UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS

CLASIFICACIÓN AUTOMÁTICA DE TEXTOS CORTOS POR GÉNERO Y GRUPO ETARIO

QUE PARA OBTENER EL TÍTULO DE
INGENIERO "**EN COMPUTACIÓN**"

PRESENTA:

"**CARLOS EMILIANO GONZÁLEZ GALLARDO**"

DIRECTORA DE TESIS:
DRA. AZUCENA MONTES RENDÓN

CIUDAD UNIVERSITARIA 10/febrero/2016

RESUMEN

La presente investigación provee de un prototipo de software para la clasificación automática de textos cortos por género y grupo etario que ha sido aplicado a la red social Twitter. Este prototipo hace uso del Aprendizaje de Máquina para entrenar un sistema clasificador a partir de características estilísticas, con la intención de hacerlo lo más independiente del idioma. Cabe señalar que un proceso de re-etiquetado denominado Normalización dinámica dependiente del contexto se lleva a cabo con la intención de aprovechar los elementos sintácticos propios de la red social.

Los resultados obtenidos son mostrados primeramente a partir de un grupo de datos provistos por los organizadores del concurso *PAN2015*; posteriormente una comparación entre la presente propuesta y las otras siete mejores propuestas es realizada a partir de los resultados oficiales liberados por los organizadores.

Se realizaron pruebas en los siguientes idiomas: español, inglés, italiano y holandés, siendo el italiano el que mejores resultados mostró e inglés el que obtuvo un rendimiento menor.

A partir de los resultados obtenidos, es posible concluir que es de gran importancia mantener toda la información que la red social pueda proveer, pues las características de los textos que presentan las redes sociales difieren en gran medida con las características de los textos de longitud amplia, por lo que es necesario buscar elementos extras que puedan ayudar a caracterizarlos de mejor forma.

ABSTRACT

The present investigation provides a software prototype for automatic classification of short texts by gender and age that has been used in the social network named Twitter. This prototype uses Machine Learning to train a classifier using stylistic features with the intention of making it as most language independent as possible. It is worth mentioning that a relabeling process named *Context dependent dynamic normalization* is used in order to take advantage of the social network syntactic elements.

The results obtained are shown in first place using a group of data provided by the *PAN2015* competition organizers; after that, a comparison between the present investigation and the other seven best proposals is made using the official results released by the organizers.

Tests in spanish, english, italian and dutch were performed; being italian the one that showed the best performance and english the one that had the lower.

Based on the obtained results, it is possible to conclude that it is very important to keep all the information that the social network can provide because the caracteristics that this texts show are very different of wide length texts, making necesary to find extra elements that can help to classify them in a better way.

Gracias a Maricela y Fernando por su dedicación y comprensión,
a Fernanda por darle el toque divertido a esas noches de trabajo,
a Andy por estar siempre presente y recordarme las cosas lindas de la vida :)
Gracias a mi Universidad y al Grupo de Ingeniería Lingüística
por brindarme los recursos y el apoyo necesario para la realización de esta tesis.

Esta investigación se realizó gracias al apoyo del proyecto *Caracterización de huellas textuales para análisis forense*; financiado por el CONACYT con la clave 215179 del Fondo Sectorial de Investigación para la Educación (SEP-CONACYT).

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

1.1. Planteamiento del problema

En un contexto general, la clasificación es definida por Manning y Schüetze como la tarea de asignarle a objetos de un universo dos o más clases o categorías [Manning and Schütze, 1999]. Esta definición puede ser aplicada a documentos textuales y se puede llevar a cabo de forma automática aplicando Aprendizaje Automático.

La Clasificación Automática de Texto es una tarea del Procesamiento del Lenguaje Natural encargada de generar un modelo capaz de predecir de forma automática a cuál de las clases existentes pertenece un nuevo texto; este modelo es creado a partir de un corpus etiquetado que contenga ejemplos de esas clases [Koppel et al., 2002].

Debido al incremento en la disponibilidad de documentos digitalizados y la necesidad de organizarlos, la Clasificación Automática de Texto ha tenido un gran impulso [Sebastiani, 2002]. Actualmente cuenta con varias aplicaciones como son: etiquetado, desambigüación de palabras, desambigüación de frases preposicionales, identificación de idioma, identificación de autor y, finalmente, perfilado de autor.

A diferencia de la identificación de autor, la cual tiene como objetivo predecir si un texto pertenece o no a un autor en específico, el perfilado de autor tiene como objetivo predecir si un texto pertenece o no a un grupo de autores que comparten ciertas características: género, edad, nivel educativo, región geográfica, etc..

En el área del perfilado de autor existen investigaciones centradas en textos literarios, documentales y ensayos. Estos géneros, aunque son notablemente diferentes, comparten dos características importantes: lenguaje estándar y longitud amplia (mayor a 250 palabras) [Peersman et al., 2011].

Existe una gran variedad de trabajos que han usado este tipo de documentos mostrando resultados bastante buenos; ejemplo de ello es el desarrollado por Koppel

et al., en el cual clasificaron por género (hombre o mujer) un corpus de textos formales con una precisión de aproximadamente el 80 % utilizando un modelo basado en un separador lineal [Koppel et al., 2002].

Doyle *et al.*, por su parte, trabajaron con una colección de de 495 ensayos de estudiantes pertenecientes al corpus *British Academic Written English (BAWE)*[1] [English, 2015] teniendo como objetivo la clasificación automática del género del autor (hombre o mujer) y reportando una precisión del 81 % [Doyle and Kešelj, 2005].

Para el estudio de estos textos de longitud amplia es común utilizar características de contenido que distingan a hombres y mujeres. Éstas han sido estudiadas en gran medida y muestran ser bastante eficientes en su labor de discernir entre el género del autor de un texto, tales como son el uso de adverbios (temporales, ordinales, de lugar, modo, etc.), estructura y contenido de los párrafos, repetición de ciertos grupos de palabras, etc. [Jones and Myhill, 2007].

