Escuela Colombiana de Ingeniería
Julio Garavito

Proyecto de Grado

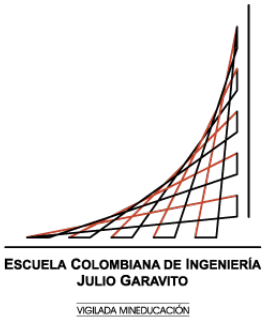
Advanced Natural Language Processing Techniques to Profile Cybercriminals

Autor:
Alejandro Anzola Ávila

Director:
Dr. Daniel Orlando Díaz López

Programa de Ingeniería de Sistemas

Bogotá, Colombia



6 de mayo de 2019

Índice general

Índice de figuras

Índice de cuadros

Escuela Colombiana de Ingeniería
Julio Garavito

Resumen

Programa de Ingeniería de Sistemas

Estudiante de Ingeniería de Sistemas

Advanced Natural Language Processing Techniques to Profile Cybercriminals

por Alejandro ANZOLA ÁVILA

In document language ... Escribir abstract

Escuela Colombiana de Ingeniería
Julio Garavito

Abstract

Programa de Ingeniería de Sistemas

Estudiante de Ingeniería de Sistemas

Advanced Natural Language Processing Techniques to Profile Cybercriminals

by Alejandro ANZOLA ÁVILA

In english....

Escribir abstract en ingles

Capítulo 1

Introducción

1.1. Objetivo general

El objetivo de este proyecto es generar herramientas y estrategias para el perfilado de cibercriminales con ayuda de metodologías de *Natural Language Processing (NLP*) aplicado a datos recolectados de comunicaciones y redes sociales.

1.2. Objetivos específicos

- Diseñar e implementar una solución de lenguaje natural para realizar el perfilado de sospechosos.
- Identificar el estado del arte en sistemas que usan NLP para apoyar agencias de seguridad del Estado.
- Implementación de artefactos para la construcción de *Dataset*s con información recolectada de medios privados como de fuentes abiertas.
- Validar la solución desarrollada frente a un escenario real.
- Modelado de diferentes metodologías, heurísticas y meta–heurísticas para NLP.

Capítulo 2

Cronograma

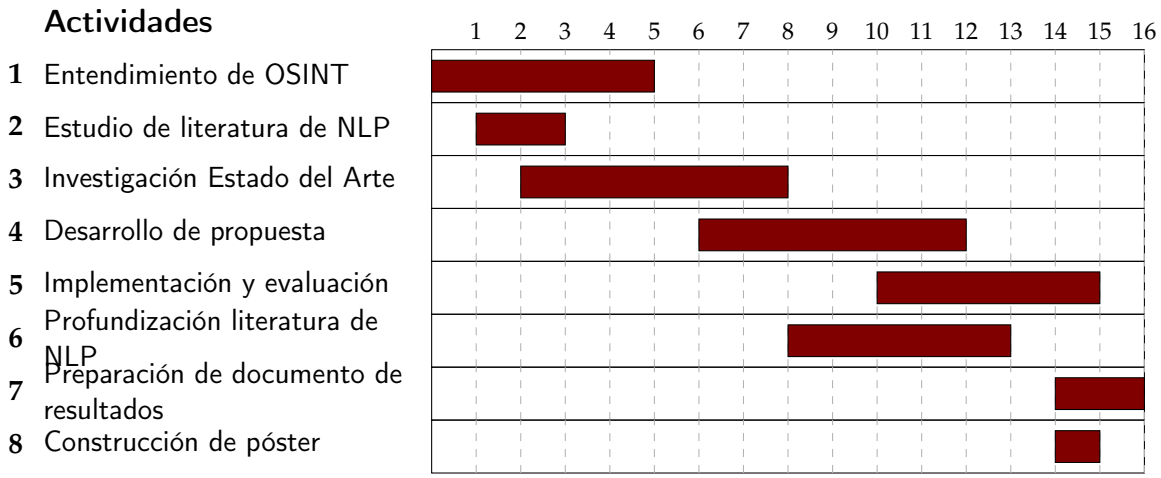


FIGURA 2.1: Diagrama Gantt de actividades de 1er periodo

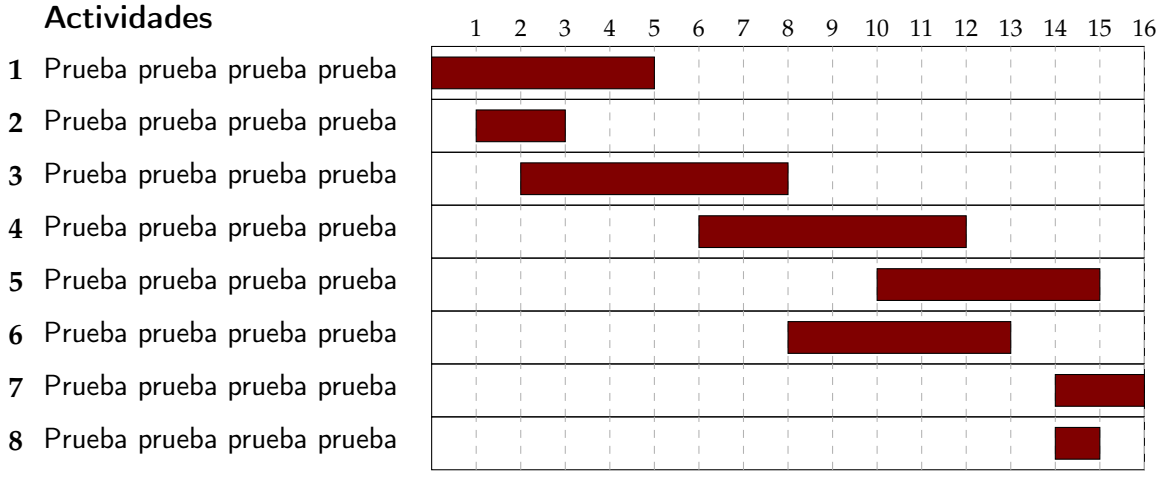


FIGURA 2.2: Diagrama Gantt de actividades de 2do periodo

	Detalle
1	Entendimiento de NLP y proyecto *Open Source Intelligence (OSINT*)
2	Estudo de la litaratura
3	Investigación del Estado del Arte
4	Desarrollo de la propuesta de investigación
5	Desarrollo de la implementación y pruebas
6	Realización de curso de NLP de National Research University Higher School of Economics de *Coursera*
7	Preparación final de documento de libro de proyectos y articulo de investigación
8	Construcción de póster para presentación en la Vitrina Académica

