Universidad Autónoma Metropolitana

Unidad Cuajimalpa

División de Ciencias de la Comunicación y Diseño

Licenciatura en Tecnologías y Sistemas de Información

Herramienta automática para diferenciar zonas dialectales de México en Twitter

Presenta:

Díaz Ávalos Alejandro

Asesores:

M.C. Ramírez de la Rosa Adriana Gabriela
Dr. Villatoro Tello Esaú

México, CDMX 2019

DEDICATORIA

A mi madre, quien es la prueba de que Dios existe en mi vida.

Índice general

Capítulo 1

Introducción

Todos los textos están caracterizados por dos componentes: el fondo y la forma. El fondo es aquella idea que queremos expresar; la forma, por otra parte, es el estilo único con el que cada persona maneja los recursos del lenguaje para dar cuerpo a dicha idea. Dado que el estilo de escritor es el reflejo de su personalidad, la manera de expresarse de cada individuo lo distingue de otros escritores [9]. Actualmente, redes sociales como `Twitter` proporcionan espacio para que sus usuarios se expresen en no más de 280 caracteres. La naturaleza espontánea y creativa de cada persona hace del `tuit` un medio discursivo único y singular entre los espacios digitales de hoy en día.

De acuerdo con el estudio "`Digital in 2018`" [13], realizado por la firma `Hootsuite`, de los 130 millones de habitantes que existen en México, aproximadamente 85 millones son usuarios de Internet y el número de usuarios activos en redes sociales es de 83 millones. En el caso de `Twitter`, esta red social cuenta con más de 35 millones de usuarios en el país y el 29 % de ellos tiene entre 25 y 34 años de edad. Este rango de edad comprende la etapa más influyente y productiva de la llamada generación `millenial`.

Una de las principales características de este sector de la población supone el acceso a la información en tiempo real y una fuerte participación en la dinámica emprendedora que se vive en estos tiempos. Según `BBVA Research`, en su investigación "La paradoja de la generación del milenio" [10], estos jóvenes en su mayoría tienen una educación de grado superior. La búsqueda de nuevas oportunidades y la inversión económica-tecnológica

en ciertas ciudades del país ha favorecido al esparcimiento de talento mexicano. Estos movimientos de migración interna son estudiados por medio de encuestas por muestreo o registros administrativos de población, debido a que, actualmente en México, no se cuenta con un sistema de registro directo que capte los cambios de residencia en el momento en que se realizan.

Este tipo de migrantes mexicanos comparte ciertas características sociodemográficas y preferencias, entre las que destacan: la edad (migran más las personas jóvenes), actividad económica (activos y con importante participación en los sectores económicos secundario y terciario), localidad de destino (zonas urbanas) y nivel educativo (superan el nivel medio superior). La actividad ocupacional de la población que migra se encuentra estrechamente vinculada con la composición de los sectores de actividad económica. Por ello, no es extraño que se concentren básicamente en el sector servicios y en la industria manufacturera.

México es un país compuesto por 32 entidades federativas y una superficie de 1 964 375 km^2.Dentro de este basto terreno existen ciertas regiones geográficas que concentran importantes metrópolis las cuales poseen un desempeño económico, implementación tecnológica, e incluso un nivel sociocultural, con un tono contrastante al resto de las regiones del país.

Según Henríquez Ureña en su publicación "Las zonas dialectales de México", el territorio mexicano puede dividirse en 6 grandes zonas dialectales: (norte, centro, costa del Golfo de México, sur, región yucateca y Chiapas), siendo este uno de los primeros esfuerzos por delimitar la geografía lingüística del país [18]. Además de este, se han realizado otros proyectos con la intención de aportar un trazado de las zonas dialectales, fijando diferentes perspectivas y variando los parámetros en sus métodos de investigación. El Atlas lingüístico de México (ALM) es uno de los resultados de estas ambiciosas investigaciones. Por su parte Martín Butragueño en un esfuerzo más reciente, y teniendo como núcleo analítico el comportamiento de tres variables fonéticas en los datos del ALM, concluye que puede dividirse en 5 regiones al español mexicano: (centro-este, centro-oeste, noroeste, noreste y sureste) [1].

Las características del lenguaje y expresiones utilizadas por zona dialectal proporcionan una huella particular que ayuda a identificar su lugar de procedencia. Esta forma de manejar el lenguaje perdura al paso del tiempo, debido a que forma parte del proceso de habla desde sus inicios.

Por lo tanto, resulta normal que una persona continúe haciendo uso de su dialecto -incluso- después de haber emigrado a otro estado de la República Mexicana.

Este estilo de escritura acorde a zonas podría verse reflejado en tuits y, con la ayuda de técnicas de perfilado de autor, permitiría ubicar a cada usuario en su correspondiente zona dialéctica. De esta manera, analizar y clasificar a un usuario de `Twitter` por el texto contenido en su `timeline` en una de las diferentes zonas dialectales del país, podría representar una solución alternativa al estudio de fenómenos de migración interna. Teniendo en cuenta que estos movimientos poblacionales son una componente que interviene en la dinámica demográfica del país; además, contar con información oportuna sobre migración favorecería a la comprensión del impacto de este componente en los cambios y las tendencias de la población en tiempos recientes.

1.1. Planteamiento del problema

El fenómeno de migración interna en México está fuertemente vinculado a las condiciones de desarrollo regional. La situación económica actual que atraviesa el país actúa como atractor de talento para jóvenes que cuentan con formación específica en ciudades donde la inversión empresarial ha crecido. Sin embargo, no se cuenta con registros administrativos que permitan contabilizar los cambios de residencia o bien no son de fácil acceso para los investigadores.

Los métodos vigentes para este análisis son encuestas demográficas: la Encuesta Nacional de la Dinámica Demográfica (ENADID) y muestras censales de población, esfuerzos realizados por organizaciones como el INEGI, el Consejo Nacional de Población y la Secretaría de Salud. Estos aportes incluyen información que estima el volumen de los movimientos migratorios internos del país y las características de los migrantes. No obstante el tiempo que transcurre entre un estudio y otro, deja brechas en las que la información que resultaría de suma importancia no es aprovechada.

Desarrollar una herramienta web capaz de clasificar a un usuario de acuerdo a su zona dialectal y ocupación, representa una alternativa a los métodos tradicionales,y favorecería al análisis de movimientos migratorios internos.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar una herramienta automática que permita caracterizar el lenguaje español escrito de las distintas regiones geográficas de México por medio de técnicas de perfilado de autor.

1.2.2. Objetivos particulares

- Realizar un análisis de las características del lenguaje español escrito de México por regiones geográficas en redes sociales.
- Evaluar el desempeño de técnicas tradicionales de perfilado de autor en las tareas de identificación de zonas dialécticas e identificación de ocupación.
- Desarrollar un método para la creación de un modelo de clasificación de usuarios de `Twitter` acorde a su zona dialéctica y ocupación, e incorporarlo en una herramienta web.

1.3. Organización del documento

Después de una introducción al contexto sobre el que se plantea este Proyecto Terminal, se presenta el capítulo de Marco teórico en el cual se muestran las definiciones, conceptos y elementos teóricos que fundamentan el presente proyecto. Por ejemplo, aprendizaje automático, clasificación de textos, perfilado de autor, modelos de representación de texto, métricas de evaluación, etc.

El capítulo de Estado del arte tiene como principal propósito brindar al lector una parte del contexto actual en tareas de perfilado de autor, resaltando la importancia de una configuración adecuada a la tarea asignada.

A lo largo del capítulo de Método se explica el proceso de elaboración del modelo clasificador como un conjunto de tareas consecuentes, para terminar mostrando los resultados obtenidos durante la evaluación de los modelos experimentales y culminar con una discusión de los mismos.

El capítulo de Integración del sistema describe el diseño y desarrollo de la aplicación web, las herramientas que se utilizaron para su elaboración y una vista a la arquitectura del sistema.

Capítulo 2

Marco teórico

Este capítulo tiene como objetivo brindar al lector un panorama conceptual, apoyado de la explicación de elementos teóricos que fundamentan el presente proyecto.

