UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Inteligencia Artificial

25/05/2024

Proyecto 3

NOMBRE ALUMNOS: Bernabé Nicolás Ricardo

Carlos Eduardo González Arceo

PROFESOR: Cecilia Reyes Peña

AYUDANTE: Tania Michelle Rubí Rojas

Problema: Detección de Idioma en texto

Cuando una computadora recibe una cadena de texto y se quiere hacer algún tipo de proceso con ella, el saber a qué lenguaje pertenece podría ayudar a facilitar su manejo.

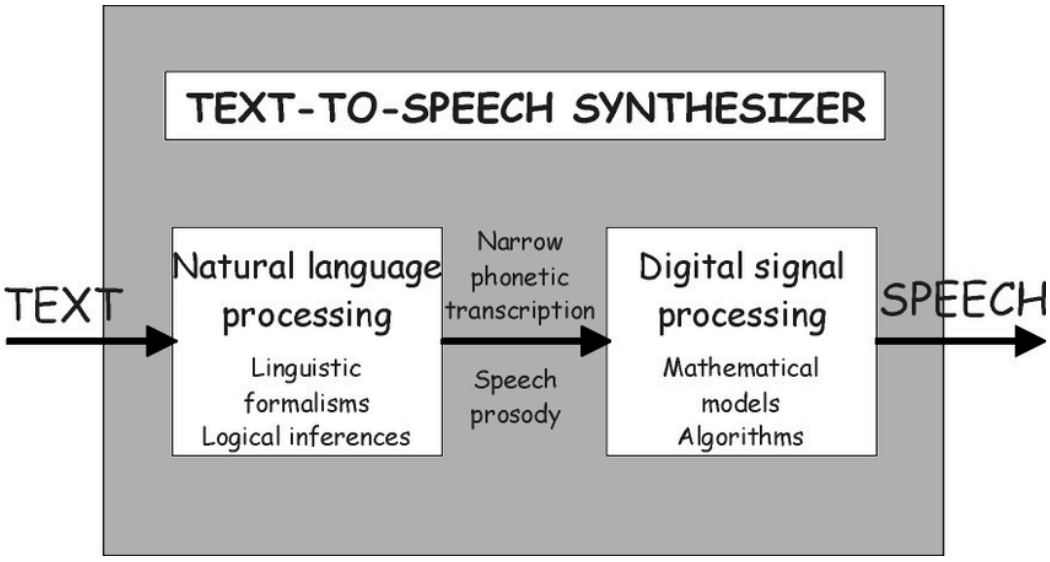
Ejemplos de usos de detección de idioma:

Detección de idioma antes de hacer traducción

Un ejemplo muy claro es la función “Detectar Idioma” del Google Traductor, si el usuario no reconoce el lenguaje del texto, puede simplemente utilizar la función que detecta el idioma y traduce al idioma del usuario.

Speech To text - Text to speech

Existen herramientas que transforman el audio a texto y el texto a audio, si le añadimos traducción automática se tendría que hacer lo siguiente:



La imagen es una simplificación de Text to speech en inglés, Se toma el texto y después se realiza un procesamiento de lenguaje natural y se obtiene información importante que ayudará al modelo de voz a generar una señal de audio que represente al texto respetando lo más que pueda el contexto del texto.

Cuando quisiéramos incluir más idiomas a nuestra herramienta, cada lenguaje tiene sus propias reglas fonéticas y sintaxis específicas, por lo que debemos de ser capaces de reconocer el lenguaje con el que vamos a trabajar, para poderlo adaptar a su modelo correspondiente.

La necesidad de comunicarnos con cualquiera, nos ha llevado a desarrollar este tipo de modelos.

Propuesta inicial:

En nuestro caso, trabajaremos con la detección de 4 idiomas, Español, Italiano Inglés y Francés, utilizaremos los siguientes dataset: Enlace, Enlace 2.
Se trata de un dataset utilizado para detección de lenguaje, cuenta con 1000 cadenas de texto con 22 lenguajes disponibles, según el proyecto decidido, sólo utilizaremos 3.

El segundo dataset es utilizado para utilizar sus textos en italiano.

Se utilizarán métodos de manejo de texto como son el Count Vectorizer y el Tf-idf Vectorizer para crear la matriz de pesos y después utilizaremos 2 modelos para comparar sus resultados entre sí. Estos serán Naive Bayes y Logistic Regression.