CUADRO 2.1: Detalle de cronograma de actividades

Capítulo 3

Marco teórico

La Web contiene una gran cantidad de opiniones respecto a productos, políticos, y mucho mas, expresado en forma de noticias, sitios de opinión, reseñas en tiendas online, redes sociales. Como resultado, el problema de "Minería de opinión" ha obtenido una atención creciente en las ultimas dos décadas y es un factor decisivo para las nuevas organizaciones (como es mencionado en [7]). De esto mismo partimos que el análisis de textos para extraer el significado y demás componentes extraíbles del texto componen un factor que debe considerarse al momento de realizar decisiones, de manera que los avances hechos hasta ahora tienen como meta una aplicación practica de lo que se conoce como Natural Language Processing.

Luego de los ataques terroristas del 11 de Septiembre de 2001 en Estados Unidos, se realizaron fuertes criticas respecto a la inteligencia, donde el director del FBI *Robert S. Mueller* indico que el principal problema que la agencia tuvo fue que se enfocaba demasiado en lidiar con el crimen luego de que fue cometido y ponía muy poco énfasis en prevenirlo (adaptado de [4]). Es por esto que el uso de NLP para temas de seguridad como también de metodologías de Machine Learning y Deep Learning han sido ampliamente utilizadas en ámbito de seguridad luego de estos eventos.

Para obtener una mejor inteligencia se necesito de mejores tecnologías a las que se tenían entonces (véase [4, pág 2]):

- Integración de datos (ó *Data Integration (DI*) en inglés)
- Análisis de vínculos (ó *Link Analysis (LA*) en inglés)
- Agentes de software (ó *Software Agents (SA*) en inglés)
- Minería de texto (ó *Text Mining (TM*) en inglés)
- Redes neuronales (ó *Artificial Neural Network (ANN*) en inglés)
- Algoritmos de Machine Learning (ó *Machine Learning Algorithms (MLA*) en inglés)

3.1. Análisis de vínculos (Link Analysis)

Es la visualización de asociaciones entre entidades y eventos, por lo general involucran una visualización por medio de una gráfica o un mapa que muestre las relaciones entre sospechosos y ubicaciones, sea por medio físico o por comunicaciones en la red.

3.2. Agentes de software (Software Agents)

Es el software que realiza tareas asignadas por el usuario de manera autónoma, donde sus habilidades básicas son:

- **Realización de tareas:** Hacen obtención de información, filtrado, monitoreo y reporte.
- **Conocimiento:** Pueden usar reglas programadas, o pueden aprender reglas nuevas (véase 3.5).
- **Habilidades de comunicación:** Reportar a humanos e interactuar con otros agentes.

3.3. Aprendizaje de maquina (Machine Learning)

De acuerdo con [6], se define como un conjunto de métodos que pueden detectar patrones automáticamente en datos, y luego usar los patrones descubiertos para predecir los datos futuros, o realizar otra clase de toma de decisiones con un grado de incertidumbre, por tal motivo es necesario el uso de teoría de probabilidad, que puede ser aplicada a cualquier tipo de problema que involucra incertidumbre.

3.3.1. Tipos de Machine Learning

Machine Learning (ML) esta principalmente dividida en dos tipos. El método predictivo o bien **aprendizaje supervisado** (*Supervised Learning*), donde el objetivo es aprender un mapeo de las entradas $\mathbf{x}$ a las salidas y, dado un conjunto de pares de etiquetas de entrada–salida $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. D se le llama el conjunto de entrenamiento y N es el numero de muestras de entrenamiento.

En la forma mas sencilla, cada entrada de entrenamiento $\mathbf{x}_i$ es un vector D–dimensional de números, a estos se le llaman *características* o *atributos*.

De manera similar la forma de la salida puede ser en principio cualquier cosa, pero la mayoría de métodos asumen que y_i es una variable *categórica* o *nominal* de algún conjunto finito, $y_i \in \{1, \ldots, C\}$, o que y_i es un escalar real, $y_i \in \mathbb{R}$. Cuando la variable y_i es categórica, al problema se le reconoce como **clasificación** o **reconocimiento de patrones**, y cuando es un valor real se le conoce como un problema de **regresión**.

El segundo tipo principal de ML es el descriptivo o **aprendizaje no–supervisado** (*Unsupervised Learning*), en este solo están disponibles los datos de entrada $D = \{\mathbf{x}_i\}_{i=1}^{N}$, y la meta es encontrar "patrones interesantes" en los datos. Este es un problema mucho menos definido, debido a que no se conocen los tipos de patrones que se quieren encontrar, y no hay una métrica obvia de error (no como aprendizaje supervisado en la que se puede comparar nuestra predicción de y para un $\mathbf{x}$ con el valor observado).

Un tercer tipo de aprendizaje de maquina es conocido como *Reinforcement Learning*, el cual es un tipo menos usado. Este es útil cuando se quiere aprender como actuar o comportarse cuando se recibe una recompensa ocasional o una señal de castigo.

3.3.2. Sistemas de Detección de Anomalías (Anomaly Detection Systems)

Existen diferentes aproximaciones para los sistemas de anomalías, sin embargo una similitud entre todos estos sistemas es que se intenta realiza una *detección de desviaciones*, y su tarea es detectar los datos *atípicos* en un sistema [9].

