proporciona un vector que refleja la importancia de cada palabra en el contexto del conjunto de documentos.

Por otro lado, la representación binaria es una forma simplificada de codificar la presencia o ausencia de términos en un documento. En este método, cada término en el vocabulario se representa como un valor binario: 1 si el término está presente en el documento y 0 si no lo está. Esta representación es especialmente útil en contextos donde la mera existencia de una palabra es más relevante que su frecuencia. Sin embargo, puede perder información sobre la relevancia de los términos, ya que no considera cuántas veces aparece cada palabra.

La tercera representación es la representación de frecuencia de término (Term Frequency) simplemente cuantifica cuántas veces aparece cada palabra en un documento. Este método es intuitivo y fácil de implementar, ya que permite al modelo captar la importancia de cada término en relación con otros términos dentro del mismo texto. No obstante, puede ser sensible a términos muy comunes que pueden distorsionar el significado general del documento.

Métodos de Machine Learning:

En esta práctica se implemtaron 5 modelos diferentes los cuales son Regresión Logística, Random Forest, KNN, MultinomialNB y SVC. Cada uno de estos modelos presenta características particulares que los hacen adecuados para distintos tipos de problemas de clasificación. Una breve descripción de estos algoritmos asi como sus ventajas y desventajas son las siguientes:

El primer modelo mencionado es la Regresión Logística, que es un modelo estadístico que se utiliza para predecir la probabilidad de una clase o evento, como la pertenencia a una categoría. En el contexto de clasificación de texto, este modelo se basa en la relación lineal entre las características del texto y la variable de clase. Su principal ventaja es la simplicidad y la interpretabilidad de los resultados. Sin embargo, su desempeño puede verse afectado en problemas no lineales o cuando hay muchas características irrelevantes.

El segundo modelo mencionado es el Random Forest y es un modelo de ensamble que utiliza múltiples árboles de decisión para mejorar la precisión de las predicciones. Cada árbol se entrena con una muestra aleatoria del conjunto de datos original, y las decisiones se toman en base a la mayoría de votos de todos los árboles. Este enfoque reduce el sobreajuste y mejora la robustez del modelo frente a datos ruidosos. Aunque Random Forest generalmente ofrece un alto rendimiento en clasificación, su complejidad puede hacer que sea más difícil de interpretar en comparación con modelos más simples.

El tercer modelo es K-Nearest Neighbors, este es un algoritmo de clasificación basado en la proximidad entre los puntos de datos. Al clasificar un nuevo punto, KNN busca los "k" ejemplos más cercanos en el conjunto de entrenamiento y asigna al nuevo punto la clase más común entre esos ejemplos. Una de las ventajas de KNN es su simplicidad y la capacidad de adaptarse a decisiones no lineales. Sin embargo, su desempeño puede verse afectado por la elección del valor de "k" y puede ser computacionalmente costoso en conjuntos de datos grandes.

El modelo Multinomial Naive Bayes es especialmente adecuado para la clasificación de texto, ya que asume que las características (palabras) son independientes entre sí. Este modelo utiliza la distribución multinomial para calcular la probabilidad de que un documento pertenezca a cada clase. Su principal ventaja es la rapidez en el entrenamiento y la predicción, así como su eficacia en problemas de clasificación de texto. Sin embargo, su suposición de independencia puede limitar su desempeño en conjuntos de datos más complejos.

Por último el Support Vector Classifier es un modelo que busca encontrar el hiperplano que mejor separa las diferentes clases en el espacio de características. SVC es particularmente potente en problemas de alta dimensionalidad, como la clasificación de texto, ya que puede utilizar diferentes núcleos (kernels) para manejar relaciones no lineales. Aunque SVC suele ofrecer un rendimiento superior, su entrenamiento puede ser más lento y requiere ajustar diversos parámetros, lo que puede complicar su implementación.

Resultados

Los mejores resultados de los modelos con diferentes configuraciones se muestran en la siguiente tabla.

Modelo	Parámetro	Normalización	Representación	F-Score prom
Regresión Lógistica	Class_weight= balanced	Sin Normalización	Binaria	0.86
Random Forest	class_weight= 'balanced', random_state=42	Stop Words + Lematización + Limpieza de Texto	Frecuencia	0.75
MultinomialNB	default	Stop Words + Lematización + Limpieza de Texto	Binario	0.86
KNN	n_neighbors = 5	Sin Normalización	TF-IDF	0.81
SVC	class_weight= 'balanced', random_state=42	Lematización + Limpieza de Texto	TF-IDF	0.80