Desarrollo:

Para empezar, preparamos los dataset y obtuvimos los textos de los idiomas necesarios, además para ser utilizados en la regresión logística le agregamos un código para poder identificarlo.

	Text	language	code
0	en navidad de poco después de que interpretó ...	Spanish	0
1	según el censo de [] había personas residien...	Spanish	0
2	en la copa mundial de fútbol sub- de pitó los...	Spanish	0
3	ally y buttons encuentran el descodificador y ...	Spanish	0
4	los primeros habitantes se establecieron cerca...	Spanish	0
...	...	...	...
693	qual è stato il tuo errore, ti diamo da mangia...	Italian	3
694	narcisa ha cambiato i suoi modi in un primo mo...	Italian	3
695	Come' Il narcisismo di adesso Marian ha detto ...	Italian	3
696	immagino che non vorrebbe più pane d'oro adess...	Italian	3
697	Terry in realtà assomigli un po 'a quell'angel...	Italian	3

3698 rows x 3 columns

El dataset terminó con 3698 líneas, 698 son textos en italiano y el resto son 1000 textos por cada lenguaje.

Una vez teniendo el dataset limpio, procedemos utilizar el count vectorizer

Count vectorizer

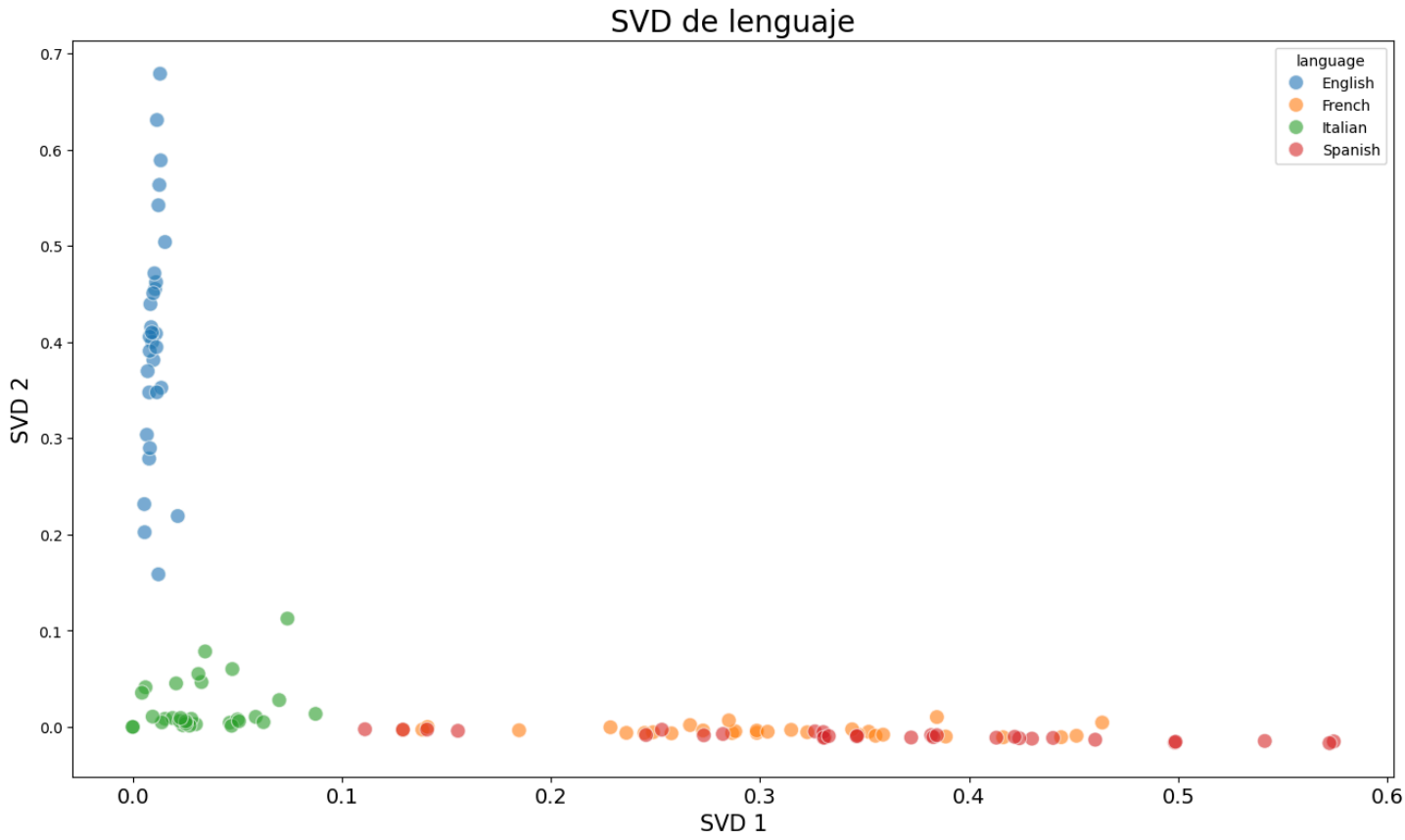
Es una herramienta que transforma texto en una matriz de conteo de tokens, es decir, se crea una matriz donde las columnas son tokens o palabras diferentes y los renglones son el conteo de las apariciones dentro del texto, generando así una matriz de token x conteo.