Uno de los que se pueden encontrar en la literatura son los sistemas de detección de anomalías basados en realizar un estimado probabilístico con alguna distribución de probabilidad de donde para una serie de características m se trata de estimar una distribución gaussiana $X \sim \mathcal{N}(\mu, \sigma^2)$ que tiene media μ y varianza σ^2 por cada característica, por lo que existirán m diferentes distribuciones .

Para estimar cada una las medias y variaciones de cada característica j, μ se estima con la Ecuación 3.1 y σ^2 se estima con la Ecuación 3.2.

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \tag{3.1}$$

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2 \tag{3.2}$$

De donde para calcular la probabilidad de que una muestra se trata de una anomalía se calcula la probabilidad $p(x)$, luego de que fueron estimadas las distribuciones de cada característica del conjunto de entrenamiento, por lo que se define un ϵ de manera heurística, de forma que se determina que una muestra anómala si $p(x) < \epsilon$, la Ecuación 3.3 representa cual es la probabilidad de una muestra x con una distribución gaussiana.

$$p(x) = \prod_{j=1}^{n} p(x_j, \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right) \tag{3.3}$$

Si el modelo esta dando como resultado muchos falsos positivos, lo que se debe hacer es reducir σ.

Este método de sistemas de detección de anomalía también es brevemente tratada en [3] con una representación de hiper-planos, que es equivalente con este descrito.

3.4. Mineria de datos (Data Mining)

Según [9], la minería de datos se define como el proceso de descubrir información útil en repositorios grandes de datos. Las técnicas de minería de datos son desplegadas para limpiar grandes bases de datos para encontrar patrones nuevos y útiles que de lo contrario podrían permanecer desconocidos. También ofrecen capacidades para predecir la salida de observaciones futuras, tales como predecir si un cliente nuevo gastara mas de $100 en una tienda.

No todas las tareas de descubrimiento de información son considerados como *Data Mining (DM)*. Por ejemplo, realizar una consulta de campos individuales usando un sistema de base de datos o encontrar una pagina web por medio de una búsqueda en Internet son tareas relacionadas con *adquisición de información*.

3.4.1. Mineria de texto (Text Mining)

Es un subcampo de Inteligencia Artificial conocida como Natural Language Processing, en donde las herramientas de minería de datos pueden capturar rasgos críticos del contenido de un documento basado en el análisis de sus características lingüísticas.

La mayoría de los crímenes son electrónicos por naturaleza, por lo que se dejan rastros textuales que investigadores pueden seguir y analizar. Estas se enfocan en el descubrimiento de relaciones en texto no–estructurado y pueden ser aplicados al problema de *búsqueda* y *localización de palabras clave*.

3.4.2. Clasificación

Clasificación es la tarea de asignarle una de varias categorías predefinidas a objetos, y es una tarea que tiene una variedad extensa de aplicaciones. Ejemplos de esto se encuentran la detección de correos no deseados en mensajes de e–mails basándose del encabezado o el cuerpo del mensaje, categorización de células benignas de malignas basándose en los resultados de escaneados MRI o incluso la clasificación de galaxias basado en su forma.

Definido formalmente, clasificación es la tarea de aprender una función objetivo f que mapea cada conjunto de atributos x a una clase predefinida de etiquetas y.

La función objetivo también se define informalmente como un *modelo de clasificación*.

Metodologías de clasificación

Existen muchos métodos para la clasificación de datos no–estructurados, entre los descritos aquí están:

- Clasificador basado en reglas (véase 3.5)
- Redes neuronales artificiales (véase 3.6)
- Maquinas de soporte vectorial (véase 3.7)
- Clasificador de Naïve Bayes (véase 3.8.2)

3.4.3. Clustering

El análisis de clusters agrupa objetos de datos basándose únicamente en la información encontrada en los datos que describen los objetos y sus relaciones. El objetivo es que objetos dentro de un grupo sean similares (o relacionados) el uno al otro, y que sean diferentes (o sin relación) a objetos en otros grupos. Entre mayor sea la similitud dentro de un grupo y entre mayor sea la diferencia entre grupos, sera mejor o mas distintivo el clustering.

Los métodos de clustering se hacen referencia comúnmente en ML como métodos no–supervisados, los cuales se describen en 3.3. Un método de estos se describe en 3.6.1 conocidos como mapas autoorganizados.

3.5. Sistemas Basados en Conocimiento (Knowledge Based Systems)

Según [8], los *Knowledge Based Systems (KBS)* son uno de los mayores miembros de la familia de *Artificial Intelligence (AI)*. El KBS consiste de una *Knowledge Base (KB)* y un programa de búsqueda llamado *Inference Engine (IE)* representado en la Figura 3.1. La KB puede ser usado como un repositorio de conocimiento de varias formas.

Existen 5 tipos de KBS, donde uno de ellos es conocido como *Expert Systems (ES)*, usados como *Rule–based Systems (RBS)*, donde su KB esta dado como reglas y el IE esta dado por algo llamado *Working Memory (WM)*, que representa los hechos que se conocen inicialmente del sistema junto con los hechos que se van dando como inferencia de las reglas.

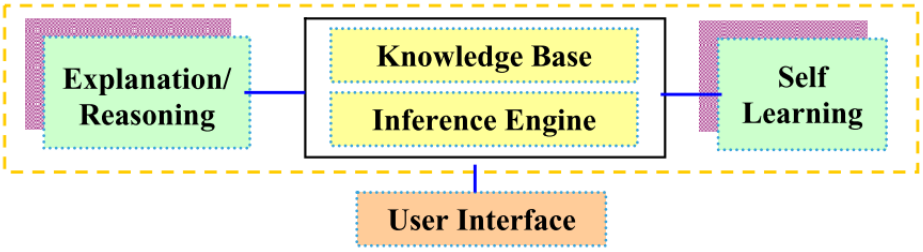


FIGURA 3.1: Arquitectura KBS. Tomado de [8]

Estas reglas pueden resumirse como una colección de condicionales de la forma **IF/ELSE** que se componen de un *antecedente* y un *consecuente*.

