UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Pasantía en la Corte Constitucional: Contribución al análisis de datos y depuración de accionados en Python

INFORME DE PASANTIA PARA OPTAR POR EL TÍTULO DE MATEMÁTICO
PROYECTO CURRICULAR DE MATEMÁTICAS

Nelson Miguel León García
Jefe inmediato: Adriana Urrea Bojacá
Profesor director: Luis Fernando Villarraga Poveda

Bogotá DC
Septiembre de 2023

Resumen

En este informe se presenta el desarrollo de pasantía en la Corte Constitucional de Colombia, específicamente en el equipo de apoyo a la presidencia de la corte. El enfoque principal de esta pasantía fue la depuración de las bases de datos relacionadas con los accionados en los casos de tutelas que llegan a revisión a la corte. Para llevar a cabo esta tarea, se empleó el lenguaje de programación Python, aprovechando las capacidades de Procesamiento de Lenguaje Natural dentro del campo de la inteligencia artificial.

Palabras clave: Accionado, expresiones regulares, Procesamiento de Lenguaje Natural, Tutela
Clasificación AMS: 62-07
Agradecimientos: Quiero expresar mi profundo agradecimiento a la memoria de mi mamá, quien siempre me brindó su estímulo y apoyo incondicional en mi trayectoria académica. También agradezco a mi hermana por sus valiosas enseñanzas y su apoyo constante a lo largo de los años. Además, agradezco a la Corte Constitucional por permitirme realizar mi práctica, en especial a Adriana Urrea y Ana María Giraldo, por su orientación y guía invaluable durante este proceso. Agradezco también a mis profesores y amigos que me acompañaron y apoyaron.

Índice

1. Objetivos **4**

1.1. Objetivo general: 4

1.2. objetivos específicos: 4

2. Preliminares **5**

2.1. Conceptos Constitucionales 5

2.1.1. Tutela 5

2.1.2. Accionado 5

2.1.3. Accionante 6

2.2. Conceptos de Procesamiento de lenguaje natural (PLN) 6

2.2.1. Expresiones regulares 6

2.2.2. Tokens 6

2.2.3. Distancia de Levenshtein 7

2.2.4. *Fuzzy Matching* 7

2.3. Clasificación y métricas de rendimiento 7

2.3.1. Matriz de confusión 8

2.3.2. *Accuracy* 8

2.3.3. *Recall* 9

2.3.4. *Precision* 9

2.3.5. Medida f1 (*f1-Score)* 9

2.3.6. Macro promedio y promedio ponderado 9

3. Modelo **11**

3.1. Herramientas 11

3.2. Datos 12

3.3. Limitaciones 12

4. Metodología **13**

4.1. Preprocesamiento de texto-limpieza de texto 13

4.2. Regex y fuzzy matching 16

4.3. Medición del modelo con matriz de confusión 18

4.4. conclusiones proyecto 19

5. Recomendaciones **20**

6. Conclusión pasantía **21**

1. Objetivos

1.1. Objetivo general:

Realizar depuración y filtrado de accionados por medio de técnicas de procesamiento de lenguaje natural con el objetivo de identificar y estandarizar las principales entidades que aparecen en las tutelas que llegan a revisión a la Corte Constitucional.

1.2. objetivos específicos:

1. Reconocer el rol de la Corte Constitucional en el sistema jurídico Colombiano.
2. Comprender la funcionalidad del procesamiento de lenguaje natural y sus conceptos básicos.
3. Analizar las bases de datos de accionados de la Corte para posteriormente poder depurarlas por medio de un modelo de procesamiento de lenguaje natural.
4. Analizar los resultados del modelo y evaluarlo a través de métricas de rendimiento.

2. Preliminares

Es importante señalar que, directamente la Corte Constitucional no falla las Tutelas, si no que, es la Corte la facultada para la revisión de los fallos emitidos por todos los jueces en Colombia, por medio de una figura que se denomina, revisión de Tutelas, esto con el fin de unificar jurisprudencia. Es así, como en el año 2022, la Corte Constitucional revisó más de 633 mil Tutelas, y las estadísticas oficiales muestran una tendencia creciente en comparación con años anteriores. Esta creciente carga de trabajo subraya la importancia de las matemáticas y su aplicación en la estadística y programación, que se convierten en aliados fundamentales para identificar y optimizar los datos mediante técnicas de análisis de datos, como el procesamiento de lenguaje natural (PLN).

En esta sección se definen los principales conceptos constitucionales utilizados y se realiza una introducción y definición sobre algunas herramientas utilizadas en el procesamiento de lenguaje natural.