2.1. Procesamiento de Lenguaje Natural

Actualmente grandes cantidades de datos son generados a cada segundo por medios de comunicación como las redes sociales. Este contenido textual además de tener una forma no estructurada, es representado en un lenguaje específico siguiendo cierta sintaxis y semántica, haciéndolo entendible por los humanos. En la actualidad, uno de los principales retos asociado a este tipo de contenido pretende analizar estos datos e intenta extraer patrones significativos y conocimiento útil.

El Procesamiento de Lenguaje Natural (PLN) se concibe como un campo especializado de ciencias de la computación, ingeniería e inteligencia artificial (IA) con raíces en la lingüística computacional; mismo que se ocupa principalmente del desarrollo de modelos computacionales que permiten la interacción entre máquinas y lenguajes naturales desarrollados para el uso humano [25].

Este procesamiento computacional del lenguaje incluye dos clases de algoritmos: los que toman el texto producido por el ser humano como entrada; y los que producen texto de aspecto natural como salida. Asimismo,

se incluyen tareas de análisis de texto cuyo propósito es extraer información y ésta tendría múltiples aplicaciones.

2.1.1. Perfilado de autor

La tarea de perfilado de autor tiene como objetivo distinguir, a partir de un texto, entre clases de autores. De esta forma, el perfilado de autor pretende modelar a grupos de autores por medio de atributos sociolingüísticos más generales. Dichos atributos son, además, indicadores de cómo los distintos grupos de autores emplean el lenguaje dependiendo de su género, edad o lenguaje nativo.

El perfilado de autor involucra la identificación y extracción de atributos textuales, la construcción de una representación adecuada del texto y la construcción de un modelo de clasificación, el cual es entrenado para posteriormente ser evaluado en la labor de identificación de perfiles de interés [2].

2.2. Clasificación de Textos

`Text analytics`, también conocido como minería de texto (`text mining`), es la metodología que se encarga de obtener conocimiento útil a partir de datos textuales. Esto involucra el uso del PLN, recuperación de información, y técnicas de `machine learning` para analizar estos datos de texto de forma no estructurada en maneras más estructuradas, y derivar patrones de estos datos, que por lo general tienen propósitos para un usuario final [25].

`Text Classification` o `Text Categorization` (TC) es una de las técnicas más importantes en minería de texto y tiene como objetivo relacionar documentos informales (como correos electrónicos, publicaciones, mensajes de texto, reseñas de algún producto, etc.) a una o varias categorías. Actualmente, el enfoque dominante para construir tales clasificadores de texto es el aprendizaje automático, que se encarga de aprender reglas de clasificación a partir de ejemplos. Es decir, un conjunto de documentos con sus categorías correspondientes [27].

Antes de aplicar cualquier técnica de aprendizaje o algoritmo a este conjunto de ejemplos, se deben convertir los datos no estructurados en

un formato aceptable por esos algoritmos. Existen diferentes formas de representar estos documentos de texto no estructurados en un formato matemáticamente computable.

A menudo, este texto debe limpiarse y procesarse para eliminar términos y datos que provoquen ruido, lo que se denomina preprocesamiento de texto [25].

Algunos ejemplos y problemas de clasificación de texto comunes son:

- Análisis de sentimiento: Proceso de comprensión sobre si un texto habla positiva o negativamente de un tema.
- Detección de temas: Proceso de identificación del tema de un texto.
- Detección de lenguaje: Proceso para detectar el idioma de un texto dado.

2.2.1. Representaciones de vectoriales de textos

El hecho de que las computadoras sólo puedan manipular representaciones numéricas implica que tratar computacionalmente un lenguaje natural deba requerir un proceso de modelado matemático.

La representación de textos es uno de los problemas fundamentales en la minería de texto y la recuperación de información. Su objetivo es representar numéricamente los documentos de texto no estructurados para hacerlos matemáticamente computables. Para un conjunto dado de documentos de texto $D = \{d_i, i = 1, 2, 3, ..., n\}$, donde cada d_i representa un documento, el problema de la representación de textos es representar cada d_i de D como un punto s_i en un espacio numérico S, donde la distancia/similitud entre cada par de puntos en el espacio S está bien definida [32].

Bolsa de Palabras

Una de las formas para representar texto más efectiva y popular en actividades de recuperación de información es la representación de Bolsa de Palabras o `Bag-of-Words` o (BoW). Cuando usamos esta representación, descartamos la estructura del texto de entrada, como capítulos, párrafos, oraciones, y el formato. Solamente contamos cuántas apariciones tiene cada palabra en cada texto del corpus. Descartar la estructura del texto y solamente contar las ocurrencias de las palabras da la imagen mental de representar el texto como una "bolsa".

El procesamiento de una BoW para un corpus de documentos tiene el siguiente procedimiento:

- 1. Tokenización: Divide cada documento en las palabras que aparecen en él (llamadas tokens).
- 2. Construcción del vocabulario: Arma un vocabulario con todas las palabras que aparecen en cualquiera de los documentos, y las enumera.
- 3. Codificar: Para cada documento, cuenta con qué frecuencia aparece cada una de las palabras aparece en ese documento [11].

Word embeddings

La idea en la que se basa `word embeddings` es que cada palabra se puede convertir en un vector de N dimensiones. Donde a cada palabra se le asigna un vector único, y las palabras similares poseen valores cercanos entre sí.

Los `word embeddings` son una forma de representar palabras como un vector de números reales, donde cada valor captura una dimensión del significado de la palabra. Como resultado, palabras semánticamente similares tienen vectores similares.

Para que estas representaciones sean realmente útiles, la meta es capturar significados, relaciones semánticas y sintácticas, similitud entre palabras, y el contexto de las palabras [17].

2.3. Aprendizaje Automático

El Aprendizaje Automático o `Machine Learning` (ML) es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de datos, identificar patrones y tomar decisiones con mínima intervención humana. De forma más concreta, el ML trata de crear programas capaces de generalizar patrones a partir de información no estructurada suministrada en forma de ejemplos [8].

Los algoritmos del ML se clasifican como supervisados o no supervisados. Los algoritmos supervisados pueden aplicar lo que se ha aprendido en el pasado a nuevos datos. Los algoritmos no supervisados pueden extraer inferencias de conjuntos de datos [31].

2.3.1. Aprendizaje supervisado

El aprendizaje supervisado comprende un conjunto de variables de entrada (x) y una variable de salida (Y), aplica un algoritmo para aprender la función de mapeo de la entrada a la salida; dicha entrada está compuesta por un conjunto de datos etiquetados.

$$Y = f(x) \tag{2.1}$$

El objetivo es aproximar la función de mapeo lo mejor posible de manera que cuando tenga nuevos datos de entrada (x) pueda predecir las variables de salida (Y) para esos datos. Es decir, se crea una función (o modelo) mediante el uso de datos etiquetados de entrenamiento que actúan como datos de entrada, dicha función es capaz de asignar etiquetas de clase a un nuevo grupo de datos.

Los problemas de aprendizaje supervisado suelen ser agrupados en dos clases: problemas de regresión y problemas de clasificación. Los problemas de regresión involucran una salida de valor real, como `peso` o `rendimiento`. Mientras que los problemas de clasificación involucran como salida una categoría, como `zona` u `ocupación` [31].

Algunos ejemplos de algoritmos de aprendizaje supervisado son:

- Regresión lineal para problemas de regresión.
- Bosques aleatorios para problemas de clasificación y regresión.
- Máquinas Soporte Vectorial para problemas de clasificación.

Los algoritmos de clasificación son algoritmos de aprendizaje supervisado que se encargan de clasificar, categorizar, o etiquetar conjuntos de datos en función de experiencia previa. Al ser algoritmos de aprendizaje supervisado requieren datos de entrenamiento. Estos datos de entrenamiento consisten en un conjunto de observaciones de entrenamiento donde cada observación es una dupla que está compuesta por una instancia de entrada, generalmente un vector de características, y un resultado de salida asociado a dicha instancia [25].