Existen dos tipos de RBS, definidos como *Deductive Systems (DS)* y *Reactive Systems (RS)*, donde el DS tiene como objetivo realizar una conclusión en base a los hechos iniciales en la WM, por el otro lado se tienen los RS, los cuales de igual manera a los DS, toman los hechos de la WM y realizan sea una acción interactiva con su entorno o bien una modificación de los hechos que se encuentran en la WM tal como la adición o eliminación de hechos. Tómese el ejemplo de la Ecuación 3.4 tomada de [5], donde x es la temperatura y AC es aire acondicionado.

$$\begin{cases} \text{IF x es moderado,} & \text{THEN y = ajustar AC a bajo} \\ \text{IF x es alto,} & \text{THEN y = ajustar AC a moderado a alto} \\ \text{IF x es muy alto,} & \text{THEN y = ajustar AC a alto} \end{cases} \tag{3.4}$$

3.6. Redes Neuronales Artificiales (Artificial Neural Network)

El estudio de redes neuronales artificiales fue inspirado por los intentos de simular los sistemas biológicos de neuronas. El cerebro humano se compone principalmente de células nerviosas llamadas *neuronas*, enlazadas con otras neuronas por medio de hebras de fibra conocidas como *axones*. Los axones son usados para transmitir impulsos nerviosos de una neurona a otra cada vez que las neuronas son estimuladas. Una neurona esta conectada a axones de otras neuronas por medio de *dendritas*, las cuales son extensiones

desde el cuerpo de la neurona. El punto de contacto entre una dendrita y un axón se conoce como *sinapsis*. Los neurólogos han descubierto que el cerebro humano aprende por medio de cambiar la fuerza de la conexión sináptica entre las neuronas a través de estimulación repetitiva por el mismo impulso.

De manera análoga a la estructura del cerebro humano, una ANN se compone de una estructura interconectada de nodos y vínculos directos.

3.6.1. Mapa autoorganizado (Self-organizing Maps)

El objetivo principal de los *Self-organizing Maps (SOM)* es de transformar una patrón de entrada m–dimensional en un mapa discreto uni– o bi–dimensional, donde sus principales características es que es un algoritmo que se basa en Unsupervised Learning, es *Feedforward*, tiene una sola capa de neuronas donde su propósito es realizar *Clustering* y una reducción de dimensionalidad sobre los datos de una forma topologicamente ordenada.

Los SOM tienen tres características distintivas:

- **Competencia:** por cada patrón de entrada, las neuronas en la red competirán entre ellas para determinar un ganador.
- **Cooperación:** la neurona ganadora determina la ubicación espacial (vecinos) alrededor de donde otras vecinas también se verán estimuladas.
- **Adaptación:** la neurona ganadora como también sus vecinas tendrán sus pesos asociados actualizados, y se tiene que los vecinos entre mas cerca estén del ganador, mayor es el grado de adaptación.

El algoritmo de aprendizaje de SOM parte de primero inicializar los pesos de las o neuronas con pesos aleatorios pequeños de una distribución de probabilidad aleatoria o uniforme, donde cada vector de entrada se define como $x = [x_1, \ldots, x_m]^T \in \mathbb{R}^m$ y la entrada general de N patrones como $\mathbf{X}^{m \times N}$, el vector de pesos de la neurona i es $\mathbf{w}_i = [w_{i1}, \ldots, w_{im}] \in \mathbb{R}^{1 \times m}$, con la matriz de pesos $\mathbf{W}^{o \times m}$.

Para alcanzar el objetivo de *competencia*, se realiza por cada patrón de entrada x_i una comparación con cada uno de los pesos de las o neuronas y se establece la de menor distancia respecto x_i (típicamente la distancia Euclidiana), dejando un ganador *winner*, tal como en la Ecuación 3.5.

$$winner = \text{argmin}_j \left\| x_i - w_j \right\| ; j = 1, \ldots, o \tag{3.5}$$

Luego de establecer la neurona ganadora, se realiza el paso para alcanzar la *cooperación*, que consiste en que por medio de una función kernel h (típicamente una una distribución gaussiana), que permite establecer un área de afectación de las otras neuronas según su ubicación física en el mapa, definidos como r_{winner} y r_j que son la ubicación de la neurona ganadora y la neurona vecina j, en el cual el grado de afectación de la neurona vecina depende de la distancia de la que esta de la neurona ganadora, definido en la Ecuación 3.6.

$$h_{j,winner}(t) = \exp\left(\frac{- \left\| r_j - r_{winner} \right\|^2}{2\sigma(t)^2} \right) \tag{3.6}$$

Parte importante del proceso de convergencia del SOM es que a medida que avanzan las iteraciones t del algoritmo el área de afectación se va reduciendo como parte del proceso de adaptación, por lo que definimos $\sigma(t) = \sigma_0 \exp(-t/\tau_1)$, donde τ_1 es una constante heurística y σ_0 la dimensión del mapa SOM.

Finalmente para alcanzar la *adaptación* se realiza una actualización de los pesos de la matriz **W** en base a la influencia de área $\sigma(t)$ y de una tasa de aprendizaje $\alpha(t) = \alpha_0 \exp(-t/\tau_2)$, donde τ_2 es otra constante heurística y α_0 es una constante de aprendizaje inicial, que debe ser $0 \leq \alpha_0 \leq 1$, la actualización se describe por la Ecuación 3.7 y el proceso puede ser visto gráficamente en la Figura 3.2, tanto de forma uni– como bi–dimensional.

$$w_j(t+1) = w_j(t) + \alpha(t) h_{j,winner}(t) \Big[x_i - w_j(t) \Big] \tag{3.7}$$