2.1. Conceptos Constitucionales

La Corte Constitucional es una institución de la Rama Judicial del Poder Público creada mediante la adopción de la Constitución de 1991 con el fin de guardar la integridad y supremacía de la Carta Política.[2] La Corte, tiene a cargo la labor de velar por la integridad de los derechos humanos fundamentales, es así, como se le atribuye la facultad de revisar, los fallos de Tutela de todos los jueces nacionales, y de esta revisión, se pueden sustraer datos, que permitan focalizar las acciones jurídicas.

El mecanismo constitucional para proteger los derechos de los ciudadanos es la tutela, a continuación se define junto con los conceptos de accionado y accionate

2.1.1. Tutela

Es una acción constitucional establecida en el artículo 86 de la Carta Política de 1991, que permite a toda persona en el territorio nacional acceder a la protección especial e inmediata de sus derechos fundamentales cuando éstos resulten amenazados o vulnerados por algún órgano del Poder Público o particulares.[2]

2.1.2. Accionado

Es la persona o entidad contra la cual se dirige la demanda o se ejerce una pretensión jurídica, para el caso de la Tutela, el accionado, es la persona, natural o jurídica, ya sea pública o privada, quien, por acción u omisión, vulnera, ataca u restringe a otro u otros un derecho fundamental.

2.1.3. Accionante

Es la parte demandante o quien inicia una acción legal contra el accionado, esta, puede ser una persona o entidad que presenta una demanda o ejerce una pretensión jurídica buscando obtener un derecho, una reparación o una solución a una controversia, en pocas palabras es quien inicia la acción jurídica.

2.2. Conceptos de Procesamiento de lenguaje natural (PLN)

El Procesamiento de lenguaje natural representa una disciplina informática que se dedica a programar a las computadoras para interpretar la comunicación humana. Esta rama de la inteligencia artificial emplea técnicas como el aprendizaje automático (machine learning) y el aprendizaje profundo (deep learning) para lograr que las máquinas sean capaces de procesar y generar lenguaje humano[4].

El PLN, interactúa en la vida cotidiana de los seres humanos, por medio de los traductores en línea, asistentes virtuales, correctores de ortografía y gramática, entre otros, estos pocos ejemplos, muestran, como las computadoras, instruidas por el Procesamiento de Lenguaje Natural, logran procesar y ejecutar, el lenguaje o formas de comunicación humanas, de una manera amena y efectiva.

El problema que se aborda es la estandarización de entidades que figuran como accionadas en las tutelas que llegan a la Corte Constitucional. Para entender cómo se ha desarrollado el intento de resolver este problema, es esencial considerar los siguientes conceptos fundamentales.

2.2.1. Expresiones regulares

Una expresión regular, también conocida como regex (regular expression) es una secuencia de caracteres que define un patrón de búsqueda en un texto. En términos formales, es una notación algebraica empleada para describir un conjunto de secuencias de caracteres [3]. Al aplicar un regex, se recorre el corpus o texto de entrada para encontrar todas las ocurrencias que se ajusten al patrón definido, devolviendo así las coincidencias encontradas.

2.2.2. Tokens

El token es una unidad semántica básica que resulta del proceso de tokenización. Cuando se realiza la tokenización de un texto, este se divide en fragmentos individuales llamados "tokens", que pueden ser palabras, números o signos de puntuación [4]. Un método sencillo para realizar esta tarea es dividir el texto en partes utilizando espacios en blanco como separadores.

2.2.3. Distancia de Levenshtein

La distancia de Levenshtein es una métrica utilizada para evaluar la similitud entre dos secuencias de caracteres. Se define como el número mínimo de acciones requeridas para transformar una secuencia en otra. Las operaciones permitidas son la inserción, eliminación y sustitución de un carácter[7]. Para calcular la distancia de Levenshtein entre dos cadenas S y T con longitud m y n respectivamente, se construye la matriz $Lev_{S,T}[n+1, m+1]$, y se calcula el valor de cada celda $Lev_{S,T}(i,j)$ en la matriz. El calculo de las celdas se define:

$$Lev_{S,T}(i,j) = \begin{cases} 0, & i=0, j=0 \\ j, & i=0, j>0 \\ i, & i>0, j=0 \\ \text{mín} \begin{cases} Lev_{S,T}(i-1,j)+1 \\ Lev_{S,T}(i,j-1)+1 \\ Lev_{S,T}(i-1,j-1)+f(i,j) \end{cases} & , \quad i>0, j>0 \end{cases}$$

donde $f(i,j) = 1$ si el i-ésimo caracter de S es diferente al j-ésima caracter de T, de lo contrario $f(i,j) = 0$. Finalmente, la celda $Lev_{S,T}(n,m)$ de la esquina inferior derecha de la matriz es la distancia entre las cadenas S y T. La interpretación de las filas del mínimo corresponden a una eliminación, una inserción y una sustitución.