Esencialmente son tres los procesos por los que atraviesa los algoritmos de clasificación:

- Entrenamiento: Es el proceso en el que el algoritmo de aprendizaje supervisado intenta inferir patrones a partir del conjunto de entrenamiento, de manera que pueda identificar qué patrones conducen a un resultado específico, estos resultados son las etiquetas de clase. El proceso de extracción de características se lleva a cabo antes del entrenamiento. El conjunto de características alimenta el algoritmo elegido, el cual trata de identificar y aprender patrones además de sus resultados correspondientes. Este proceso tiene como resultado un modelo de clasificación. El cual se espera esté lo suficientemente generalizado de modo que pueda predecir las etiquetas para nuevos ejemplos.
- Evaluación: En este proceso se obtiene el desempeño del modelo de clasificación para ver que tan bien aprendió en el proceso anterior. Para esto, usualmente se utiliza un conjunto de validación y se prueba el desempeño del modelo prediciendo sobre este conjunto y comparando nuestras predicciones contra las etiquetas reales `(ground truth)`.

- Optimización: Este proceso tiene el objetivo de optimizar el modelo para maximizar su capacidad de predicción y reducir los errores. Cada modelo es en realidad una función matemática compuesta por varios parámetros que determinan, la complejidad del modelo, capacidad de aprendizaje, entre otros. Estos son conocidos como `hiperparámetros` y son definidos previo al entrenamiento del modelo.

Algunos de los algoritmos de aprendizaje supervisado más aplicados a clasificación son los siguientes: clasificadores lineales, regresión logística, Clasificador Naïve Bayes, Máquinas de Soporte Vectorial, Árboles de decisión, etc.

- Los modelos lineales para la clasificación separan vectores de entrada en clases usando límites de decisión lineales (hiperplano). El objetivo de la clasificación en clasificadores lineales en aprendizaje automático, es agrupar elementos que tienen valores de características similares, en grupos.

- Un Clasificador Naïve Bayes asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable. Estos clasificadores consideran que cada una de las características contribuye de manera independiente a la probabilidad de un hecho, además de estar fundamentados en el teorema de Bayes.

- Máquinas de Soporte Vectorial son un reciente conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios ATT. Una Máquina de Soporte Vectorial construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta [3].

2.3.2. Evaluación del modelo de clasificación

El desempeño de los modelos de clasificación generalmente se basa en qué tan bien predicen los resultados para nuevos conjuntos de datos. Por lo general, este rendimiento se mide contra un conjunto de datos de prueba

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 2.1: Estructura de la matriz de confusión.

que consiste en un conjunto de datos que no se usaron para influir o entrenar el modelo de clasificación.

Este nuevo corpus se representa de la misma manera que se representó el texto para entrenar el modelo. Se aplica el modelo clasificador al conjunto de prueba y se recopilan las predicciones. Estas predicciones se comparan con las etiquetas reales para ver con qué precisión pronosticó el modelo [25].

Una de las herramientas fundamentales en el proceso de evaluación del desempeño de un algoritmo de clasificación es la matriz de confusión (figura 2.1), la cual muestra un conteo de los aciertos y errores de cada una de las clases predichas [29].

- VP es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.
- VN es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.
- FN es la cantidad de positivos que fueron clasificados incorrectamente como negativos.
- FP es la cantidad de negativos que fueron clasificados incorrectamente como positivos.

Algunas métricas de evaluación que se pueden obtener a partir de la matriz de confusión y así que determinar el rendimiento de un modelo, son:

- Exactitud: Es la relación entre el número de predicciones correctas y el número total de muestras de entrada.

$$Exactitud = \frac{VP + VN}{Total}$$

- Precisión: Es intuitivamente la habilidad del clasificador para no etiquetar como positiva una muestra que es negativa.

$$Precisión = \frac{VP}{VP + FP}$$

- Recuerdo: Es intuitivamente la habilidad del clasificador para encontrar todas las muestras positivas.

$$Recuerdo = \frac{VP}{VP + FN}$$

- Puntaje F1: Es el promedio ponderado entre Precisión y Recuerdo. Esta puntuación es útil si se tiene una distribución de clases desigual.

$$PuntuaciónF1 = 2 * \frac{Precisión * Recuerdo}{Precisión + Recuerdo}$$

Los métodos de aprendizaje automático son técnicas principalmente estadísticas que por lo general reciben como entrada vectores con valores reales. Considerando que la palabra es la unidad básica del lenguaje natural [19], existe un gran interés en generar métodos eficientes para la construcción de representaciones vectoriales para las palabras del lenguaje.

Puesto que el desempeño de la mayoría de los algoritmos de aprendizaje supervisado va ligado a la calidad de estas representaciones, es importante que el vector asociado a la palabra mantenga cierta estructura semántica y sintáctica, dentro de sus valores.

Capítulo 3

Estado del arte

La comunicación por medios electrónicos, y en particular las redes sociales, cobran especial importancia en la vida cotidiana de todos. Es por esto, que el análisis de textos provenientes de redes sociales se ha convertido en un tema de investigación popular entre la comunidad de lingüística computacional.

Casos de redes sociales como `Twitter`, se encuentran en constante crecimiento debido a la información generada por una de las más grandes comunidades de usuarios activos. Analizar información proveniente de este tipo de plataformas se ha convertido en una tarea de suma importancia en áreas como seguridad, marketing, ciencias forenses, entre otras.

Perfilado de autor es una de las principales tareas de clasificación a partir de texto aplicada a contenido de redes sociales. La clasificación de autores basada en el estilo de sus textos tiene aplicación en diversas áreas de conocimiento: la psicología, la lingüística, el PLN, etc. Además, analizar el estilo de los textos permite extraer características demográficas de un individuo: género, edad, lengua materna, personalidad, religión, clase socio-económica, lugar de residencia, profesión, etc[22].

Aplicación web para identificar personalidad, género y edad de usuarios en Twitter [12], es uno de los trabajos que busca identificar el perfil de un usuario mediante el análisis de sus tuits. Este tiene como objetivo validar la información del perfil por un cuestionario de personalidad para expandir el conjunto de datos etiquetados utilizado en el proyecto.

En dicho trabajo se utilizaron los datos del PAN-2015 [1] y el enfoque de representación basado en grafos. Con esto se planeaba identificar la personalidad, el género y la edad de usuarios en `Twitter`. Cabe remarcar que la representación basada en grafos para entrenar modelos de clasificación, no había sido empleada en clasificación no temática de textos. Por lo tanto, basaron la evaluación de dicha representación en la tarea de perfilado de autor.

A su vez, en [6] se presenta un algoritmo que combina las características representadas por los n-gramas de caracteres y los n-gramas de etiquetas gramaticales (POS) para clasificar documentos multilengua de redes sociales. El algoritmo se aplicó sobre los siguientes corpus: `Author Profiling` de PAN-CLEF 2015 y Comentarios de la Ciudad de México en el tiempo (CCDMX). En este estudio se mejoró el rendimiento del sistema de clasificación a través de una normalización dinámica dependiente del contexto, porque les permitió extraer la mayor cantidad de información estilística codificada en los documentos.

3.1. MEX-A3T

Existen competencias académicas que tienen como propósito avanzar en el estado del arte para este tipo de tareas de clasificación. PAN lab [2] y RepLab [3] en `Conference and Labs of the Evaluation` (CLEF) [4] son algunos de los forums de evaluación más conocidos para la tarea de perfilado de autor, han tenido varias ediciones donde continuamente los participantes mejoran los métodos de clasificación.

MEX-A3T [5] es una competencia académica que forma parte de IberLEF 2019 [6] y tiene como objetivo fomentar la investigación sobre el análisis de contenido de redes sociales en español mexicano. MEX-A3T considera dos tareas clave, perfilado de autor y detección de agresividad, ambas utilizando

[1] https://pan.webis.de/clef15/pan15-web/
[2] https://pan.webis.de/
[3] http://www.clef-initiative.eu/track/replab
[4] http://www.clef-initiative.eu/
[5] https://sites.google.com/view/mex-a3t/
[6] https://sites.google.com/view/iberlef-2019/

tuits mexicanos en español, esto implica tratar con una variedad del español que tiene rasgos culturales que lo hacen significativamente diferente del español peninsular.