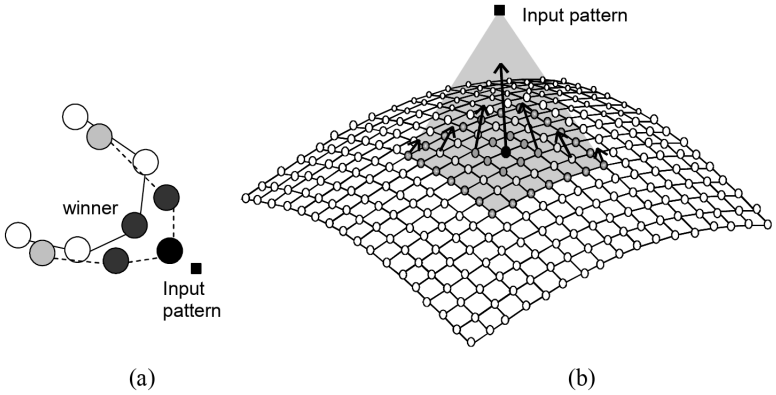


FIGURA 3.2: Proceso de adaptación de SOM, (a) uni–dimensional, (b) bi–dimensional. Tomado de [1]

Luego de que el algoritmo de aprendizaje termina de realizar las iteraciones, la salida de este es la matriz de pesos **W**, en la Figura 3.3 se puede apreciar una aproximación del algoritmo con un mapa uni–dimensional tratando de aproximar una función sinusoidal con ruido adicionado en un gráfico 2D.

En la Figura 3.4 se puede ver una aplicacion de los SOM, en donde se realiza una clusterizacion de casos de homicidios donde los parametros son caracteristicas de los homicidios, segun [4] este resultado da una buena aproximacion para sospechar de que estos son cometidos por personas distintas o si bien estan siendo perpetrados por un mismo individuo o grupo.

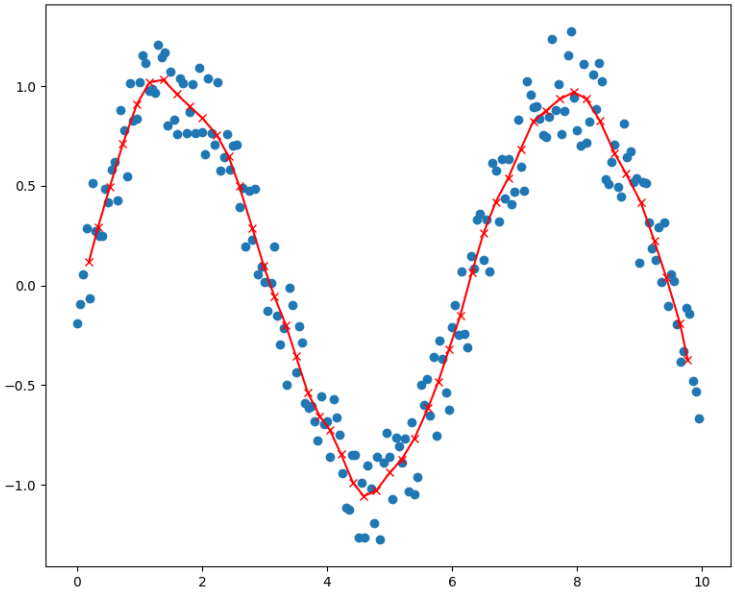


FIGURA 3.3: Ejemplo de salida de SOM uni-dimensional. Implementación propia.

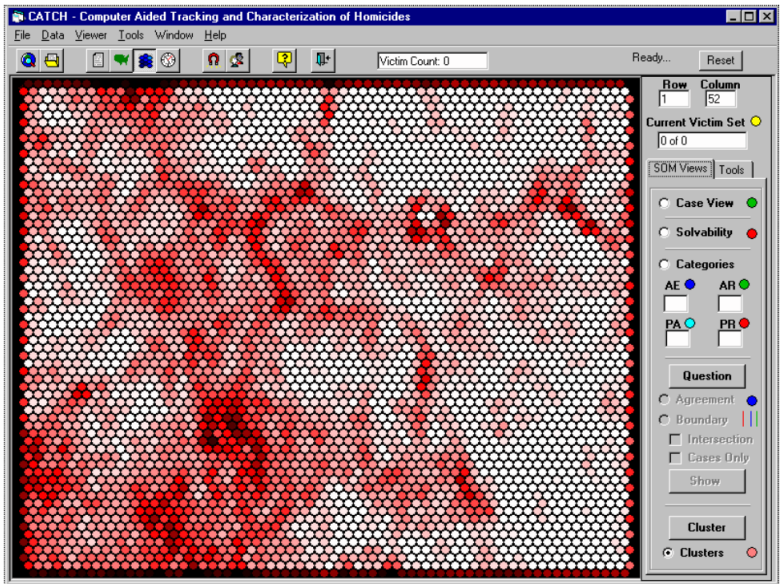


FIGURA 3.4: Ejemplo de uso de SOM en aplicaciones de perfilado. Tomado de [4]

3.7. Maquina de soporte vectorial (Support Vector Machine)

Support Vector Machine (SVM) es una técnica de clasificación que tiene sus raíces en la teoría de aprendizaje estadístico que ha mostrado resultados empíricos prometedores en muchas aplicaciones practicas, desde reconocimiento de dígitos escritos a mano a categorización de texto. SVM también funciona muy bien con datos de alta dimensionalidad. Otro aspecto destacable de esta aproximación es que representa la frontera de decisión usando un subconjunto de las muestras de entrenamiento, conocidos como los *support vectors*.

3.7.1. Maximum Margin Hyperplanes

Se puede entender a los *Maximum Margin Hyperplanes* como hiper-planos que ayudan a separar datos en un hiper-espacio y que poseen un margen de decisión entre los datos, como ejemplo tómese la Figura 3.5, donde el hiper-plano B_1 tiene un margen de decisión mas grande que el hiper-plano B_2.