Por otro lado, el creciente acceso a Internet ha dado pie a otro tipo de textos que poseen características propias y son difícilmente comparables con los textos mencionados anteriormente; esto se debe principalmente a la necesidad de comunicarse rápidamente y a la libertad que existe para publicar contenido sin revisión. En septiembre de 2015, Facebook reportó un promedio de mil millones de usuarios activos al día[2] [Facebook, 2015], mientras que Twitter, 320 millones[3] [Twitter, 2015]; esto es un ejemplo de la gran cantidad de personas que utiliza diariamente Internet.

Trabajos realizados sobre clasificación de texto procedente de Internet (textos cortos) se han enfocado en la predicción de género y grupo etario dentro de publicaciones en blogs y redes sociales [Goswami et al., 2009, Mukherjee and Liu, 2010, Peersman et al., 2011, Schler et al., 2006]. Algunas de las características comunes que tienen este tipo de publicaciones son las siguientes:

- Longitud corta (menor o igual a las 250 palabras)
- Vocabulario no estandarizado
- Libre uso de mayúsculas y de signos de puntuación para exaltar ideas
- Uso de emoticonos para mostrar emociones
- Uso de vocabulario propio del blog o la red social

La mayor parte de los trabajos de investigación realizados a la fecha dan soluciones dependientes de la lengua; por ejemplo Schler *et al.* utilizaron un corpus de Blogger[4]

[1] http://www2.warwick.ac.uk/fac/soc/al/research/collect/bawe/
[2] http://newsroom.fb.com/company-info/
[3] https://about.twitter.com/es/company
[4] http://blogger.com/

excluyendo explícitamente aquellas publicaciones dentro del blog que no estuvieran en inglés [Schler et al., 2006]. La presente propuesta contempla la clasificación por género y edad para textos cortos procedentes de redes sociales, sin importar el idioma en el que se encuentren escritos, buscando aprovechar la estructura y la información sintáctica que provee la red social.

1.2. Objetivos

1.2.1. Objetivo principal

Desarrollar un prototipo de *software* que permita la clasificación automática de textos cortos mediante un algoritmo de aprendizaje, haciendo uso de características relacionadas al género y grupo etario de una persona.

1.2.2. Objetivo secundario

Proveer de una herramienta independiente del idioma que pueda ser aplicada a corpus de diferentes redes sociales.

1.3. Hipótesis

Es posible desarrollar un algoritmo capaz de clasificar automáticamente textos cortos independientemente del idioma, aprovechando la estructura y la información sintáctica proporcionada por la red social.

1.4. Justificación

Al hacer una revisión de la investigación realizada en el área de la clasificación automática de textos cortos es posible concluir lo siguiente:

- Existen avances en un número limitado de lenguas, siendo prácticamente nula la investigación en español.
- La mayoría de los algoritmos son dependientes del idioma, por lo que resulta difícil la posibilidad de aplicarlos a otras lenguas.

- Actualmente no se están aprovechando los recursos que la propia red social puede proporcionar, tales como léxico predefinido y elementos propios de la red social.

Estas conclusiones motivan la realización de la presente investigación para así poder proveer de una herramienta que pueda ser aplicada en un gran número de lenguas sin tener que hacer modificaciones al algoritmo.

1.5. Alcances y limitaciones

Aunque se pretende desarrollar un algoritmo independiente de la lengua, sólo es aplicable a aquellas lenguas que generen su escritura a partir de un alfabeto; ejemplo de ésto son los siguientes:

- Alfabeto latino

  A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z
  a b c d e f g h i j k l m n ñ o p q r s t u v w x y z

- Alfabeto cirílico

  А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ь Ю Я
  а б в г д е ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ь ю я

Lo anterior se debe a que en escrituras ideográficas como el chino cada elemento (palabra) es una unidad y resulta complicado separar los elementos que la componen en unidades más pequeñas; tarea que la presente investigación realiza.

El presente estudio no pretende descubrir la tendencia de las personas a orientarse por una forma de escritura propia de los hombres o de las mujeres; por lo que parte de la premisa de que la forma en que una persona se comunica está totalmente definida por el género al que pertenece.

1.6. Estructura del documento

En el Capítulo 2 se presenta el marco teórico que sirve como base para la presente investigación. Dentro de este capítulo se explica de forma general lo que es el Aprendizaje Automático así como los diferentes tipos de Aprendizaje Automático que existen. Posteriormente, se explica lo que es el Procesamiento del Lenguaje Natural y cómo la clasificación y el agrupamiento de textos pertenecen a esta ciencia. Finalmente,

se explican las características utilizadas en la Clasificación Automática de Texto así como las características comúnmente utilizadas para identificar el género y la edad de las personas.

El Capítulo 3 presenta trabajos relacionados con esta investigación, como son el uso de n-gramas para el Procesamiento del Lenguaje Natural e investigaciones realizadas para la identificación automática de género y edad.

El Capitulo 4 se divide en dos grandes secciones: en la primera se explica a detalle cada una de las fases y sus etapas del algoritmo propuesto; en la segunda se presenta el protocolo experimental, se explica en dónde se aplicó el algoritmo propuesto y se describe el corpus utilizado. Finalmente, se da una descripción de la experimentación realizada durante la etapa de entrenamiento para el concurso *PAN2015*.

Los resultados obtenidos de la experimentación realizada se detallan en el Capítulo 5. En éste, se describe la medida de desempeño utilizada para evaluar el rendimiento de la presente propuesta, así como un análisis de resultados de los cuatro idiomas en donde fue aplicado (español, inglés, italiano y holandés).

En el Capítulo 6 se describen brevemente las otras siete mejores propuestas dentro del concurso *PAN2015* y se hace una comparación de resultados entre éstas y la presente propuesta.

El Capítulo 7 presenta las conclusiones así como las contribuciones y el trabajo futuro de este proyecto de investigación.