2.2.4. *Fuzzy Matching*

La coincidencia difusa, o "fuzzy matching", es una tecnica computacional que compara dos cadenas de texto para determinar si son similares. Es útil para corregir errores en los datos, como errores de ortografía o inconsistencias en la forma de ingreso de datos. Se basa en funciones de similitud como la distancia de Levenshtein, Jaccard y Jaro-Winkler, que miden la distancia entre las cadenas. Una técnica común valida las tuplas entrantes contra las relaciones de referencia, asignándoles un coeficiente de similitud para aceptar o rechazar las coincidencias [1].

2.3. Clasificación y métricas de rendimiento

Un problema de clasificación tiene como objetivo asignar una etiqueta o categoría a un conjunto de datos de entrada, según sus características o atributos observados. Hay dos tipos de problemas de clasificación: binarios y multiclase. En un problema binario, solo hay dos opciones posibles, como el resultado de una prueba de embarazo que puede ser positivo o negativo. Por otro lado, en un problema multiclase, hay más de dos alternativas.

Las métricas utilizadas para evaluar el rendimiento del modelo de clasificación son las métricas asociadas a la matriz de confusión definida a continuación

2.3.1. Matriz de confusión

Una matriz de confusión es una tabla de contingencia utilizada para medir el desempeño de un modelo de clasificación. [6]. La siguiente tabla muestra la estructura de una matriz de clasificación binaria junto con sus respectivos valores:

		Predicción	
		Positivo	**Negativo**
Real	**Positivo**	Verdaderos Positivos (TP)	Falsos Negativos (FN)
	Negativo	Falsos Positivos (FP)	Verdaderos Negativos (TN)

La diagonal principal muestra las predicciones correctas y la diagonal secundaria indica las predicciones incorrectas. los valores se definen como:

- TP = *True positive*, son los verdaderos positivos, es decir la muestra es positiva y se clasifica como positiva.
- FP = *False positive*, son los falsos postivos, que corresponden al error tipo I, son las muestras negativas que fueron clasificadas como positivas.
- FN = *False negative*, es el error tipo II, la muestra es positiva y fue clasificada negativa.
- TN = *True negative*, son los verdaderos negativos, es decir, las muestras negativas clasificadas como negativas.

A continuación se presentan las métricas de la matriz de confusión:

2.3.2. *Accuracy*

Es la proporción de predicciones correctas sobre el total de muestras. Se calcula sumando las predicciones correctas en la diagonal de la matriz de confusión y dividiéndolas por el total de muestras

caso binario

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

clasificación multiclase:

$$Accuracy = \frac{\text{Clasificaciones correctas}}{\text{Todas las clasificaciones}}$$

2.3.3. *Recall*

El *recall* tambien conocido como sensibilidad, representa la tasa de verdaderos positivos, es el número de muestras positivas correctamente clasificadas en relación con el número total de muestras positivas.

$$recall = \frac{TP}{TP + FN}$$

2.3.4. *Precision*

Es el valor de predicción positiva, es decir, la proporción de muestras positivas que fueron correctamente clasificadas respecto al número total de muestras positivas predichas.

$$precision = \frac{TP}{TP + FP}$$

2.3.5. Medida fl (*f1-Score)*

Es la media armonica del *recall* y *precision*. El valor de la medida F varía de cero a uno, y los valores altos de la medida F indican un alto rendimiento de clasificación.

$$\textit{F1-score} = \frac{2precision * recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

2.3.6. Macro promedio y promedio ponderado

Existen diferentes formas de promediar los valores de las medidas binarias para las distintas clases, y cada una puede ser adecuada para algún caso [5]. Un promedio general no tendría en cuenta los diferentes pesos que puede tener cada variable, mientras que uno que dependa de la frecuencia de cada clase podria dar un mejor enfoque sobre el comportamiento de los datos.

La siguiente notación es importante para entender cada definición matemática de los promedios

- y es el conjunto de etiquetas verdaderas.