La tarea de perfilado de autor tiene como objetivo identificar dos dimensiones no estándar: lugar de residencia y ocupación de usuarios de `Twitter`. Como insumo de trabajo se construyó un corpus, el cual está compuesto por 5 mil `timelines` de usuarios mexicanos, y contiene las etiquetas correspondientes para poder realizar la clasificación (lugar de residencia y ocupación).

3.1.1. Resumen de los enfoques presentados

A continuación se presenta un resumen de los métodos utilizados por los equipos participantes del MEX-A3T, en la tarea de perfilado de autor en términos de preprocesamiento, representación y algoritmos de clasificación.

- CIC GIL `Approach to Author Profiling in Spanish Tweets: Location and Occupation` [14]

- Tarea: Perfilado de autor.
- Nombre del equipo: CIC-GIL
- Preprocesamiento: Todas las letras convertidas a minúsculas, normalización de números, menciones, `hashtags`, ligas de imágenes y `urls`; remplazamiento de jerga mexicana por su versión estándar.
- Representación: n-gramas de caracteres, n-gramas de palabras y regionalismos para la tarea de lugar de residencia.
- Clasificación: Algoritmo de Regresión Logística (además de SVM y Bayes).
- F1-macro score: 0.7363 para la tarea de lugar de residencia y 0.4089 para la tarea de ocupación.

-INGEOTEC at `MEX-A3T: Author profiling and aggresiveness analysis in Twitter using micro-TC and EvoMSA` [20]

- Tarea: Perfilado de autor.
- Nombre del equipo: INGEOTEC
- Preprocesamiento: Stemming.
- Representación: n-gramas de caracteres, n-gramas de palabras, `skip-grams`, con pesado tf y tdidf.
- Clasificación: SVC con un kernel lineal.
- F1-macro score: 0.8155 para la tarea de lugar de residencia y 0.4470 para la tarea de ocupación.

- `Author Profiling Aggressiveness Detection in Spanish Tweets: MEX-A3T 2018` [4]

- Tarea: Perfilado de autor.
- Nombre del equipo: Aragon-Lopez
- Representación: Bolsa de Términos, Atributos de Segundo Orden, n-gramas de caracteres y palabras.
- Clasificación: Modelos CNN como CNN-Rand, CNN-Static y CNN-NonStatic.
- F1-macro score: 0.8388 para la tarea de lugar de residencia y 0.4910 para la tarea de ocupación.

- `The Winning Approach for Author Profiling of Mexican Users in Twitter at MEX.A3T@IBEREVAL-2018` [24]

- Tarea: Perfilado de autor.
- Nombre del equipo: MXAA
- Representación: Los autores utilizaron una técnica llamada pureza personal discriminatoria o `(DPP)` por sus siglas en inglés.

- Clasificación: Máquina de Soporte Vectorial `SVM` con normalización L2.
- F1-macro score: 0.8301 para la tarea de lugar de residencia y 0.5122 para la tarea de ocupación.

A continuación se presenta en el cuadro 3.1 una tabla comparativa de los enfoques participantes en la tarea de perfilado de autor.

Enfoque	**Preprocesamiento**	**Representación**	**Clasificación**
CIC-GIL	Minúsculas Normalización Reemplazamiento de jerga	N-gramas de caracteres N-gramas de palabras	Algoritmo de regresión logística SVM Bayes
INGEOTEC	Stemming	N-gramas de caracteres N-gramas de palabras Skip-grams	SVC
Aragon-Lopez		Bolsa de términos Atributos de segundo orden N-gramas de caracteres N-gramas de palabras	Modelos CNN
MXAA		Pureza Personal Discriminatoria	SVM

Cuadro 3.1: Resumen de los enfoques.

3.1.2. Análisis de los resultados

Para obtener los puntajes de evaluación los autores se apoyaron de la plataforma `EVALL`, el cual es un servicio de evaluación para tareas de recuperación de información y tareas de PLN. Es un `framework` de evaluación que recibe como entrada las instancias con su clasificación y devuelve una evaluación del rendimiento completa. Los resultados de los participantes son los siguientes:

Como `baseline` se implementaron dos de las aproximaciones más populares, considerados duros de vencer para ambas tareas: i) un modelo

de clasificación entrenada con la representación de `BoW` y otro clasificador entrenado sobre una representación de 3-gramas de caracteres (Trigramas).

La aproximación del equipo Aragon-Lopez obtuvo el mejor desempeño para la tarea de lugar de residencia, mientras que el método del equipo MXAA obtuvo el mejor rendimiento en la tarea de ocupación. En general el rendimiento del equipo MXAA fue calificado como el mejor.

Equipo	Ocupación	Lugar de residencia	F promedio
MXAA	0.5122	0.8301	0.6711
Aragon-Lopez (intento 1)	0.4910	0.8388	0.6649
INGEOTEC	0.4470	0.8155	0.6312
CIC-GIL (intento 2)	0.4894	0.7363	0.2168
CIC-GIL (intento 1)	0.4727	0.7310	0.6018
`BoW`	0.47675	0.6295	0.5531
`Trigramas`	0.41875	0.6004	0.5095
Aragon-Lopez (intento 2)	0.3824	06.619	0.5007

Cuadro 3.2: Rendimiento de la medida F macro para ambos rasgos.

Los métodos utilizados por los participantes de la competencia MEX-A3T, consultados y los citados proporcionaron diversas formas de trabajar con textos no formales. En el caso particular de los tuits, las reglas y características presentes en su naturaleza, los hacen especialmente manipulables. Entonces, a partir de las nociones propuestas previamente se definió un método de trabajo para este contenido, considerando ciertas operaciones de preprocesamiento y como representación del texto: bolsa de palabras y `word embeddings`.

Capítulo 4

Método

Este capítulo incluye una descripción del método empleado para la construcción del modelo, detallando los experimentos realizados y concluyendo con una discusión de los resultados obtenidos. Las representaciones del texto así como los algoritmos de aprendizaje supervisado fueron elegidos debido a que son recursos comúnmente mencionados en el ámbito del `Text Classification`. Entonces, las decisiones de diseño presentes en el método son además un `baseline` para futuras réplicas del método.

Con la intención de tener una mejor estructura en el proyecto, el proyecto está dividido en dos grandes bloques de contenido: Bloque 1: Experimentación y optimización del modelo, y Bloque 2: Desarrollo e implementación del sistema.

De acuerdo al esquema de aprendizaje supervisado (figura 4.1), se irán revisando los elementos involucrados en cada uno de los procesos que componen las etapas relacionadas a cada bloque.

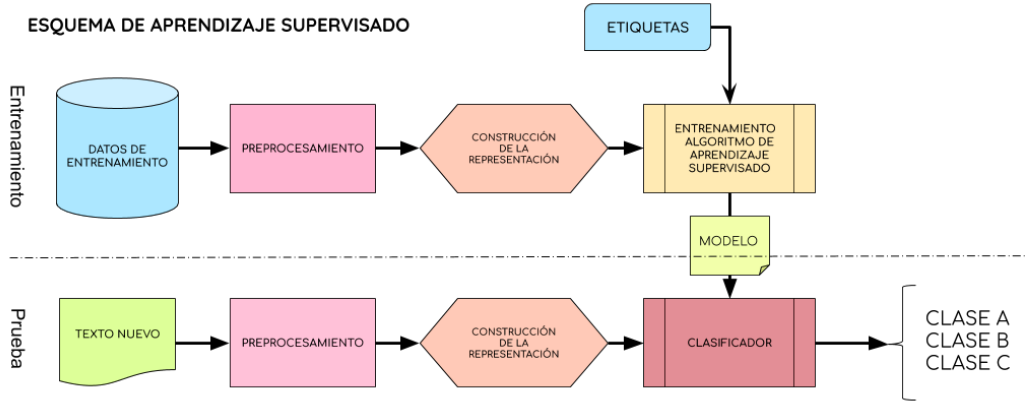

Figura 4.1: Esquema de aprendizaje supervisado

4.1. Bloque 1. Experimentación y optimización del modelo

El insumo principal sobre el que se trabaja en este proyecto es un corpus generado por un equipo de trabajo parte de la competencia MEX-A3T, el cual consiste de 5 mil perfiles de usuarios de `Twitter` mexicanos. Estos perfiles fueron descargados aleatoriamente en el periodo de Junio a Noviembre del 2016.