Capítulo 2

Marco teórico

2.1. Aprendizaje Automático

El Aprendizaje Automático o también llamado Aprendizaje de Máquina es una rama de la Inteligencia Artificial que tiene como objetivo resolver problemas computacionales que no pueden ser resueltos por un algoritmo estático. Un algoritmo que sigue la lógica del Aprendizaje Automático debe ser capaz de adaptarse o aprender de los datos de entrada para lograr generar una salida basada en esos datos [Alpaydin, 2010].

El término *aprender* es un tanto difícil de conceptualizar ya que en realidad no es posible igualar el término para un humano que para una computadora tal y como la conocemos en la actualidad, pero se puede describir de la siguiente forma para hacerlo funcional: mejorar automáticamente con la experiencia; o en palabras más precisas,

Un programa de computadora se dice que aprende de la experiencia *E* con respecto a una clase de tarea *T* y una media de rendimiento *P*, si su rendimiento en la tareas en *T*, medidas por *P*, mejoran con la experiencia *E* [Mitchell, 1997].

Formalmente, el Aprendizaje Automático es la programación de computadoras para optimizar un criterio de desempeño, usando datos de ejemplo o experiencia pasada [Alpaydin, 2010].

Las aplicaciones que se le pueden dar al Aprendizaje Automático son bastas. Imaginemos una librería en línea; los clientes de esta tienda generan una cuenta dentro de la cual se va almacenando todo su historial de compras. Una vez que el cliente hace su primera compra, el sistema muestra una serie de libros que, basándose en la información previa (primera compra) y en compras de otros clientes, le pueden ser de interés. Los libros mostrados se van modificando dependiendo de las siguientes compras que se hagan por parte del cliente. A los sistemas basados en esta forma de funcionamiento se les conoce como Sistemas de Recomendación, entre ellos se

encuentran Amazon[1], Netflix[2] y Mercado Libre[3].

Otro ejemplo del uso que se le puede dar al Aprendizaje Automático son los sistemas clasificadores de *spam* [Alpaydin, 2010]. Estos clasificadores tienen como tarea decidir si un nuevo correo es o no un correo basura (*spam*); para lograr su objetivo, el sistema debe llevar a cabo un proceso en el cual logre aprender los patrones que caracterizan tanto a un correo basura como a un correo legítimo [Metsis et al., 2006]. Hoy en día todos los sistemas de correo electrónico cuentan con un clasificador de *spam*.

Para catalogar los diferentes tipos de Aprendizaje Automático, se retoma una estructura propuesta por Russell *et al.* [Russell and Norvig, 2003] y posteriormente, retomada por Alpaydin [Alpaydin, 2010] al ejemplificar los diferentes usos que se le pueden dar al Aprendizaje Automático, obteniendo así cuatro grupos diferentes:

- Aprendizaje por asociación

  Este tipo de aprendizaje consiste en encontrar una regla de asociación dada una probabilidad condicional. En el sistema de recomendación descrito anteriormente (librería en línea) es posible plantear un ejemplo práctico; si ocho de cada diez clientes que ven la película A también ven la película B, es posible establecer una probabilidad $P(B|A) = 0.8$. De esta forma, si un nuevo cliente ve la película A es muy probable que tenga interés en ver la película B. Por lo cual se le hace la recomendación de B.

- Aprendizaje supervisado

  Este tipo de aprendizaje considera como premisa que los datos con los que se cuenta están etiquetados y, debido a esto, es posible crear una fase de entrenamiento y otra de prueba para medir qué tan eficiente es el algoritmo. A continuación se describen los dos tipos de problemas que es posible enfrentar haciendo uso del Aprendizaje supervisado.

  - Clasificación

    El objetivo de los problemas de clasificación es predecir a cuál o a cuáles clases pertenece una muestra que el sistema clasificador nunca ha visto. Para lograr esta tarea es necesario hacer uso de una regla de clasificación, misma que es generada a partir de muestras de entrenamiento de todas las clases existentes [Alpaydin, 2010].

    Dicho formalmente, sea el conjunto U conformado por los pares ordenados $(m_i, c_j) \in MXC$ en donde M es el conjunto de muestras de entrenamiento y C el conjunto de clases existentes. Existe una función ϕ optimizada por U denominada clasificador, capaz de asignar una clase $c_j \in C$ a una nueva muestra $n \notin M$.

[1] https://www.amazon.com.mx/
[2] http://www.netflix.com/
[3] http://www.mercadolibre.com.mx/

El clasificador de *spam* ya mencionado es un claro ejemplo de este tipo de aprendizaje pues tiene dos clases establecidas (*spam* y correo legítimo) (C). Haciendo uso de correos de entrenamiento (U) se crea una regla de clasificación (ϕ) que posteriormente se usará para decidir si un nuevo correo ($n \notin M$) es un correo basura (c_{spam}) o un correo legítimo ($c_{legítimo}$).

- Regresión

  Los problemas de regresión tienen como objetivo asignar un valor numérico a una muestra que el sistema regresor nunca ha visto.
  De forma similar a los sistemas clasificadores y debido a que es un algoritmo de Aprendizaje supervisado, es necesario contar con muestras de entrenamiento etiquetadas para optimizar un modelo matemático; la diferencia con los sistemas regresores es que no es necesario contar con muestras etiquetadas para todos los valores de salida posibles [Alpaydin, 2010].

  Formalizando; sea el conjunto U conformado por los pares ordenados $(m_i, \textit{número}) \in M$ en donde M es el conjunto de muestras de entrenamiento. Existe un modelo ε generado por U denominado regresor, capaz de asignar un valor continuo a una nueva muestra $n \notin M$. El modelo ε obtenido puede ser un modelo lineal, un modelo cuadrático, un polinomio de mayor o, incluso, modelos no lineales en caso de ser necesario.