- $\hat{y}$ es el conjunto de valores predecidos por el modelo
- L Conjunto de etiquetas
- y_l subconjunto de y con etiqueta l
- $\hat{y}_l$ subconjunto de $\hat{y}$ con etiqueta l
- $P(A,B) := \frac{|A \cap B|}{|B|}$ para los conjuntos A y B
- $R(A,B) := \frac{|A \cap B|}{|A|}$ para los conjuntos A y B
- $F_1(A,B) := 2\frac{P(A,B)xR(A,B)}{P(A,B)+R(A,B)}$

Macro promedio (*macro avg*)

es una medida de evaluación de modelos de clasificación que calcula la media aritmética de las métricas binarias, otorgando igual peso a cada clase.

$$\text{Macro promedio de } \textit{precision:} \quad \frac{1}{|L|}\sum_{l \in L} P\left(y_l, \hat{y}_l\right)$$

$$\text{Macro promedio de } \textit{Recall:} \quad \frac{1}{|L|}\sum_{l \in L} R\left(y_l, \hat{y}_l\right)$$

$$\text{Macro promedio de } \textit{F1-Score:} \quad \frac{1}{|L|}\sum_{l \in L} F_1\left(y_l, \hat{y}_l\right)$$

Promedio ponderado *(weighted avg)*

Es una forma de ajustar la evaluación del modelo para tener en cuenta la distribución desigual de las clases. En lugar de asignar un peso igual a cada clase, el promedio ponderado asigna un peso proporcional a la frecuencia relativa de cada clase en los datos verdaderos.

$$\text{Promedio ponderado de } \textit{precision:} \quad \frac{1}{\sum_{l \in L}\left|y_l\right|}\sum_{l \in L}\left|y_l\right| P\left(y_l, \hat{y}_l\right)$$

$$\text{Promedio ponderado de } \textit{Recall:} \quad \frac{1}{\sum_{l \in L}\left|y_l\right|}\sum_{l \in L}\left|y_l\right| R\left(y_l, \hat{y}_l\right)$$

$$\text{Promedio ponderado de } \textit{F1-Score:} \quad \frac{1}{\sum_{l \in L}\left|y_l\right|}\sum_{l \in L}\left|y_l\right| F_1\left(y_l, \hat{y}_l\right)$$

3. Modelo

Para la Corte Constitucional, es importante, identificar, cuáles son las entidades que más se ven involucradas en la vulneración de derechos fundamentales, es por esta razón que el proyecto se enfocó, en identificar y estandarizar los datos de estas, de allí se seleccionaron cuatro grupos de entidades:

a. Entidades prestadoras de salud (EPS)

b. Fondos de pensiones

c. Ministerios de Gobierno

d. Unidad para la atención de víctimas (UARIV).

El modelo desarrollado, utiliza expresiones regulares, junto con un fuzzy matching para mejorar la precisión en la identificación de las entidades, y se evaluó su desempeño utilizando una matriz de confusión. Sin embargo, solo se probó el rendimiento con las EPS.

3.1. Herramientas

El proyecto se desarrolló completamente en Python, debido a sus amplios recursos en ciencia de datos. También se realizaron algunas verificaciones y etiquetados manuales en Microsoft Excel.

Las librerías de Python utilizadas son las siguientes:

- Pandas: Se utiliza para el análisis y la manipulación de datos. Proporciona estructuras de datos flexibles y de alto rendimiento para trabajar con datos tabulares, como hojas de cálculo y bases de datos relacionales.
- Re: Se emplea para trabajar con expresiones regulares y proporciona una amplia variedad de funciones, tales como la búsqueda, el reemplazo y la división.
- Spacy: Es una biblioteca de procesamiento del lenguaje natural. Proporciona herramientas para el etiquetado gramatical, el análisis sintáctico, la identificación de entidades nombradas entre otras. Spacy es muy popular en la comunidad de NLP debido a su velocidad y precisión.
- Spaczz: Proporciona coincidencia difusa y funcionalidad adicional de coincidencia de expresiones regulares para Spacy. Utiliza emparejadores del módulo fuzz de RapidFuzz que se basan en la distancia de Levenshtein.
- Sklearn: Es una biblioteca de aprendizaje automático que se utiliza para construir modelos predictivos. Ofrece herramientas para la clasificación, regresión, agrupación y otras funcionalidades similares.