Cada perfil es etiquetado con información sobre la ocupación y lugar de residencia del usuario. Para la etiqueta de ocupación, se consideraron las siguientes 8 clases: Arts, Student, Social, Sciences, Sports, Administrative, Healt, y Others. Mientras que para el lugar de residencia, se consideraron las siguientes 6 clases: North, Northwest, Northeast, Center, West, y Southeast.

4.1.1. Estadísticas del corpus

El corpus del MEX-A3T está seccionado en un apartado para el entrenamiento del modelo y otro apartado para la validación de este modelo (figura 4.2). Como se puede notar en el cuadro 4.1 la clase mayoritaria corresponde a la región `Center`, con más del 36 % de los perfiles, mientras que la clase minoritaria es la región `North`, con el 3 % de las instancias. Por otro lado, en el cuadro 4.2, se muestra la distribución del rasgo de ocupación. Donde la clase mayoritaria es `Students` con casi el 50 % de los perfiles, mientras que `Sports` es la clase minoritaria, con aproximadamente 1 % de los las instancias.

En el cuadro 4.3 se presentan algunas estadísticas adicionales del corpus para la tarea de perfilado de autor. Para esto se consideraron números, signos de puntuación y emojis como términos. Además se normalizaron menciones, `hashtags` y `urls`.

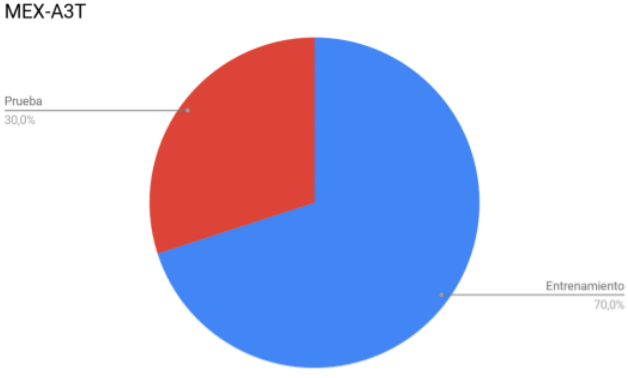

Figura 4.2: Particiones del corpus.

Clase	Corpus de entrenamiento(%)	Corpus de prueba(%)
North	106 (3.02)	34 (2.26)
Northwest	576 (16.45)	229 (15.26)
Northeast	914 (26.11)	389 (25.93)
Center	1266 (36.17)	554 (36.93)
West	322 (9.20)	144 (9.60)
Southeast	316 (9.02)	150 (10.0)
$\sum$	3500	1500

Cuadro 4.1: Corpus mexicano para perfilado de autor: distribución del rasgo de lugar de residencia.

Clase	Corpus de entrenamiento(%)	Corpus de prueba (%)
Arts	240 (6.85)	103 (6.86)
Student	1648 (47.08)	740 (49.33)
Social	570 (16.28)	234 (15.60)
Sciences	185 (5.28)	65 (4.33)
Sports	45 (1.28)	26 (1.73)
Administrative	632 (18.05)	264 (17.60)
Health	105 (3.00)	43 (2.86)
Others	75 (2.14)	25 (1.66)
$\sum$	3500	1500

Cuadro 4.2: Corpus mexicano para perfilado de autor: distribución del rasgo de ocupación.

Medida	Corpus de entrenamiento	Corpus de prueba	Corpus completo
Tuits por perfil	1354.21	1353.38	1353.96
Número de términos	78,542,124	34,032,819	112,574,943
Tamaño del vocabulario	2,540,580	1,274,902	3,506,826

Cuadro 4.3: Estadísticas del Corpus mexicano para perfilado de autor.

En este Bloque 1, se describirá la sección de entrenamiento y prueba del esquema de aprendizaje supervisado (cuadro 4.1), donde el objetivo radica

en construir un modelo clasificador basado en el corpus antes descrito, este debe ser capaz de realizar una clasificación para nuevas instancias (`timelines`), al final del bloque se reportan los resultados obtenidos para las diferentes configuraciones utilizadas.

4.2. Etapa 1: Identificación y extracción de atributos

La primer tarea del bloque consiste en conocer el corpus, los `timeline` de `Twitter` contenidos en este corpus deben almacenarse en una estructura de datos que permita la manipulación de su contenido. El corpus tiene la siguiente estructura: un archivo de texto por cada usuario (3500 para entrenamiento y 1500 para prueba) y un archivo `ground truth`.

Se depositó el texto y sus respectivas etiquetas en dos archivos (entrenamiento y prueba) CSV para un manejo ordenado del corpus.

4.2.1. Etapa 1.1: Preprocesamiento

Una vez teniendo una estructura sobre la cual trabajar, se puede aplicar preprocesamiento a los datos, esto consiste en homogeneizar los datos dentro del corpus. Tareas de limpieza, tokenización y análisis de tuits son posibles gracias a bibliotecas enfocadas al manejo de contenido de `Twitter`. En el preprocesamiento se unificaron diversas etiquetas por categorías (`URLs`, `hashtags`, menciones, palabras reservadas (`RT, FAV`), `emojis`), además se convirtió todo el texto del `timeline` a minúscula, y removieron signos de puntuación.

En caso de que exista algún `timeline` vacío este se elimina del corpus durante el preprocesamiento, por lo que el corpus de entrenamiento y prueba cuentan con 3470 y 1493 instancias respectivamente.

4.2.2. Etapa 1.2: Representación de los documentos

El texto de los corpus tomó dos representaciones: bolsa de palabras (BoW) y `word embeddings`. Se realizaron experimentos con ambas representaciones con el fin de compararlas en función del desempeño de tres de los algoritmos de clasificación más utilizados en la clasificación de textos.

Bolsa de Palabras

El modelo de bolsa de palabras es una forma de extraer características del texto para usarlo en el modelado, esta representación del texto describe la ocurrencia de palabras dentro de un documento. Como se ha mencionado, cualquier información sobre el orden o estructura de las palabras queda descartada en este modelo.

En este proyecto la implementación de la representación de bolsa de palabras fue por medio de las funciones: `CountVectorizer y TfidfVectorizer`, ambas parte del conjunto de herramientas para `machine learning Scikit-learn` [1]. Las cuales a grandes rasgos, convierten el texto del corpus en una matriz término-documento.

Word Embeddings

Como se mencionó previamente, existe otro gran enfoque en la construcción de representaciones vectoriales basado en técnicas de modelado del lenguaje mediante redes neuronales, en el cual las palabras o frases del vocabulario son vinculadas a vectores de números reales bajo ciertas relaciones, por ejemplo semánticas. Se ha demostrado que el uso de éstos vectores de palabras aumenta el rendimiento de tareas relacionadas al procesamiento del lenguaje natural.

`FastText` [2] es un método dedicado al cálculo de vectores de palabras, disponible como biblioteca enfocada a tareas de aprendizaje y clasificación de textos, desarrollada por `Facebook AI Research` [16].

La competencia MEX-A3T liberó como recurso `FastText Word Embeddings for Spanish Language Variations`, modelo entrenado con tuits mexicanos y compuesto por 1,247.3 M de tokens, de 100 dimensiones cada uno. En este modelo se consultan las palabras que componen el `timeline` del usuario, dado que cada palabra es un vector de dimensión 100, el vector representativo del documento es el resultado de la suma de todos los vectores de las palabras del texto y dividido por el número de palabras. Un arreglo de vectores es la representación del corpus, donde cada fila (vector) representa un `timeline` entero.

[1] https://scikit-learn.org/stable/
[2] https://fasttext.cc/

4.3. Etapa 2. Entrenamiento del modelo de clasificación

De acuerdo al esquema de aprendizaje supervisado (figura 4.1), la tarea de entrenamiento del modelo clasificador implica la elección de un algoritmo de aprendizaje supervisado, el cual es entrenado con un corpus etiquetado. Este modelo debe ser capaz de realizar una clasificación para nuevas instancias, en este caso nuevos `timelines` de usuarios.