  Modelo lineal: $y = w_1 \cdot x + w_0$
  Modelo cuadrático: $y = w_2 \cdot x^2 + w_1 \cdot x + w_0$
  Modelo cúbico: $y = w_3 \cdot x^3 + w_2 \cdot x^2 + w_1 \cdot x + w_0$

- Aprendizaje no supervisado

  El Aprendizaje no supervisado basa su funcionamiento en el descubrimiento de regularidades de las muestras de entrada, ya que éstas no se encuentran etiquetadas [Alpaydin, 2010]. A diferencia del Aprendizaje supervisado, en el Aprendizaje no supervisado no existe una etapa de entrenamiento ni un modelo o función a optimizar que dé como resultado una clase o un valor numérico.

  Un tipo de aprendizaje no supervisado es el *clustering* o agrupamiento. El objetivo del agrupamiento es identificar las características más representativas de las muestras de entrada para lograr agrupar a las muestras que tengan la mayor cantidad de elementos en común entre ellas.

  Es posible hacer uso del agrupamiento para una infinidad de aplicaciones, ya que sólo es necesario contar con una cantidad significativa de datos y seleccionar las características correctas para representar cada muestra de los datos.

  Para ejemplificar el agrupamiento, imaginemos una búsqueda $\beta = \textit{gato}$ en Google[4]. Es muy probable que el buscador regrese un conjunto de resultados D relacionados a los gatos domésticos (*Felis silvestris catus*) pero es posible que también regrese un conjunto de resultados E relacionados a la herramienta llamada *gato hidráulico*, teniendo así un conjunto de resultados

[4] https://www.google.com.mx/

$R = \{(\{D\} \in \textit{Felis silvestris catus}), (\{E\} \in \textit{gato hidráulico})\}$. Todos los elementos dentro de D, al igual que todos los elementos dentro de E deben de compartir características que pueden ser identificadas automáticamente y así lograr separar R en sus dos componentes D y E (Figura 2.1).

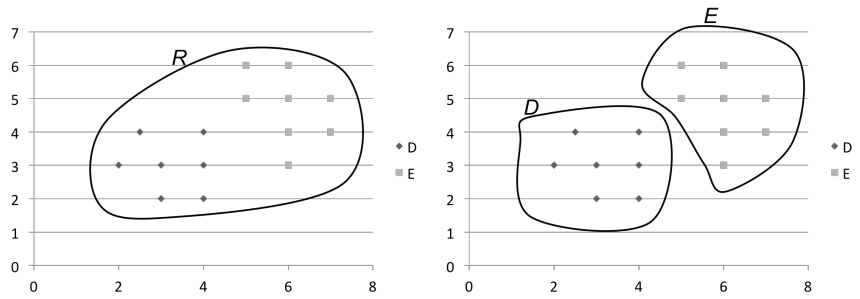

Figura 2.1: Ejemplo de Aprendizaje no supervisado (agrupamiento).

Es importante señalar que tanto para el Aprendizaje supervisado como para el Aprendizaje no supervisado, la selección correcta de características es de suma importancia, pues éstas son las que representarán cada muestra de los datos. En caso de hacer una selección incorrecta es posible que no sean las características que mejor representen los datos para el estudio realizado; dando como resultado un sistema mal entrenado en el caso del Aprendizaje supervisado y un agrupamiento erróneo en el caso del Aprendizaje no supervisado.

- Aprendizaje por reforzamiento

  El Aprendizaje por reforzamiento tiene como característica principal que su salida es una secuencia de acciones que trabajan en conjunto para alcanzar una meta. Por su parte, cada una de las acciones no es importante y sólo es una buena acción si ayuda al conjunto a alcanzar la meta. Normalmente, este tipo de aprendizaje sirve para juegos, en los cuales se debe seguir una secuencia de acciones para ganar [Alpaydin, 2010].

2.1.1. Ejemplos

A continuación, se muestran dos ejemplos de Aprendizaje automático: Bayes ingenuo y K-medias. El primero, se enfoca en la probabilidad condicional y en el Teorema de Bayes para generar un modelo de Aprendizaje supervisado. Bayes ingenuo es uno de los algoritmos clásicos de Aprendizaje supervisado ya que es bastante sencillo de implementar y permite conocer el comportamiento y la interacción de las características de las muestras.

En segundo ejemplo es un algoritmo de Aprendizaje no supervisado que separa las muestras en un número definido de grupos basándose en la distancia que existe entre cada una de las muestras que se van a agrupar.

2.1.1.1. Bayes ingenuo

Un ejemplo del aprendizaje supervisado es el clasificador Bayes Ingenuo. Éste se basa en la probabilidad condicional y en el Teorema de Bayes para separar las clases definidas y predecir la clase a la que pertenece una nueva muestra.

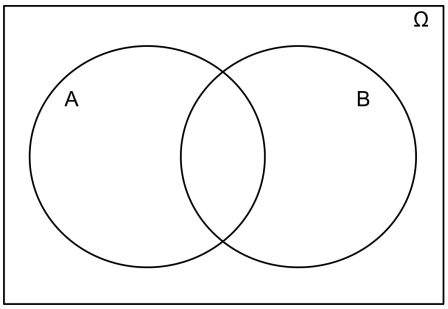

Figura 2.2: Probabilidad de eventos.

La probabilidad condicional permite saber la probabilidad de que ocurra un evento dato que otro evento ya ocurrió. En la Figura 2.2, la probabilidad condicional indica la probabilidad de que ocurra A dado que B ya ha ocurrido.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2.1}$$

De la misma forma, de la Figura 2.2 es posible obtener la probabilidad de que ocurra B dado que A ya ha ocurrido.