De esto, se obtuvieron 30582 tokens diferentes. una cantidad bastante manejable para una computadora. debemos recalcar que pertenecen a 4 lenguajes distintos, por lo que es claro que iban a ser bastantes.

Tf-idfVectorizer

Una vez teniendo nuestra matriz de conteo, hay que asignarles pesos para obtener información importante, en este caso Tf-idf es una muy buena opción porque nuestros vectores van a quedar llenos de información en zonas específicas del vector y vacías en otras, ya que al tener un bag of words de 4 idiomas y 30 000 tokens, hay tokens que sólo van a ser representativos en vectores de un solo idioma, cosa que se va a aprovechar utilizando tf-idf.

Visualización de los datos



Como se puede apreciar en la imagen, nuestros datos se concentran en 3 zonas principales, Verde(Italiano), Azul(Inglés), Rojo-Naranja(Español y francés),

Se aprecia que hay zonas de división claras que quizá se podrían utilizar para hacer uso de herramientas de clusterización pero esto daría resultados muy bajos en el caso de la zona Rojo-Naranja, esto se debe a que el Francés con el Español están bastante relacionados con respecto a palabras muy específicas, y estas darían valores relevantes para ambos lenguajes dentro del tf-idf.

Es curioso que no esté revuelto ahí el Italiano siendo que también es una lengua romance, pero muy posiblemente sea porque no compartimos palabras exactamente iguales aún compartiendo una sintaxis similar al español.

Esto se debe a que tf-idf no recuerda contextos ni orden, para eso se tendría que usar un modelo de n-gramas, pero para propósitos de este proyecto no usaremos ni clusterización ni n-gramas.