Los algoritmos de clasificación implementados en el proyecto contemplan algunos de los tipos de clasificadores más comunes en el campo del `machine learning`. El funcionamiento de estos algoritmos queda descrito de la siguiente forma:

- Se espera que las `Support Vector Machines` separen el conjunto de datos de entrada de la mejor manera posible. La distancia entre los puntos más cercanos se conoce como el margen. La idea central de SVM es encontrar un hiperplano marginal máximo (MMH) que divida mejor el conjunto de datos en clases.
- El caso de un clasificador Naive Bayes supone que la presencia de una característica particular en una clase no está relacionada con la presencia de ninguna otra característica. Incluso si estas dependen unas de otras o de la existencia de otras características, todas estas propiedades contribuyen independientemente a la probabilidad.
- `Decision Tree` crea modelos de clasificación en una estructura arbórea, compuesta por nodos de decisión y nodos hoja. Un nodo de decisión tiene dos o más ramas y un nodo hoja representa una clasificación o decisión.

Algoritmos	**Implementación**
C-Support Vector Classification	SVC ( )
Naive Bayes classifier for multinomial models	MultinomialNB( )
Decision Trees	DecisionTreeClassifier( )

Cuadro 4.4: Algoritmos de aprendizaje supervisado parte de Scikit-Learn.

4.3.1. Configuración de los modelos

A continuación se presentan las configuraciones con las que se generó cada modelo para la tarea de lugar de residencia y ocupación:

Bolsa de Palabras

Se utiliza este modelo de representación con la intención de verificar si la ocurrencia de las palabras es una característica que resulta determinante en la clasificación de usuarios por lugar de residencia y ocupación, intuyendo que documentos similares tienen un contenido similar.

Los modelos de clasificación tuvieron en cuenta solamente la presencia de los términos en los documentos. De esta manera los pesos son 1 o 0, dependiendo de la aparición o no del termino en el texto. Además existen otros tipos de pesado, como TF-IDF que mide con qué frecuencia aparece un término dentro de un documento determinado, y lo compara con el número de documentos que mencionan ese término dentro de una colección entera de documentos.

Identificador	Implementación	Algoritmo clasificador	Pesado
BoW_1	CountVectorizer	SVC	Binario
BoW_2	CountVectorizer	DT	Binario
BoW_3	CountVectorizer	NB	Binario
BoW_4	TdidfVectorizer	SVC	Binario
BoW_5	TdidfVectorizer	DT	Binario
BoW_6	TdidfVectorizer	NB	Binario

Cuadro 4.5: Configuraciones del modelo basado en representación bolsa de palabras. (Cada modelo se aplicó a ambas tareas de clasificación.)

Word Embeddings

En el caso de esta representación vectorial se piensa que palabras similares o relacionadas mantienen vectores similares y siendo este el caso, un vector podría contener la estructura de un `timeline` entre sus dimensiones.

Se ha definido un par de algoritmos (`Support Vector Classifier y Decision Tree`) para ser evaluados bajo la misma representación.

Identificador	Representación	Algoritmo clasificador
WE_1	`word embeddings`	SVC
WE_2	`word embeddings`	DT

Cuadro 4.6: Configuraciones del modelo basado en representación de `word embeddings`. (Cada modelo se aplicó a ambas tareas de clasificación.)

4.4. Etapa 3. Validación y optimización

La tarea final del Bloque 1 consiste en evaluar los modelos de clasificación generados en la etapa anterior. Dicha evaluación se aplica sobre un corpus de prueba que consta de un corpus de 1493 documentos, las métricas reportadas son exactitud y puntaje F1, mismas que sirven como referencia en comparación de los resultados obtenidos por los equipos participantes de la competencia MEX-A3T.

4.4.1. Tarea: Lugar de residencia

ID	Exactitud	Puntaje F1 (micro)	Puntaje F1 (macro)
BoW_1	0.747	0.747	0.613
BoW_2	0.609	0.609	0.489
BoW_3	0.546	0.546	–
BoW_4	0.711	0.711	0.439
BoW_5	0.563	0.563	0.459
BoW_6	0.486	0.486	0.193
WE_1	0.512	0.512	0.210
WE_2	0.401	0.401	0.282

Cuadro 4.7: Resultados para la tarea de lugar de residencia.

4.4.2. Tarea: Ocupación

ID	Exactitud	Puntaje F1 (micro)	Puntaje F1 (macro)
BoW_1	0.731	0.731	0.412
BoW_2	0.577	0.577	0.279
BoW_3	0.668	0.668	0.243
BoW_4	0.712	0.712	0.287
BoW_5	0.557	0.557	0.268
BoW_6	0.517	0.517	–
WE_1	0.691	0.691	0.210
WE_2	0.585	0.585	0.281

Cuadro 4.8: Resultados para la tarea de ocupación.

Como resumen de los experimentos, podemos notar que el algoritmo de aprendizaje supervisado `Support Vector Classifier` y la representación por bolsa de palabras, poseen el mejor desempeño durante el proceso de prueba. Dicho algoritmo conserva la ventaja en ambas representaciones.

Tal parece que comprobar la frecuencia de un término actúa mejor que una concepción semántica del texto, en términos de representación. También se comprobó la eficiencia de `Support Vector Machines` en tareas de clasificación.

Comparación con Estado del Arte

En el cuadro 4.9 se presentan primeramente los mejores resultados de la competencia MEX-A3T, seguido de ellos el mejor resultado obtenido durante la experimentación.

ID	Ocupación (F-macro)	Lugar de residencia (F-macro)	F promedio (F-macro)
MXAA	0.512	0.830	0.671
Aragon-Lopez (intento 1)	0.491	0.838	0.664
BoW (Baseline)	0.477	0.629	0.533
Trigramas (Baseline)	0.418	0.600	0.509
BoW_1	0.412	0.613	0.512

Cuadro 4.9: Comparación de resultados con Estado del Arte

Como se puede notar la propuesta BoW1 tiene un menor rendimiento en contraste con los métodos desarrollados durante MEX-A3T, superando únicamente al `baseline` de Trigramas. Los equipos MXXA y Aragon-Lopez obtuvieron los mejores resultados.

Algoritmos de clasificación como los Modelos CNN y representación de textos como la técnica de Pureza Personal Discriminatoria definieron la diferencia en términos de rendimiento, pues demostraron que estas configuraciones alternativas funcionaron.

Es importante mencionar que el proyecto no está enfocado en vencer los resultados descritos en el Estado del arte, sino en generar un método base. Durante el diseño del método se implementaron diferentes recursos con la intención de tener un `baseline` lo más destacado posible, eligiendo entre dos formas de representación de texto y al menos dos algoritmos de clasificación.

Capítulo 5

Implementación del sistema

Recapitulando, el modelo obtenido en Bloque 1 tiene la capacidad de clasificar nuevos `timelines` de usuarios de Twitter y fue generado a partir de un proceso de perfilado de autor. Después de seleccionar el modelo con mejor desempeño, el siguiente paso consiste en incorporar dicho modelo en una herramienta web.

MIMTA nace como el resultado del método antes descrito, siendo una herramienta web capaz de clasificar a un usuario de `Twitter` acorde a su zona dialéctica y ocupación.

El modelo clasificador de MIMTA está basado en la representación de texto de Bolsa de Palabras, SVC como algoritmo clasificador y un pesado binario. Se desarrollaron diferentes funciones con el propósito de formar una herramienta que aporte al estudio de la migración interna mexicana, obteniendo información del perfil, contenido del `timeline` y la probabilidad por clase para las tareas de lugar de residencia y ocupación.