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \tag{2.2}$$

El Teorema de Bayes permite expresar la probabilidad condicional de que ocurra A dado que B ya ha ocurrido ($P(A|B)$) en términos de la probabilidad condicional de que ocurra B dado que A ya ha ocurrido ($P(B|A)$); para lograr esto, el Teorema de Bayes cosidera que $P(A|B) = P(B|A)$.

$$\begin{aligned}
P(A \cap B) &= P(A|B) \cdot P(B) && \text{de (2.1)} \\
P(B \cap A) &= P(B|A) \cdot P(A) && \text{de (2.2)} \\
\text{si } P(A|B) &= P(B|A) \\
P(A|B) \cdot P(B) &= P(B|A) \cdot P(A) \\
P(A|B) &= P(B|A) \cdot \frac{P(A)}{P(B)}
\end{aligned} \tag{2.3}$$

A partir de (2.3) y al renombrar variables se obtiene la siguiente ecuación:

$$P(h|D) = P(D|h) \cdot \frac{P(h)}{P(D)} \tag{2.4}$$

En donde $h \in H$ y H es el conjunto de clases definidas. $P(h|D)$ se denomina la probabilidad *a posteriori* de h ya que se refleja el hecho de que h se obtiene una vez que se han observado los datos de entrenamiento D [Mitchell, 1997].

Para este algoritmo es necesario calcular todas probabilidades *a posteriori* $P(h|D) \in H$ para obtener la hipótesis máxima *a posteriori* y así decidir la clase a la que pertenece el elemento.

$$h_{MAP} = \underset{h \in H}{\text{argmáx}}\, P(h|D) \tag{2.5}$$

2.1.1.2. K-medias

K-medias es un algoritmo de Aprendizaje no supervisado de tipo particional, el cual separa las muestras en K grupos especificados *a priori* conforme a una función de distancia. Cada uno de los *clusters* o grupos tiene un centroide que atrae a las muestras del grupo. El algoritmo es el siguiente:

1. Asignar de forma aleatoria la posición inicial de los K centroides dentro del dominio de los datos
2. Mientras no se cumpla el criterio de paro:
   a) Asociar a cada muestra del conjunto el centroide más cercano (haciendo uso de alguna función de distancia)
   b) Calcular los nuevos centroides a partir de la nueva distribución de las muestras asociadas a los centroides

- Criterios de paro
  - Se ha completado un número establecido de iteraciones
  - No existen cambios entre iteraciones en la asignación de las muestras a los grupos o *clusters*
  - La posición de los centroides no cambia entre iteraciones

Un ejemplo de la implementación de K-medias se muestra en la Figura 2.3; en un principio (A) todas las muestras U se encuentran en un mismo grupo ya que $C = \{\}$.

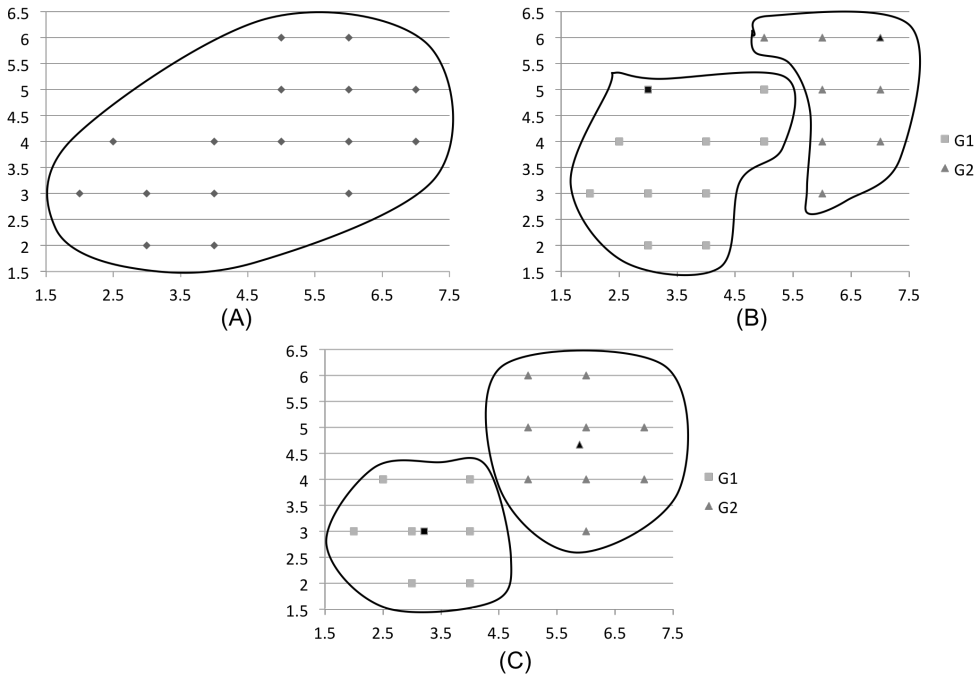

Figura 2.3: Ejemplo del algoritmo K-medias

Al colocar dos centroides de forma aleatoria ($C = \{G1, G2\}$) (representados por los elementos más oscuros), las muestras son asociadas a su centroide más cercano (B). Para la cuarta iteración (C), los documentos dejan de ser reasignados a los centroides finalizando así el algoritmo.

2.2. Procesamiento del Lenguaje Natural

La Lingüística es la ciencia que estudia todas las manifestaciones del lenguaje humano o lenguaje natural construyendo sus modelos y descripciones [Saussure, 2008, Gelbukh and Sidorov, 2006]. Estas tareas son tradicionalmente realizadas de forma manual teniendo ciertas limitantes prácticas como el procesamiento y análisis de grandes cantidades de texto, búsqueda de contraejemplos y la creación de estadísticas complejas [Gelbukh and Sidorov, 2006]. La Computación ha sido de gran utilidad para el desarrollo de la Lingüística, ya que la ha provisto de un universo de herramientas y enfoques que la convierten en una ciencia exacta [Gelbukh and Sidorov, 2006].