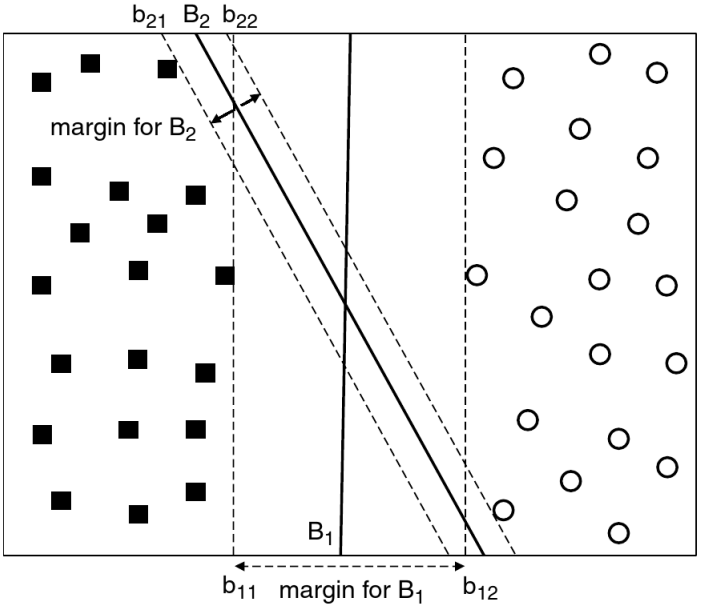


FIGURA 3.5: Maximum Margin Hyperplanes. Tomado de [9]

Finalmente, el objetivo final de los SVM es la búsqueda de un hiper-plano con el mayor margen de decisión. Existen dos tipos de SVM, el lineal y el no–lineal. El lineal realiza la separación de los datos con su hiper-plano a partir de los datos de entrada en su espacio vectorial original, mientras que el no–lineal consta de realizar una transformación de los espacios de los datos de entrada a uno en que sea linealmente separable (véase como ejemplo la Figura 3.6), sin embargo al realizar la transformación, el algoritmo de SVM se ve afectado por la dimensionalidad de la entrada, por lo que existe lo que se conoce como la función *kernel* para remediarlo.

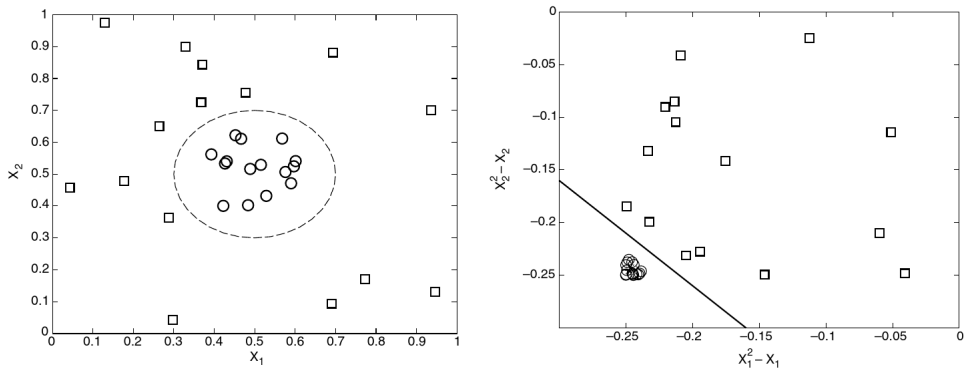


FIGURA 3.6: Transformación de espacios en Support Vector Machine. Tomado de [9]

3.7.2. Función Kernel

La función polinomial de similaridad, K, la cual es calculada en el espacio original de los datos de entrada, se le conoce como la **función Kernel**. En principio se asegura que la función kernel puede ser expresada siempre como el producto punto entre dos vectores de entrada en algún espacio de alta dimensionalidad, la función de kernel también tiene la particularidad de que el computo de los productos punto con la función toman considerablemente menos tiempo que realizar la transformación de espacios, dejando de lado la transformación, acelerando la tarea de clasificación.

3.8. Clasificadores Bayesianos

En muchas aplicaciones de relaciones entre el conjunto de atributos y la etiqueta es no–determinante. Es decir, la etiqueta de clase de un dato de un conjunto de prueba no puede ser determinado con certeza a pesar de ser un atributo idéntico a los atributos de entrenamiento. Esto puede ser producto de que los datos poseen ruido o la presencia de ciertos factores que afectan la clasificación pero no son incluidos en el análisis. Para esto es crucial el teorema de Bayes, el cual es un principio estadístico que combina el conocimiento previo de las clases con la nueva evidencia que se obtiene de los datos.

3.8.1. Teorema de Bayes

El teorema de Bayes dice que para un par de variables aleatorias X y Y y que $P(X = x|Y = y)$ la probabilidad de que la variable X tome el valor x dado que el valor de la variable Y es y. Se tiene entonces la Ecuación 3.8.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{3.8}$$

Usando el teorema de Bayes para clasificación

Para denotar el problema de clasificación desde una perspectiva estadística se define a X como el conjunto de atributos y Y como el conjunto de etiquetas de clase. Si la etiqueta de clase tiene una relación no–determinante con los atributos, entonces se pueden tomar a X y a Y como variables aleatorias y capturar su relación probabilística con $P(Y|X)$, conocida como la probabilidad posterior para Y, dada su probabilidad previa $P(Y)$.

Durante la fase de entrenamiento, es necesario aprender las probabilidades posteriores $P(Y|X)$ para cualquier combinación de X y Y basándose en la información recolectada de los datos de entrenamiento.

Dado que lo que se quiere realizar es una clasificación que represente la probabilidad de que dado un valor de $X = x$ este relacionado con que $Y = y$, se puede reconocer primero que X se mantiene constante para lo que son los datos de entrenamiento, y que lo desconocido sea la clasificación $Y = y$ con probabilidad $P(Y|X)$, al conocer esta probabilidad, un valor de prueba X' puede ser clasificado por medio de encontrar la clase Y' que maximice la probabilidad posterior $P(Y'|X')$.

3.8.2. Clasificador Naïve Bayes

Un clasificador de Naïve Bayes estima la probabilidad condicional de las clases por medio de suponer que los atributos son condicionalmente independientes, dado la etiqueta de clasificación y. La suposición de independencia condicional se puede dar por la Ecuación 3.9.

$$P(X|Y = y) = \prod_{i=1}^{d} P(X_i|Y = y) \tag{3.9}$$

Donde cada conjunto de atributos $X = \{X_1, \ldots, X_d\}$ que consiste de d atributos.