3.2. Datos

Los datos trabajados son datos anonimizados, ya que no poseen ningún identificador que ayude a detectar cuáles son los accionados. Se tuvieron tres bases de datos de accionados: la primera con 16.899 registros, la segunda con 273.354 y la última con 12.926 entradas. Solo se conoce la fecha del primer conjunto de datos, que corresponde a la última semana de febrero. Las dos primeras bases de datos fueron utilizadas para desarrollar el algoritmo, y el tercer conjunto de datos se utilizó para evaluar el rendimiento del modelo. Dado que el objetivo era identificar a los accionados, solo se utilizó la variable de demandados, omitiendo otras variables como el número de expediente o la fecha. Es importante destacar que varias de las entidades accionadas tienen alcance a nivel nacional, lo que enriquece significativamente la base de datos en términos de información disponible.

3.3. Limitaciones

Dado que los datos son anonimizados, no pueden asociarse directamente con las entidades. Esta propiedad de los datos puede generar desafíos significativos debido a las variaciones en la forma en que las personas se refieren a las entidades y a los posibles errores de redacción, tipeo u ortografía.

Las siguientes consideraciones sobre la calidad de la escritura son clave para comprender la dificultad de trabajar con este tipo de datos:

- Variaciones en las referencias: Las personas pueden referirse a la misma entidad de diferentes maneras. Por ejemplo, una empresa podría mencionarse por sus siglas, su nombre completo o algún apodo. Esto complica la tarea de identificar todas las posibles referencias a la misma entidad.
- Nombres largos y complejos: Algunas entidades tienen nombres muy largos o complejos, lo que aumenta la probabilidad de cometer errores al mencionarlos con frecuencia.
- Errores de redacción y ortografía: Los errores de redacción, tipeo y ortografía son comunes en los datos escritos por personas. Estos errores pueden dificultar la identificación precisa de las entidades.
- Recordación inconsistente: Las personas pueden recordar y escribir nombres de entidades de manera inconsistente con el tiempo.

Debido a estos desafíos, es esencial realizar una depuración o limpieza de datos. La limpieza de datos implica la corrección de errores ortográficos, la estandarización de nombres de entidades y la identificación y consolidación de diferentes formas de referirse a una misma entidad.

Por otra parte, podría existir una alternativa viable para abordar la cuestión planteada. Esta alternativa radica en el uso de un sistema de asignación de identificadores exclusivos para poder identificar plenamente a una persona natural o jurídica. Un ejemplo de esto sería la cédula de ciudadanía y el número de identificación Tributaria NIT, que se utiliza en Colombia para identificar a las empresas y personas que realizan actividades económicas. Este enfoque simplificaría y agilizaría el proceso de identificación y clasificación.

Sin embargo, el NIT no es un dato requerido para la acción de tutela, ya que solicitar ese dato en una tutela no seria razonable dado que impondría barreras, tales como solicitar datos que le sean difíciles de conseguir al accionante, quien busca de manera inmediata garantizar un derecho fundamental o frenar la vulneración de este. Por lo tanto se optó por una solución en PLN planteada a lo largo del presente informe.

4. Metodología

La ejecución de esta práctica se dividió en múltiples fases. Inicialmente, se llevó a cabo un minucioso estudio acerca de la relevancia de la Corte Constitucional mediante un análisis exploratorio de datos e información disponible en la web. El objetivo primordial de esta etapa era determinar la importancia de la Corte Constitucional en el marco del sistema jurídico colombiano.

A continuación, se realizó un proceso de preprocesamiento del texto en una base de datos de accionados. Este proceso incluyó una cuidadosa limpieza de datos para eliminar ruido, signos de puntuación, palabras vacías y otros elementos innecesarios.

Posteriormente, se utilizó la herramienta regex para identificar y estandarizar los principales accionados de la base de datos. Además, se integró la técnica Regex con el uso de Fuzzy Matching, con el fin de perfeccionar la precisión. Dado que los nombres pueden estar escritos de varias maneras, esta combinación de técnicas contribuyó significativamente a perfeccionar la precisión.

Finalmente, se construyó una matriz de confusión con el propósito de evaluar el rendimiento del algoritmo, utilizando una base de datos que había sido etiquetada manualmente como referencia. Los resultados de esta evaluación proporcionaron información valiosa sobre la eficacia del algoritmo y su capacidad para identificar accionados con precisión.