Esta intersección entre la Lingüística y la Computación ha dado pie a dos nuevas ciencias: la Lingüística Computacional y el Procesamiento Automático del Lenguaje Natural; este último, mejor conocido como Procesamiento del Lenguaje Natural o PLN debido a la influencia anglosajona. Por un lado, la Lingüística Computacional se enfoca en construir modelos del lenguaje formalizados para ser entendidos por las

computadora. Por otro lado, el PLN aborda la aplicación de dichos modelos. Ambas disciplinas comparten el mismo objetivo, por lo que es posible que en la práctica no exista diferencia entre una y otra [Gelbukh and Sidorov, 2006].

El PLN abarca diversas tareas como desambiguación de palabras, identificación de colocaciones, etiquetado gramatical, traducción automática, identificación de entidades nombradas, agrupamiento y clasificación de textos.

2.3. Clasificación y agrupamiento automático de textos

Como se mencionó anteriormente, el PLN contempla tareas como la clasificación y el agrupamiento de textos. Es común que estas tareas sean abordadas por medio de Aprendizaje Automático, siendo la clasificación una instancia del Aprendizaje supervisado y el agrupamiento una instancia del Aprendizaje no supervisado. Algunas aplicaciones de la clasificación y el agrupamiento automático de textos son las siguientes:

- Filtrado de *spam*
- Atribución de autoría
- Análisis de sentimientos
- Identificación de idioma
- Identificación de género literario
- Agrupamiento por temática
- Identificación de género del autor
- Identificación de grupo etario

2.3.1. Clasificación Automática de Texto

Retomando el concepto de clasificación dado anteriormente y aplicándolo al PLN; la Clasificación Automática de Texto hace referencia a la asignación de un valor $c_j \in C$ a una muestra $n \notin M$. Este valor es asignado por un clasificador ϕ, optimizado por el conjunto U que está conformado por los pares ordenados $(m_i, c_j) \in MXC$. Dependiendo de la naturaleza del problema, es posible utilizar una clasificación monoclase o una clasificación multiclase.

La clasificación monoclase es utilizada cuando la naturaleza de la tarea radica en decidir si los elementos a clasificar pertenecen o no a específicamente una clase. Por ejemplo: en el problema planteado del clasificador de *spam*, la tarea para el clasificador es determinar si el elemento analizado pertenece a la clase *spam* o a la clase *"no spam"*. Es posible ver esta decisión como una decisión binaria (1 si pertenece y 0 si no pertenece).

Hacer una clasificación monoclase puede no ser la solución en otro tipo de tareas. Imaginemos un escenario en donde se cuenta con un conjunto compuesto por textos en el dialecto del español peninsular y por textos en el español de México, algunos de estos textos están escritos en prosa y otros en verso.

$$U = \{(\{V\} \in Peninsular), (\{W\} \in Mexico), (\{X\} \in verso), (\{W\} \in prosa)\}$$

El objetivo de la tarea es asignarle a una nueva muestra $m \notin M$ uno de los siguientes pares de clases:

- Textos escritos en prosa en el español peninsular.

$$(c_{prosa}, c_{Peninsular})$$

- Textos escritos en verso en el español peninsular.

$$(c_{verso}, c_{Peninsular})$$

- Textos escritos en prosa en el español de México.

$$(c_{prosa}, c_{Mexico})$$

- Textos escritos en verso en el español de México.

$$(c_{verso}, c_{Mexico})$$

Como se muestra en los grupos a identificar, es necesario hacer una clasificación que involucre más de una clase por muestra analizada, si bien es cierto que la clasificación sigue siendo binaria (pertenece o no a cierta clase), se encuentran involucradas combinaciones de las clases existentes.

Para lograr este tipo de clasificaciones es necesario que las múltiples etiquetas asignadas a cada instancia sean estocásticamente independientes [Sebastiani, 2002]. Por ejemplo, un texto puede estar escrito en verso y al mismo tiempo en el español de México ya que la clase "verso" y la clase "México" son independientes entre sí. El caso contrario sería tener la clasificación siguiente: "texto escrito en prosa y verso";

esto no es posible ya que la clase “verso” y la clase “prosa” no son independientes entre ellas dentro del ejemplo.

El enfoque supervisado aprovecha el conocimiento *a priori* de las características del corpus como son la cantidad de clases y el etiquetado del mismo. Existe gran cantidad de algoritmos supervisados que pueden ser empleados para la Clasificación Automática de Texto; la decisión de tomar alguno en específico depende de su complejidad, funcionamiento y rendimiento. Para ejemplificar, a continuación se describe un algoritmo llamado Máquina de Vectores de Soporte. Dicho algoritmo es utilizado en gran medida actualmente y es el algoritmo utilizado en la presente investigación.

2.3.1.1. Máquina de Vectores de Soporte

La Máquina de Vectores de Soporte o SVM, por sus siglas en inglés (Support Vector Machine), es un algoritmo de aprendizaje con un gran fundamento matemático que simplifica los problemas de clasificación y regresión transformando los vectores de características de las muestras en una representación tal que puedan ser tratadas por un separador lineal.

La idea detrás de SVM es encontrar el hiperplano óptimo que separe las muestras de dos clases; éste se debe localizar a una distancia máxima entre las dos clases (margen) [Alpaydin, 2010, Cristianini and Shawe-Taylor, 2000]. Como se muestra en la Figura 2.4 (A), existe una cantidad infinita de hiperplanos que separan ambas clases pero es necesario utilizar aquel que minimice el error por *overfitting*. Para minimizar este error, SVM utiliza ciertos vectores de soporte que sirven para anclar el hiperplano óptimo (Figura 2.4 (B)).