Manejo de los datos

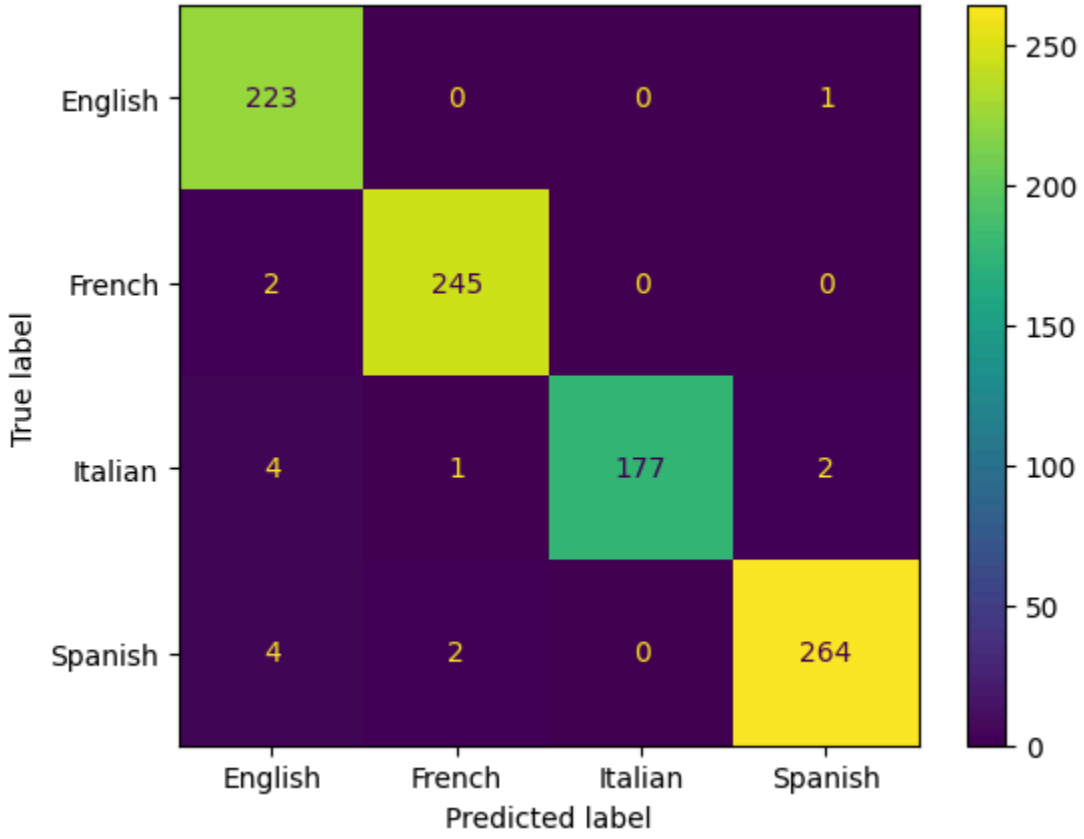
Para evaluar los modelos se utilizará cross validation, por lo que separaremos nuestro dataset 75% datos de entrenamiento y 25% datos de prueba.

Multinomial Naive Bayes

Multinomial Naive Bayes es una técnica que calcula una probabilidad de que un elemento a cierta clase, dado el el elemento, en nuestro caso el texto. entonces, utilizando la matrix resultante de tf-idf, es sencillo calcular la probabilidad de cada palabra dado el lenguaje.

Resultados de MNB

(Nota: Las clases son 0: "Spanish", 1: "Italian", 2: "French", 3: "English")



Como se puede apreciar en la matriz de confusión, hubo 10 textos los cuales fueron clasificados como inglés dentro de la predicción, por lo que parece que los textos que no lograron ser identificados fueron directamente al inglés. En el caso del Francés, 3 textos fueron clasificados como francés, pero eran Italiano-Español, quizá las palabras dentro de este texto también aparecían más en los textos en español o inglés. Observamos que el italiano y el español fueron los que más clasificaciones erróneas tuvieron.

En general el modelo de Multinomial Naive Bayes logró predecir correctamente la gran mayoría de los textos de prueba, por lo que podemos concluir que el modelo funciona correctamente.

	precision	recall	f1-score	support
English	0.96	1.00	0.98	224
French	0.99	0.99	0.99	247
Italian	1.00	0.96	0.98	184
Spanish	0.99	0.98	0.98	270
accuracy			0.98	925
macro avg	0.98	0.98	0.98	925
weighted avg	0.98	0.98	0.98	925

Analizando un poco más a profundidad, notamos que el modelo se mantiene consistente para todos los lenguajes, se esperaba que quizá el español y el francés tuvieran resultados más bajos por lo visto en el SVD. pero el modelo los logró reconocer correctamente.