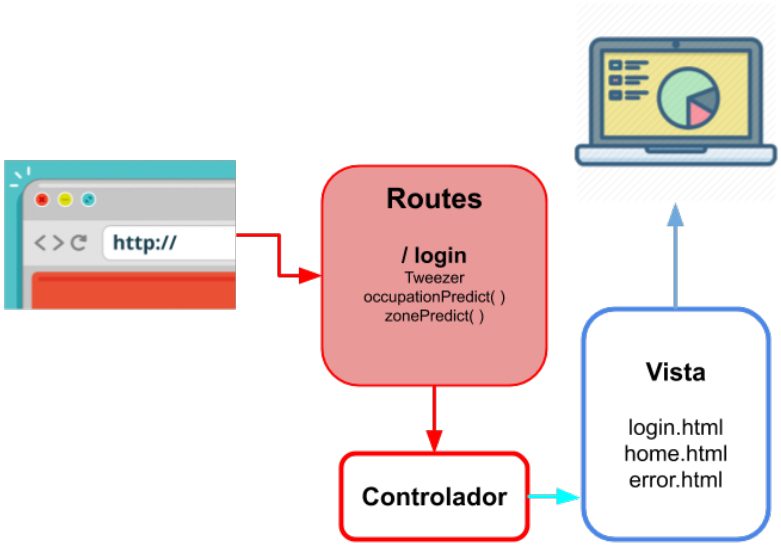

Figura 5.1: Arquitectura MVC de MIMTA.

En el patrón de diseño Modelo Vista Controlador (MVC) un usuario hace una petición para ver una página accediendo a un `URL`, el Controlador recibe la petición, usa al Modelo para recuperar todos los datos necesarios, organizarlos y enviarlos a la Vista, la cual usa esos datos para renderizar una página web presentada al usuario a través del navegador.

Para el caso de MIMTA, el Controlador administra sólo una vista y esta despliega al módulo `Tweezer`, que es el encargado de recopilar la información necesaria, utilizando un `username` como entrada; además de aplicar preprocesamiento al `timeline` y depurar la información sobre los tuits. También invoca a las funciones de predicción para las tareas de lugar de residencia y ocupación, para finalmente renderizar un `HTML` con el contenido del tablero. En la figura 5.1 se muestra la arquitectura de MIMTA, definiendo la ubicación de cada módulo que compone la herramienta.

5.1. Bloque 2. Desarrollo e implementación de la herramienta web

Una de las opciones de acceso a información relacionada con movimientos migratorios internos en México es la proporcionada por la CONAPO, en su sitio web 'Datos Abiertos' [1], donde se pueden descargar archivos CSV con datos de diferente índole social. El sistema descrito en el presente documento, genera datos individuales de este tipo y aplicados a un gran conjunto de personas podría crear recursos digitales igualmente útiles.

5.1.1. Etapa 4. Incorporación del modelo en una aplicación

La Etapa 4 describe el entorno en el que se deposita el modelo clasificador, `frameworks` e instancias de bibliotecas componen la herramienta web MIMTA. Cada módulo desarrollado interactúa con diferentes elementos propios y de terceros, en esta etapa se diseñan e implementan estas relaciones.

Etapa 4.1 Herramientas de desarrollo

El sistema tiene una estructura bastante simple por lo que no resulta necesario un `framework` muy complejo, después de una comparación entre distintas herramientas se optó por `Flask`, que dada su estructura minimalista, facilitaría el trabajo en gran medida. Además de elegir el `API Tweepy`, que consultando sus métodos y variables de retorno, facilidad de implementación y basta documentación, mostró ser una herramienta adecuada para el proyecto. Finalmente el resultado del procesamiento de los datos del `timeline` para la tarea de lugar de residencia y ocupación, además de la actividad reciente, se ven reflejados en forma de gráficas de barras, implementaciones de la biblioteca `Chart.js`.

- `Flask: Microframework` escrito en Python planeado para facilitar el desarrollo de aplicaciones web bajo el patrón Modelo-Vista-Controlador (MVC). Posee las funcionalidades necesarias para crear una aplicación web funcional, extensiones adicionales están disponibles a la medida de los requerimientos.

[1] https://datos.gob.mx/busca/dataset/migracion-interna

- `Tweepy`: Este `API` proporciona acceso a todos los métodos de la `API RESTful`de `Twitter`. Cada método puede aceptar varios parámetros y devolver respuestas.
- `Chart.js`: Biblioteca de `JavaScript` que permite dibujar diferentes tipos de gráficos utilizando la etiqueta canvas de `HTML5`.

Etapa 4.2 Diseño de la aplicación web

El hecho de integrar el modelo clasificador en una herramienta web implica que esta debe ser capaz de recopilar el `timeline` completo para una cuenta de `Twitter` y mostrar los resultados de esta clasificación, además de información relevante sobre el perfil en cuestión.

MIMTA está inspirado en un cuadro de mando, por lo que sus elementos buscan una distribución adecuada para su correcta interpretación. Los colores utilizados en su diseño buscan generar una idea de ambiente minimalista y "fresco" (figura 5.3).

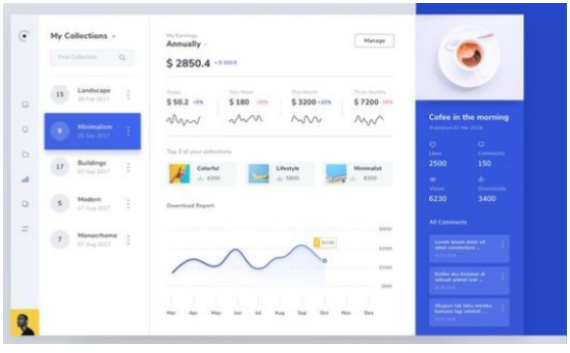

Figura 5.2: Ejemplo de cuadro de mando.

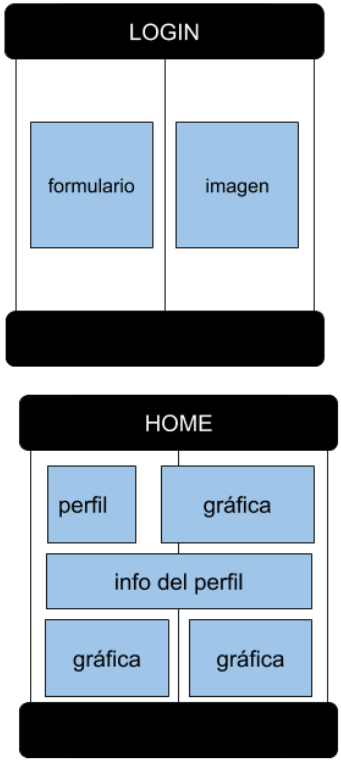

Figura 5.3: Mockups de MIMTA.

Acceso

Home

Figura 5.4: MIMTA.

Etapa 4.3 Desarrollo de la aplicación web

La aplicación web recibe como entrada un `username` de `Twitter`, para tener acceso a los recursos y métodos de esta red social se trabaja con `Tweepy`. A través de esta conexión se obtiene: perfil, actividad reciente, información del contenido del perfil y las gráficas de probabilidades para las tareas clasificatorias (figura 5.4).

En el caso de los `word embeddings`, se debe hacer una consulta por cada palabra al modelo de `fastText`. El texto completo del `timeline` queda representado por un vector de 100 dimensiones, después de sumar todos los vectores y dividirlos por el número de palabras presentes en el texto.

Una vez que el texto es operable, es dado como entrada al modelo clasificador y este obtiene la probabilidad de cada una de las clases para las que fue entrenado (6 para la tarea de lugar de residencia y 8 para la tarea de ocupación).

Finalmente, estos resultados son procesados por el módulo de graficos `Chart.js`, el cual despliega gráficos de barra para representar la probabilidad de cada clase (figura 5.5).