Como funciona el clasificador Naïve Bayes

Con la suposición de independencia condicional, en vez de computar la probabilidad condicional de clases para cada combinación de X, solo se debe realizar para establecer la probabilidad condicional de cada X_i, dado Y.

Para clasificar un dato de prueba, el clasificador computa la probabilidad posterior para cada clase Y como se muestra en la Ecuación 3.10.

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(X)} \tag{3.10}$$

Capítulo 4

Estado del arte

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus

eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Capítulo 5

Propuesta

La propuesta consta de *n* modelos, pensados para el análisis de texto en redes sociales como Twitter en aras de realizar un perfilado de ciber-criminales potenciales, esto por medio de NLP, de donde se parte varias metodologías que hacen uso de tecnologías Estado-del-Arte.

¿cuantos modelos?

5.1. Análisis

averiguar que se pone en esta sección

5.2. Diseño

Como parte de la propuesta se proponen *n* modelos para tratar diferentes aspectos en perfilado de donde se representan los diferentes modelos en la Figura 5.1.

¿cuantos modelos?

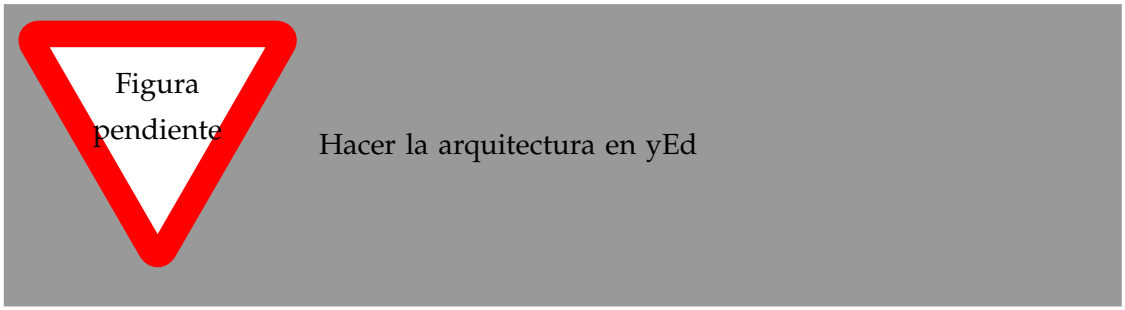


FIGURA 5.1: Arquitectura de propuesta.

5.2.1. Predicción de etiquetas de Twitter con modelos lineales

En Twitter, las publicaciones que se realizan tienen la posibilidad de incluir menciones de temas de tendencia por el conocido *hashtag*, escrito como `#Tema`, y tiene la gran utilidad de realizar una mención explicita del tema que se quiere tratar y donde además la tarea de encontrar textos directamente relacionados con un tema son fácilmente localizables.

Así mismo, en la literatura de NLP es muy común el uso de diferentes representaciones de palabras o conjuntos de palabras. Una representación de palabras típicas es por medio de los *Bag of Words (BoW)*, donde se establece un diccionario de palabras de

tamaño N, y donde cada palabra tiene un vector que lo representa. A cada palabra se le asigna un identificador único en ese diccionario, por lo que existiría una traducción de palabra a identificador y una secuencia de palabras para poder ser recuperado por medio del índice, como se muestra en la Ecuación 5.1 y la Ecuación 5.2.

$$\text{word2idx} = \Big\{(\text{word}_i, i) : \forall i \in [1, \ldots, N]\Big\} \tag{5.1}$$

$$\text{idx2word} = \Big[\text{word}_i\Big], \forall i \in [1, \ldots, N] \tag{5.2}$$

Otra representación común en NLP es la de *Term Frequency - Inverse Document Frequency (TF-IDF)*, que se divide en dos partes, expresadas en las Ecuación 5.3, la Ecuación 5.4 y la composición de ambas en la Ecuación 5.5, D es el corpus de palabras. Este consiste en penalizar palabras que ocurren mucho en un documento pero no mucho en el corpus o bien de penalizar la palabras que se repiten poco en un documento pero se repiten mucho en el corpus, por lo que un punto medio entre ambos es recompensado.

$$\text{tf}(t, d) = \text{Frecuencia del termino (o n–grama) } t \text{ en el documento } d \tag{5.3}$$

Existen diferentes variaciones para realizar representar el conteo de términos **tf** de forma normalizada, como se representa en el Cuadro 5.1.

Esquema	**Peso de tf**
Binario	$0, 1$
Conteo directo	$f_{t,d}$
Frecuencia de términos	$f_{t,d} / \sum_{t' \in d} f_{t',d}$
Normalización logarítmica	$1 + \log(f_{f,d})$

CUADRO 5.1: Variaciones de **tf**

$$\text{idf}(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right); N = |D| \tag{5.4}$$

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D) \tag{5.5}$$

Creación del Corpus de palabras

Debido a que los textos que se encuentran en Twitter no tienen forma estructurada es necesario realizar un preprocesamiento de cada post recolectado para luego contar la frecuencia de cada palabra dentro del corpus y así establecer las primeras N palabras mas usadas que van a componer el corpus de palabras.

Para realizar el preprocesamiento de cada texto se hacen los siguientes pasos:

1. Convertir todas las palabras a minúscula (e.g. “LaTeX” → “latex”)
2. Reemplazar todos los caracteres especiales de texto a espacios en blanco (e.g. “`@;,:\n\t\r`” → “␣␣␣␣␣␣␣”)

3. Remover todos los símbolos extraños, es decir todo lo que no sean numeros, ni letras ni los simbolos que se encuentran normalmente en tweets (e.g. “ %()&$!^” → “”)

4. Remover todas las *stopwords*, que son palabras que no añaden ningún valor semántico al texto
(e.g. “las palabras son una forma de expresarnos” → “palabras forma expresarnos”)

Luego se realiza un conteo de todas la palabras presentes dentro del corpus de donde se sacan las N primeras palabras para incluirlas en el BoW.