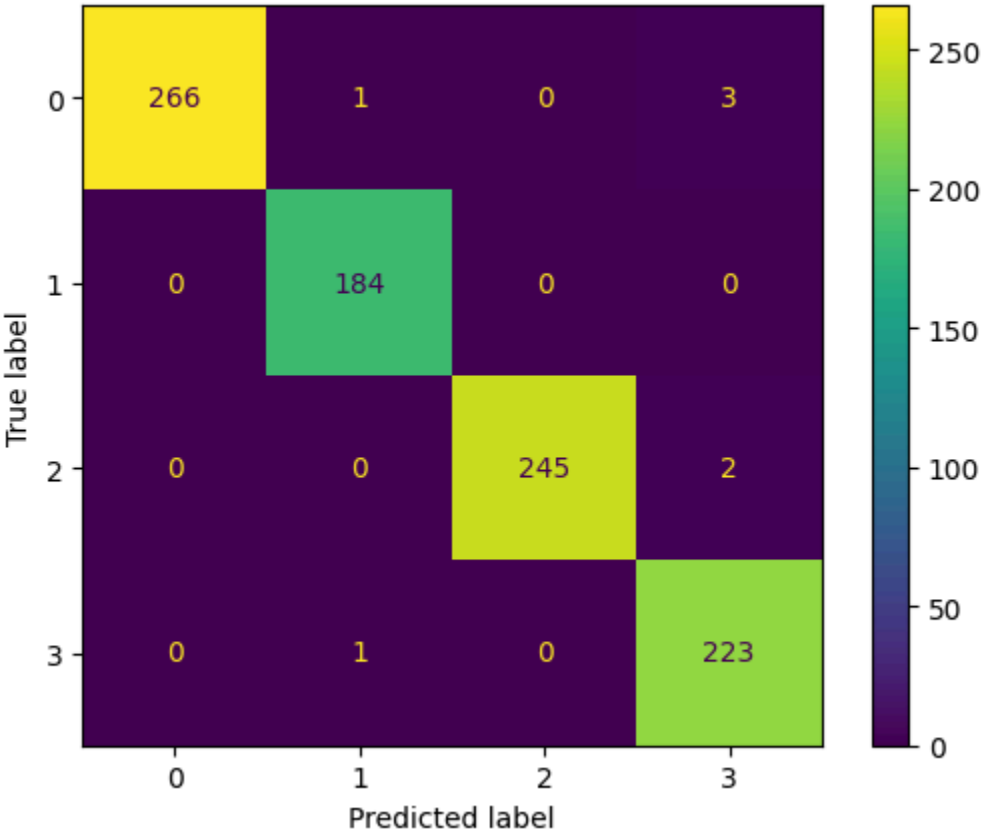
Concentrándose más en el f1-score vemos que todas las clases tienen un f1-score de 0.98, lo que nos hace concluir que efectivamente, el modelo es correcto y puede predecir textos.

Logistic Regression

Para poder comparar nuestro modelo de Naive Bayes, utilizamos Logistic Regression con su configuración Multinomial para hacer clasificación entre varias categorías, lo que hace para conseguirlo es utilizar one vs rest, que consiste en utilizar el funcionamiento de Logistic Regression varias veces en todas las combinaciones posibles, obteniendo una probabilidad en cada paso, después obtiene la probabilidad más grande, la cual representa a la categoría clasificada.

Resultados de Logistic regression

(Nota: Las clases son 0: "Spanish", 1: "Italian", 2: "French", 3: "English")



Notamos que este modelo tuvo un grado de error un poco más bajo que Naive Bayes, por lo que sólo la clase 3 (Inglés) e Italiano fueron clasificados erróneamente, por lo que también en este caso pareciera que Logistic regression es un mejor modelo.

	precision	recall	f1-score	support
0	1.00	0.99	0.99	270
1	0.99	1.00	0.99	184
2	1.00	0.99	1.00	247
3	0.98	1.00	0.99	224
accuracy			0.99	925
macro avg	0.99	0.99	0.99	925
weighted avg	0.99	0.99	0.99	925

Observando el reporte del modelo, Logistic regression obtuvo un accuracy del 99% y un f1-score del 99% por lo que resulta también en un modelo adecuado y satisfactorio.

Haciendo pruebas, notamos que si se utilizan textos muy cortos, los modelos no hacen predicciones correctas, a menos que se utilicen palabras tal cual aparecen en la matriz de tf-idf, por lo que se recomienda hacer uso de textos más o menos largos (10 palabras aprox).

Problemas enfrentados

Existió un problema al intentar utilizar el mismo conjunto de entrenamiento y prueba, ya que MultinomialNB sí puede aceptar categorías en texto pero regresión logística no, por lo que optamos en utilizar el código del lenguaje para ambos.

Reflexión

Dentro de las cosas que se pueden hacer en NLP, realizar cada pequeño proceso de manera correcta y eficiente nos provee de un pequeño bloque que pude ser utilizado con otros bloques para hacer modelos mucho más grandes, tales son las arquitecturas de los modelos de ChatGPT o Bert, el convertir texto a un vector es parte fundamental casi todos los procesos dentro de este campo, y conocer su funcionamiento y sus usos nos ayudarán en algún momento si es que nos dedicamos a esta rama.