4.1. Preprocesamiento de texto-limpieza de texto

Inicialmente solo se trabajó con la base de datos de 16.899 accionados, Los datos de los demandados sin modificar tienen el siguiente aspecto:

Demandado
JUZGADO 005 CIIVL MUNICIPAL DE EJECUCION DE SE...
SAVIA SALUDEPS YOTROS
NUEVA EPS
UNIDAD ADMINISTRATIVA ESPECIAL PARA LA ATENCIO...
REGISTRADURIA NACIONAL DEL ESTADO CIVIL COLOMB...
...
FIDUPREVISORA S.A.
SUPERINTENDENCIA DE SERVICIOS PUBLICOS DOMICIL...
UNIDAD ADMINISTRATIVA ESPECIAL PARA LA ATENCIO...
NUEVA EPS
SECRETARIA DISTRITAL DE TRANSITO Y SEGURIDAD V...

Figura 1: Algunos accionados de la base de datos

Con el fin de facilitar el procesamiento posterior del texto, se aplicaron diversas transformaciones mediante el uso de expresiones regulares. Estas transformaciones se llevaron a cabo para eliminar caracteres no informativos tales como signos de puntuación o caracteres no alfabeticos y convertir el texto a minúsculas. A continuación se presenta la función creada para realizar esta tarea:

```
def Limpiar_texto(texto):
    texto = unidecode(texto)
    # Eliminar cualquier signo de puntuación
    puntuación = r'.,;:¡!¿?@#$%&[\{}<>~=+\*/|\\_^`"\']'
    #puntuación = r'[^\w\s]|(?<=\w)\.(?=[a-zA-Z])'
    texto = re.sub(puntuación, '', texto)

    # Eliminar cualquier carácter que no sea una letra o un número
    alfanumérico = r'\w+(?:-\w+)*'
    texto = re.sub(alfanumérico,
                   lambda x: x.group(0).replace('-', ' '), texto)

    # Eliminar cualquier caracter que no sea una letra o un número

    #o un espacio en blanco
    texto = re.sub(r'[^a-zA-Z0-9\s]', '', texto)
    # eliminar espacios en blanco iniciales
    texto = re.sub(r"^\s+", "", texto)
    doc = nlp(texto.lower())

    return " ".join(unidecode(token.norm_)
                    for token in doc if not token.is_punct)
```

Posteriormente, el texto fue tokenizado para examinar la frecuencia de los tokens con el objetivo de identificar y eliminar las palabras irrelevantes para el análisis de texto debido a su alta frecuencia y falta de significado semántico. Esto permitió identificar algunas de las principales entidades al revisar los tokens que no fueron eliminados, ya que la combinación de varias de las palabras más frecuentes revela algunas de las entidades que más se repiten. La siguiente tabla muestra el conjunto de palabras de mayor frecuencia en la base de datos sin tener en cuenta las que fueron eliminadas.

Tokens	**Frecuencia**	**Tokens**	**Frecuencia**	**Tokens**	**Frecuencia**
eps	4039	colpensiones	752	instituto	521
alcaldia	1552	unidad	731	colombia	478
secretaria	1393	uariv	692	sanitas	473
nacional	1350	civil	685	seguridad	460
salud	1207	municipal	677	transporte	453
nueva	1169	ministerio	621	administrativa	440
juzgado	1014	transito	568	especial	432
bogota	876	victimas	534	reparacion	424

4.2. Regex y fuzzy matching

Después de realizar la limpieza de datos, se crea un diccionario de expresiones regulares que define un patrón de búsqueda para cada entidad accionada y devuelve la estandarización elegida para cada una. A continuación se muestra el diccionario de las EPS:

```
entidades = {"asmet salud eps": r"asme(.*)[dt]",
        "asociacion indigena cauca aic eps": r"\baic\b|^(?=.*asociacion)
        (?=.*indigena)(?=.*cauca).*",
        "aliansalud eps": r"(alian\s?salud|alianza salud)",
        "ambuq eps": r"ambuq|^(?=.*asociacion)(?=.*mutual)(?=.*barrios)
        (?=.*unidos)(?=.*quibdo).*",
        "anas wayuu eps": r"^(?=.*anas)(?=.*wayu(u)?).*",
        "cajacopi eps": r"^(?=.*caja\s?copi)(?=.*eps)",
        "capital salud eps": r"(?=.*capital)(?=.*salud)|(?=.*capital)
        (?=.*eps)",
        "capresoca eps": r"capresoca",

                                .
                                .
                                .

        "sura eps": r"(?=.*sura)(?=.*eps)|(?=.*entidad)(?=.*salud)
        (?=.*sura)"}
```