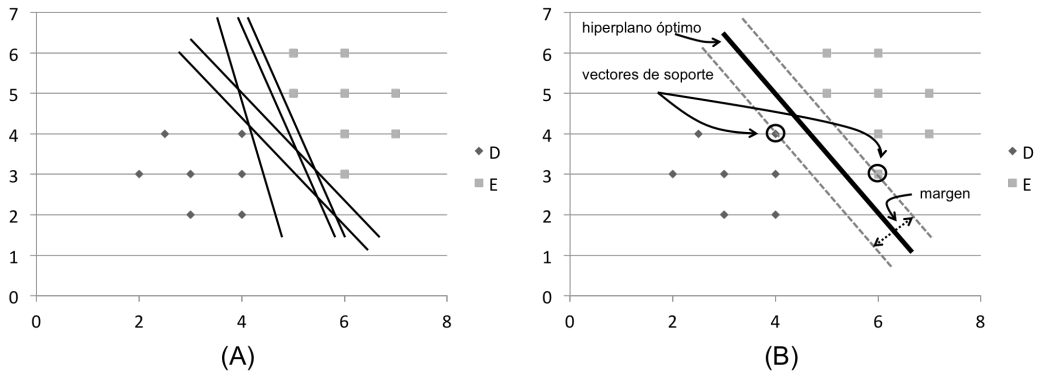

Figura 2.4: Funcionamiento de SVM

Las clases D y E de la Figura 2.4 corresponden a los valores -1 y +1 respectivamente; por lo que a una muestra $X = \{x^t, r^t\}$ se le debe asignar $r^t = -1$ si $x \in D$ y $r^t = +1$ si $x \in E$. De esta forma el objetivo de SVM es encontrar un vector $\vec{w}$ perpendicular al hiperplano óptimo y un escalar w_0 tal que

$$\vec{w}^T x^t + w_0 \geq +1 \quad \text{si} \quad r^t = +1$$
$$\vec{w}^T x^t + w_0 \leq -1 \quad \text{si} \quad r^t = -1$$

o, agrupando ambas ecuaciones [Alpaydin, 2010, Sra et al., 2012]

$$r^t(\vec{w}^T x^t + w_0) \geq +1 \tag{2.6}$$

El hiperplano óptimo está entonces definido por la función $\vec{w}^T x^t + w_0 = 0$ en donde $\vec{w}$ es un vector perpendicular al hiperplano y $\frac{|w_0|}{||\vec{w}||}$ es la distancia perpendicular del hiperplano al origen (Figura 2.5).

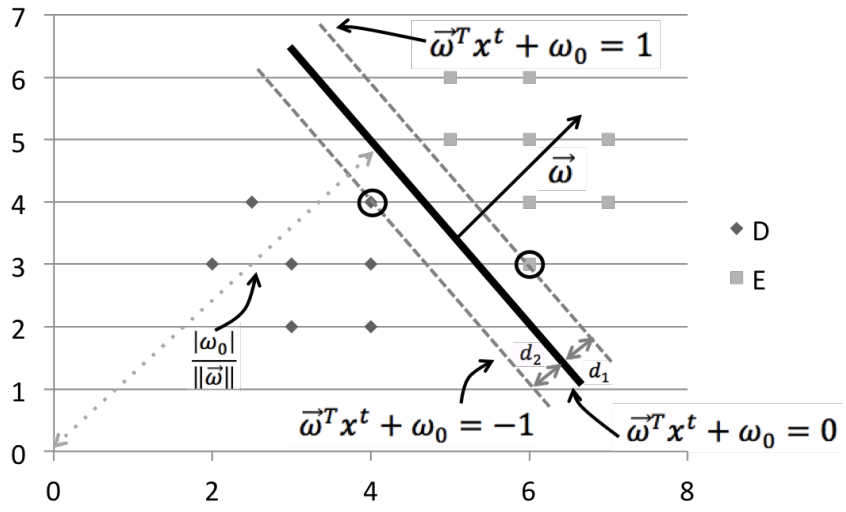

Figura 2.5: Obtención del hiperplano óptimo (SVM)

La distancia del hiperplano óptimo a cualquiera de los hiperplanos que componen el margen (d_1 y d_2) está dada por $\frac{1}{||\vec{w}||}$, por lo que la distancia a maximizar es $\frac{2}{||\vec{w}||}$ bajo la restricción impuesta por (2.6) [Sra et al., 2012].

$$max(\frac{2}{||\vec{w}||}) \quad \text{tal que} \quad r^t(\vec{w}^T x^t + w_0) \geq +1 \tag{2.7}$$

En caso de que no sea posible separar las clases por medio de un separador lineal, SVM mapea los vectores a otro espacio en donde sí sea posible. Este mapeo de espacios es realizado por medio de una función no lineal que se conoce como función Kernel. La gran ventaja de realizar esta transformación de espacios es que la noción básica sigue siendo la misma.

2.3.2. Agrupamiento Automático de Textos

Imaginemos ahora un escenario en donde se cuenta con un conjunto U de 5000 textos que tienen que ser agrupados por un conjunto C de temas y no es posible contar con personal que haga esta labor; tampoco se posee la información referente a las temáticas abordadas ($C = \{\}$). Bajo este esquema no es posible hacer uso de un enfoque supervisado por dos razones: primera, no hay clases definidas que puedan ser asignadas a los textos; segunda, al no existir clases definidas es imposible contar con los textos etiquetados. En este caso es necesario hacer uso del enfoque no supervisado.

El enfoque no supervisado del PLN tiene como objetivo descubrir las regularidades en las características de los textos de entrada. El *clustering* o agrupamiento automático de textos pertenece a este enfoque. Su finalidad es identificar las regularidades en las características de los textos de entrada para lograr agrupar aquellos textos que tengan la mayor cantidad de elementos comunes entre sí. La cantidad de grupos generados es indefinida debido a las siguientes razones:

- No se cuenta con la cantidad real de grupos
- Dependiendo del algoritmo a implementar:
  - Es posible establecer una cantidad finita de grupos
  - Es posible dejar que el algoritmo genere automáticamente la cantidad de grupos

A continuación, se presenta un algoritmo de agrupamiento llamado MAJORCLUST. En este algoritmo, propuesto por Stein *et al.* en [Stein and Niggemann, 1999], el número de grupos que son generados se descubre automáticamente.