Luego de esto se generan las etiquetas (ó *tags* en ingles) que se toman directamente de los textos de entrenamiento, de estos también se les realiza un conteo, que servirán para la predicción de las etiquetas según el contenido del texto.

Conversión de textos a vectores

Para realizar la conversión de textos a vectores y así poder representar un texto como un vector se procede a primera realizar una generación de identificadores para cada palabra de manera como se describió en el inicio de la Sección 5.2.1.

La manera en que se representa un texto en forma de vector es por medio de la sumatoria de los vectores que representan cada palabra.

Generación del conjunto de entrenamiento, validación y pruebas

Para la generación de los conjuntos se toman las sumatorias generadas de cada tweet en su forma de vector y se coloca en una matriz de $T^{m \times N}$, donde m son el numero de muestras de Twitter y N el tamaño del Corpus.

Clasificador de regresión logística

Pendiente

Bla

Multiclasificador de One VS Rest

Pendiente

Bla

5.2.2. Reconocimiento de Named Entities con redes Long Short Term Memory

Pendiente

Bla

incluir diagramas de pagina de Google

5.2.3. Búsqueda de tweets relacionados con *embeddings* de StarSpace

Bla

Pendiente

5.3. Resultados

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu

neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula

hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Glosario

Símbolos

Clustering

Pendiente. 9

Coursera

Sitio web de cursos de aprendizaje en https://www.coursera.org. 3

Dataset

Pendiente. 1

Feedforward

Pendiente. 9

Maximum Margin Hyperplanes

Hiperplanos que permiten separar datos en espacios de alta dimensionalidad con un margen asociado para separarlos.. 12

Reinforcement Learning

Pendiente. 5

Supervised Learning

Pendiente. 5

Unsupervised Learning

Pendiente. 5, 9

Artificial Intelligence (AI)

Pendiente. 8

Bag of Words (BoW)

Pendiente. 17, 19

Data Integration (DI)

Para acceder a múltiples y diversas fuentes de información. 4

Data Mining (DM)

Proceso de descubrir automáticamente información útil en repositorios grandes de datos. 6

Deductive Systems (DS)

Pendiente. 8

Expert Systems (ES)

Pendiente. 8

Inference Engine (IE)

Pendiente. 8

Knowledge Based Systems (KBS)

Pendiente. 8

Knowledge Base (KB)

Pendiente. 8

Link Analysis (LA)

Para visualizar asociaciones y relaciones criminales y terroristas. 4

Machine Learning (ML)

Informalmente ha sido definido como "El campo de estudio que le da a computadores la habilidad de aprender sin ser explícitamente programados", este tiene tres tipos de algoritmos de aprendizaje: aprendizaje supervisado, aprendizaje no–supervisado, y aprendizaje por refuerzo. 5, 7

Machine Learning Algorithms (MLA)

Para extraer perfiles de perpetradores y mapas gráficos de crímenes. 4

Artificial Neural Network (ANN)

Para predecir la probabilidad de crímenes y nuevos ataques terroristas. 4, 9

Natural Language Processing (NLP)

Rama de la inteligencia artificial que lidia con la interacción entre computadores y humanos usando el lenguaje natural. 1–4, 17, 18

Open Source Intelligence (OSINT)

Disciplina responsable de la adquisición, procesamiento y posterior transformación en inteligencia de información obtenida de fuentes públicas como prensa, radio, televisión, internet, informes de diferentes sectores y, en general, cualquier recurso de acceso público (Tomado de [3]). 3

Reactive Systems (RS)

Pendiente. 8

Rule–based Systems (RBS)

Pendiente. 8

Software Agents (SA)

Para el monitoreo, obtención, análisis y actuación sobre la información. 4

Self-organizing Maps (SOM)

Pendiente. 9, 10

Support Vector Machine (SVM)

Pendiente. 12

Text Mining (TM)

Búsqueda sobre terabytes de información en documentos, paginas web y correos electrónicos. 4

Term Frequency - Inverse Document Frequency (TF-IDF)

Pendiente. 18

Working Memory (WM)

Pendiente. 8

Bibliografía

[1] Leandro Nunes De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. Chapman y Hall/CRC, 2006.

[2] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[3] Martín José Hernández Medina, Ricardo Andrés Pinto Rico y Cristian Camilo Pinzón Hernández. *Inteligencia de fuentes abiertas para el contexto colombiano*. Inf. téc. Escuela Colombiana de Ingeniería Julio Garavito, 2018.

[4] J. Mena. *Investigative Data Mining for Security and Criminal Detection*. Elsevier Science, 2003. ISBN: 9780080509389. URL: https://books.google.com.co/books?id=3mDlrtJuZv4C.

[5] Jerry M Mendel. *Uncertain Rule-Based Fuzzy Systems*. ISBN: 9783319513690. DOI: 10.1007/978-3-319-51370-6.

[6] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013. ISBN: 9780262018029 0262018020. URL: https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.

[7] Ana Maria Popescu y Orena Etzioni. «Extracting product features and opinions from reviews». En: *Natural Language Processing and Text Mining* (2007), págs. 9-28. ISSN: 08247935. DOI: 10.1007/978-1-84628-754-1_2. arXiv: 0309034 [cs].

[8] Priti Srinivas Sajja y Rajendra Akerkar. «Knowledge-based systems for development». En: *Advanced Knowledge Based Systems: Model, Applications & Research* 1 (2010), págs. 1-11.

[9] Pang-Ning Tan, Michael Steinbach y Vipin Kumar. *Introduction to Data Mining*. US ed. Addison Wesley, mayo de 2005. ISBN: 0321321367. URL: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0321321367.

[10] Patrick Henry Winston. *Artificial Intelligence (3rd Ed.)* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992. ISBN: 0-201-53377-4.

[11] L. Wu y col. «StarSpace: Embed All The Things!» En: *arXiv preprint arXiv:1709.03856* (2017).