El patron de búsqueda de expresiones regulares no identifica errores ortográficos o de tipeo, dejando de clasificar gran parte de los datos, es por ello la necesidad de aplicar un modelo de *fuzzy matching* dado que la capacidad de búsqueda es superior y puede encontrar casos que el regex no logra identificar. El código implementado del *fuzzy matching* con las entidades prestadoras de salud buscados por la coincidencia difusa lucen de la siguiente forma:

```
doc_matches = []
matcher = FuzzyMatcher(nlp.vocab) # Creación de un objeto FuzzyMatcher

matcher.add("asmet salud eps", [nlp("asmet salud"), nlp("asmet"),
            nlp("eps asmet salud"), nlp("asmet salud eps")])
matcher.add("asociacion indigena cauca aic eps", [nlp("aic epsi"),
            nlp("epsi aic"), nlp("aic eps"), nlp("eps aic"),
            nlp("asociacion indigena cauca"),
            nlp("asociacion indigena cauca epsi"),
            nlp("epsi asociacion indigena cauca"),
            nlp("asociacion indigena cauca eps"),
            nlp("eps asociacion indigena cauca")])
matcher.add("aliansalud eps", [nlp("aliansalud"), nlp("aliansalud eps"),
            nlp("eps aliansalud")])
matcher.add("ambuq eps", [nlp("asociacion mutual barrios unidos quibdo"),
            nlp("ambuq")])
matcher.add("anas wayuu eps", [nlp("anas wayuu")])
matcher.add("cajacopi eps", [nlp("cajacopi eps"), nlp("eps cajacopi")])
matcher.add("capital salud eps", [nlp("capital salud")])
matcher.add("capresoca eps", [nlp("capresoca")])
                        .
                        .
                        .
matcher.add("sura eps", [nlp("sura eps"), nlp("eps sura")])
```

La implementación de la coincidencia difusa depende de un coeficiente de similitud o ratio que varía según sea conveniente, ya que es necesario limitar el alcance del *fuzzy matching* para reducir el riesgo de errores en la clasificación. Se realizaron varias pruebas en la ejecución del algoritmo para seleccionar el ratio idóneo para cada entidad, el criterio usado para su elección fue donde no hubiese ninguna clasificación errónea. El siguiente código muestra los coeficientes de las EPS.

```
clasificaciones = {
    ("asmet salud eps", 86): "asmet salud eps",
    ("aliansalud eps", 91): "aliansalud eps",
    ("ambuq eps", 86): "ambuq eps",
    ("anas wayuu eps", 75): "anas wayuu eps",
    ("cajacopi eps", 75): "cajacopi eps",
    ("capital salud eps", 92): "capital salud eps",
    ("capresoca eps", 78): "capresoca eps"
                        .
                        .
                        .
    ("sura eps", 86): "sura eps"}
```

Finalmente las expresiones regulares y el *fuzzy matching* fueron combinados en una sola función de búsqueda, que al encontrar los patrones de busqueda retorna en una nueva columna los nombres ya estandarizados de las entidades.

4.3. Medición del modelo con matriz de confusión

El modelo es evaluado como un modelo de clasificación multiclase, por lo tanto se utiliza la matriz de confusión para evaluar su rendimiento. Para medir el rendimiento del modelo, fue necesario realizar un etiquetado manual con muestras de cada entidad. Este etiquetado se llevó a cabo en Excel y solo se etiquetaron las EPS. Por lo tanto, el modelo solo se evaluó con estas entidades y no con los 4 grupos de entidades con los que se trabajó. La figura 2 muestra las métricas correspondientes a las primeras EPS en orden alfabético.

	entidad	precision	recall	f1-score	support
0	aliansalud eps	0.950000	0.987013	0.968153	154.000000
1	ambuq eps	0.909091	0.833333	0.869565	12.000000
2	anas wayuu eps	1.000000	1.000000	1.000000	14.000000
3	asmet salud eps	1.000000	0.998958	0.999479	1919.000000
4	asociacion indigena del cauca aic eps	1.000000	1.000000	1.000000	209.000000
5	cajacopi eps	1.000000	1.000000	1.000000	1390.000000
6	capital salud eps	1.000000	1.000000	1.000000	1128.000000
7	capresoca eps	1.000000	0.985876	0.992888	354.000000

Figura 2: Métricas matriz de confusión

Se puede observar que las métricas de rendimiento para cada caso presentan valores significativamente altos, lo que indica que el modelo funciona de manera efectiva.