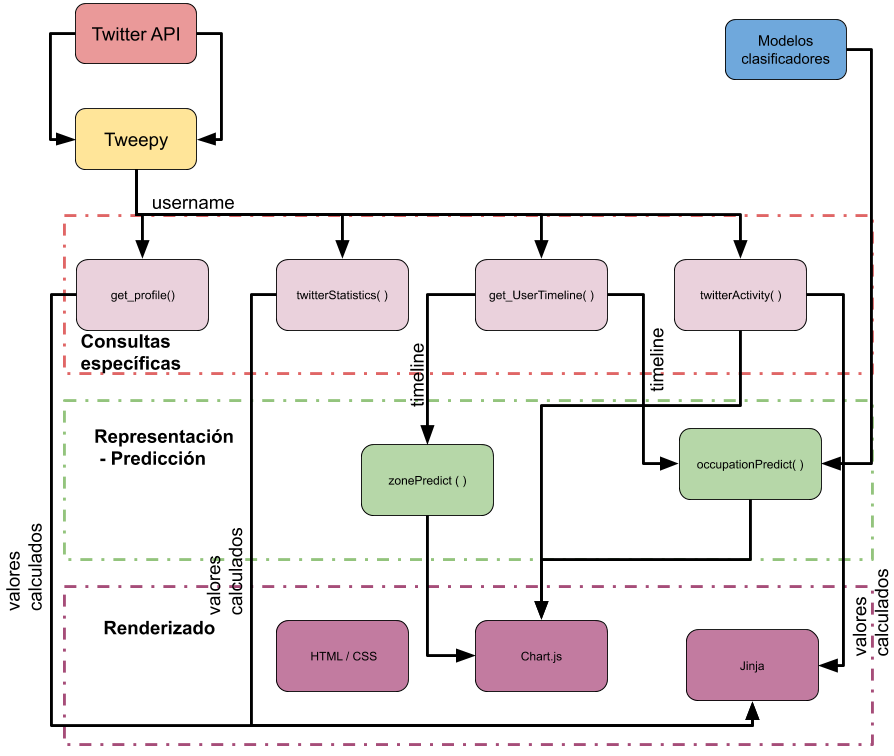

Figura 5.5: Estructura de los procesos en MIMTA.

Capítulo 6

Conclusiones

Se produjo una herramienta alternativa para el estudio de movimientos de migración interna en México, a partir de un método construido en etapas, denominado MIMTA. Esta herramienta web pretende cumplir con el rol asignado proporcionando resultados que podrían llegar a convertirse en una referencia para este tema social, en caso de que se aplique a una muestra poblacional amplia, pues permitiría equilibrar las muestras por clase.

El método expuesto en este proyecto está compuesto por dos bloques: el Bloque 1 está constituido por etapas fundamentales en el esquema de aprendizaje supervisado, y puntualiza el proceso de elaboración de un modelo clasificador y el Bloque 2 que detalla los módulos que componen la aplicación y las herramientas empleadas para su desarrollo. La factibilidad de que el modelo se incorpore en otros `frameworks web` más robustos que `Flask` como por ejemplo `Django`, permite afirmar que se construyó un método base y con propiedades incrementales al permitir la incorporación de la recolección de nuevas instancias.

Las configuraciones de preprocesamiento, representación de textos y algoritmos de aprendizaje supervisado, fueron elegidas acorde a revisiones de otros trabajos relacionados y propuestas personales. De esta forma se obtuvieron resultados variando los parámetros de dichas configuraciones, esto no implica que puedan existir configuraciones que arrojen mejores resultados. Pues se podría ahondar en ambas tareas y utilizar otros recursos para aumentar el rendimiento de los modelos clasificadores.

MIMTA es el resultado del método y `baseline` funcional, está abierto a modificaciones en su parte nuclear (modelo clasificador) y en la presentación de resultados. Se podrían recopilar otro tipo de datos y generar gráficas que representen los resultados con otros elementos. Con la finalidad de generar un recurso mantenible, y siempre con la intención de facilitar a sus usuarios una experiencia alternativa a los métodos tradicionales de estudio de movimientos migratorios.

Bibliografía

[1] La división dialectal del español mexicano.

[2] Faima Abbasi. Data forensics: Author attibution author profiling. pages 1–2.

[3] J.E.T. Akinsola. Supervised machine learning algorithms: Classification and comparision. *International Journal of Computer Trends and Technology*, 48(3):129–131, 2017.

[4] López Monrroy A.P. Aragón, E.M. Author profiling and aggressiveness detection in spanish tweets: Mex-a3t 2018. *In Proceedings of the Third Workshop on Evaluation of Human Language Technologies for iberian Languages (IberEval), CEUR WS Proceedings*, 2018.

[5] Jason Brownlee. Supervised and unsupervised machine learning algorithms.

[6] Azucena Montes-Rendón y Gerardo Sierra Carlos-Emiliano González-Gallardo, Juan-Manuel Torres-Moreno. Perfilado de autor multilingüe en redes sociales a partir de n-gramas de caracteres y de etiquetas gramaticales. *LinguaMática*, 8(1):21–29, 2016.

[7] CONAPO. Datos abiertos.

[8] Armando Suárez Cueto. Aprendizaje automático. 01 2008.

[9] Universidad Autónoma de Nuevo León. El análisis literario.

[10] Fintech e innovación. ¿quiénes son los 'millennials' y por qué son una generación única?

[11] Andreas C. Muller Sarah Guido. *Introduction to Machine Learning with Python*. O'Reilly Media, Inc, United States of America, 2017.

[12] Janet V. Hernández-García. Aplicación web para identificar personalidad, género y edad de usuarios en twitter. pages 93–106, 2016.

[13] Hootsuite. Digital in 2018: World's internet users pass the 4 billion mark - we are social.

[14] J.R. Mónica-S. Grigori I. Markov, H. Gómez-Adorno. cic-gil approach to author profiling in spanish tweets: Location and occupation. *In Proceedings of the Third Workshop on Evaluation of Human Language Technologies for iberian Languages (IberEval), CEUR WS Proceedings*, 2018.

[15] Yoshua Bengio Joseph Turian, Lev Ratinov. Word representations: A simple and general method for semi-supervised learning. Technical report, Association for Computational Linguistics, 2010.

[16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, April 2017.

[17] Yang Li and Tao Yang. *Word Embedding for Understanding Natural Language: A Survey*, volume 26. 05 2017.

[18] Juan M. Lope-Blanch. Nueva revista de filología hispánica. 19(1):1–11, 1970.

[19] Juan A. Martínez López. La palabra como unidad de significado: algunas excepciones.

[20] E.S. Tellez-D. Moctezuma V. Salgado J. Ortiz-Bejar C.N. Sánchez M. Graff, S. Miranda-Jiménez. Ingeotec at mex-a3t: Author profiling and aggressiveness analysis in twitter using micro-tc and evomsa. *In Proceedings of the Third Workshop on Evaluation of Human Language Technologies for iberian Languages (IberEval), CEUR WS Proceedings*, 2018.

[21] MEX-A3T. Mex-a3t: Authorship and aggressiveness analysis in twitter. case study in mexican spanish.

[22] Manuel Montes-y-Gómez Hugo Jair Escalante-Luis Villaseñor-Pineda Verónica Reyes-Meza Antonio Rico-Sulayes Miguel Álvarez Carmona, Estefanía Guzmán-Falcón. Overview of mex-a3t at ibereval 2018: Authorship and aggressiveness analysis in mexican spanish tweets. pages 1–23.

[23] Avinash Navlani. Text analytics for beginners using nltk.

[24] A.P. López-Monroy R.M. Ortega-Mendoza. The winning approach for author profiling of mexican users in twitter at nex-a3t@ibereval-2018. *In Proceedings of the Third Workshop on Evaluation of Human Language Technologies for iberian Languages (IberEval), CEUR WS Proceedings*, 2018.

[25] Dispanjan Sarkar. *Text Analytics with Python. A practical real-world approach to gaining actionable insights from your data.* Apress, Bagalore, Karnataka, 2016.

[26] SAS. Natural language processing.

[27] Fabrizio Sebastiani. Text categorization. pages 1–2.

[28] Fabrizio Sebastiani. Machine learning in automated text categorization. 34(1):1–47, March 2002.

[29] sitiobigdata.com. Machine learning: Seleccion métricas de clasificación.

[30] Aized Soofi and Arshad Awan. Classification techniques in machine learning: Applications and issues. 13:459–465, 08 2017.

[31] Expert System Team. What is machine learning? a definition.

[32] Jun Yan. *Text Representation*, pages 3069–3072. Springer US, Boston, MA, 2009.