48	accuracy	0.997777	0.997777	0.997777	0.997777
49	macro avg	0.809140	0.806143	0.807587	81429.000000
50	weighted avg	0.998850	0.997777	0.998310	81429.000000

Figura 3: Métricas matriz de confusión

El *accuracy*tiene un valor de 0,997 lo que indica que la proporción de predicciones correctas sobre el total de muestras es casi del 100 %. Además, el promedio macro está por encima del 80 % y el promedio ponderado supera el 99 %.Esta diferencia entre los promedios demuestra que el rendimiento por categorías es bastante alto, aunque en general el promedio es ligeramente inferior. Esto no es sorprendente, dado que no todas las etiquetas tienen el mismo número de muestras.

4.4. conclusiones proyecto

- Aunque las métricas de rendimiento fueron bastante altas, esto podría ser contraproducente porque podría arrojar un indicio de tener el modelo sobre entrenado, sin embargo el modelo fue probado con la ultima de las bases de datos y no con las 2 primeras con las que fue realizado.

- El *fuzzy matching* muestra una capacidad sólida para buscar entidades, por lo que se recomienda utilizarlo en tareas de procesamiento de lenguaje natural.
- La principal desventaja de utilizar el *fuzzy matching* es la calibración de los ratios, ya que un ratio demasiado bajo puede filtrar entidades que no corresponden a las buscadas, mientras que uno demasiado alto puede ser demasiado ajustado y dejaría por fuera datos que sí corresponden a los buscados.
- Esta practica laboral hace parte del fortalecimiento digital en la Justicia en Colombiana. La modernización de la justicia implica grandes ventajas como la extracción de información útil y relevante.

5. Recomendaciones

- Se recomienda evaluar constantemente los modelos con nuevos conjuntos de datos, dado que siempre pueden aparecer nuevas entradas que no se han tenido en cuenta antes y esto puede afectar severamente el rendimiento.
- Se sugiere revisar periódicamente los ratios del *fuzzy matching* y ajustarlos según corresponda, ya que pueden ser sensibles a datos nuevos.
- Se recomienda implementar diferentes algoritmos de PLN como el modelo NER (Named Entity Recognition) para comparar varias alternativas distintas a *fuzzy matching* y poder elegir el que mejor rendimiento tenga.
- Se sugiere que el programa de matemáticas de la Universidad Distrital considere la posibilidad de ampliar su oferta académica al incluir cursos especializados en estadística. Esta medida se justifica por la creciente importancia que los datos han adquirido en el ámbito laboral y profesional.
- Se recomienda al proyecto de matemáticas dictar cursos dedicados a la programación y el dominio de herramientas para el manejo de datos. Estas habilidades son de una importancia crítica en el entorno laboral actual.
- Se recomienda la inclusión de cursos que aborden el desarrollo y aplicación de modelos matemáticos prácticos en el entorno laboral. Esta iniciativa tiene la finalidad de preparar a los estudiantes para enfrentar los desafíos del mundo profesional de manera efectiva.

6. Conclusión pasantía

El uso de herramientas matemáticas y estadísticas en una institución judicial como la Corte Constitucional puede parecer inapropiado a simple vista debido a la percepción general de que el sistema legal se basa principalmente en la interpretación de leyes, jurisprudencia y argumentación legal. Sin embargo, en la era actual, donde la información y los datos desempeñan un papel fundamental en prácticamente todos los aspectos de la sociedad, incluso las instituciones judiciales se están viendo cada vez más presionadas para adoptar enfoques más analíticos y basados en datos.Además, es importante destacar la relevancia que tiene la interdisciplinariedad en el ámbito laboral, ya que el mundo real opera de esa forma. Es muy enriquecedor a nivel educativo poder interactuar con profesionales de diferentes áreas y poder aportar como matemático.

Referencias

[1] Surajit Chaudhuri, Kris Ganjam, Venkatesh Ganti, and Rajeev Motwani. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, page 313–324, New York, NY, USA, 2003. Association for Computing Machinery.

[2] Corte Constitucional. Preguntas frecuentes. https://www.corteconstitucional.gov.co/ preguntasfrecuentes.php, 2023.

[3] Daniel Jurafsky. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Stanford University, third edition draft edition, 2019.

[4] Uday Kamath, John Liu, and James Whitaker. *Deep Learning for NLP and Speech Recognition.* Springer, 2019.

[5] scikit learn. Metrics and scoring: quantifying the quality of predictions. https://scikit-learn.org/stable/modules/model$_evaluation$, 2023.

[6] Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 17(1):168–192, 2018.

[7] Shengnan Zhang, Yan Hu, and Guangrong Bian. Research on string similarity algorithm based on levenshtein distance. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 2247–2251, 2017.