2.3.2.1. MAJORCLUST

MAJORCLUST, al igual que K-medias, es un algoritmo de agrupamiento de tipo particional que se basa en la teoría de grafos para intuir de forma automática la cantidad de grupos a crear [Stein and Niggemann, 1999]. Para sintetizar la descripción del algoritmo MAJORCLUST, en la Figura 2.6 se muestra el algoritmo formal presentado en [Stein and Niggemann, 1999].

A continuación se presenta un ejemplo con seis documentos que hablan de "gato":

1. *el gato hidraulico es utilizado comunmente para levantar autos y objetos pesados utilizando muy poca fuerza*
2. *el gato hidraulicoes una maquina empleada para la elevacion de cargas mediante el accionamiento manual de una manivela o una palanca*

MAJORCLUST.
Input. A graph $G = \langle V, E \rangle$.
Output. A function $c : V \mapsto \mathbf{N}$, which assigns a cluster number to each node.

(1) $n = 0$, $t = \mathit{false}$
(2) $\forall v \in V$ **do** $n = n + 1$, $c(v) = n$ **end**
(3) **while** $t = \mathit{false}$ **do**
(4) $\quad t = \mathit{true}$
(5) $\quad \forall v \in V$ **do**
(6) $\quad\quad c^* = i$ **if** $\left|\{u : \{u, v\} \in E \wedge c(u) = i\}\right|$ is max.
(7) $\quad\quad$ **if** $c(v) \neq c^*$ **then** $c(v) = c^*, t = \mathit{false}$
(8) $\quad$ **end**
(9) **end**

Figura 2.6: Algoritmo MAJORCLUST

3. *el gato hidraulico es una maquina empleada para la elevacion de cargas que puede ser mecanica o hidraulica*

4. *el gato domestico sea cual sea su raza son todos miembros de una misma especie Felis catus*

5. *el gato domestico Felis silvestris catus es una subespecie de mamifero carnivoro de la familia Felidae*

6. *del latin vulgar cattus de origen incierto*

Como se puede apreciar, los documentos (1, 2 y 3) son relativos a la herramienta denominada "gato hidráulico" y los documentos (4, 5, 6) hacen referencia a un animal denominado "gato doméstico". A partir de Tf-Idf, MAJORCLUST obtiene los ángulos coseno entre cada documento; agrupando finalmente de la siguiente forma:

- Grupo 1
  - *el gato hidraulico es utilizado comunmente para levantar autos y objetos pesados utilizando muy poca fuerza*
  - *el gato hidraulicoes una maquina empleada para la elevacion de cargas mediante el accionamiento manual de una manivela o una palanca*
  - *el gato hidraulico es una maquina empleada para la elevacion de cargas que puede ser mecanica o hidraulica*
  - *del latin vulgar cattus1 de origen incierto*
- Grupo 2
  - *el gato domestico sea cual sea su raza son todos miembros de una misma especie Felis catus*
  - *el gato domestico Felis silvestris catus es una subespecie de mamifero carnivoro de la familia Felidae*

2.3.3. Características para la Clasificación Automática de Texto

Para la Clasificación Automática de Texto y para toda tarea del Aprendizaje Automático es necesario cuantificar y pasar a un espacio vectorial las características de los objetos de estudio. En el caso de la Clasificación Automática de Texto las características son el texto en sí, siendo posible separarlas en dos grupos: características de contenido y características estilísticas [Argamon et al., 2009].

Cada grupo logra extraer cierta información del texto y dependiendo de lo que se quiera clasificar y el tipo de texto que se tenga, existe la posibilidad de utilizar características de contenido, características estilísticas o una mezcla de ambas.

- Características de contenido

  Este grupo involucra aquellas características que proveen de información referente al contenido, significado o gramática del texto [Felice and Specia, 2012]. Las características que componen este grupo son las siguientes [Felice and Specia, 2012, Nguyen et al., 2011, Schler et al., 2006]:

  - Palabras de contenido
  - Categorías de palabras
  - Clases de palabras
  - Bolsa de palabras
  - Entidades nombradas

- Características estilísticas

  El objetivo de este grupo de características es extraer el estilo de escribir de las personas (estilometría) [Mukherjee and Liu, 2010]. Las características que componen este grupo son las siguientes [Mukherjee and Liu, 2010, Nguyen et al., 2011, Schler et al., 2006]:

  - Etiquetas gramaticales (POS)
  - Palabras funcionales
  - Léxico del contexto
  - Longitud de oraciones
  - Uso de emoticonos
  - Uso de coloquialismos
  - Uso de hipervínculos
  - Uso de signos de puntuación

2.3.4. Características lingüísticas de género y edad

2.3.4.1. Características lingüísticas de género

Investigaciones han demostrado que hombres y mujeres utilizan el lenguaje de forma diferente independientemente del tipo de lengua en cuestión
[Talbot, 2010, Holmes and Meyerhoff, 2008]. En el caso de la lengua escrita, Kanaris trabajó con textos redactados por niños y niñas de nivel primaria; concluyó que generalmente las mujeres escriben textos más largos y más complejos, así mismo utilizan una mayor cantidad de verbos y adjetivos diferentes. Por su parte, los niños reflejaron una tendencia a ser más orientados a eventos y a incluirse como el centro de atención en sus escritos, pues utilizan en mayor medida el pronombre singular en primera persona y, a diferencia de las niñas, no es común que centren la atención en otra persona u objeto [Kanaris, 1999].

En [Jones and Myhill, 2007], las autoras analizan cómo hombres y mujeres de nivel secundaria organizan los párrafos y qué tan cohesivas son las conexiones entre ellos; descubrieron importantes diferencias en aspectos como:

- Número de adverbios
- Número de veces que una palabra o frase es repetida
- Número que veces que se repite un sustantivo propio
- Número de sinónimos
- Número de hipónimos
- Número de anáforas
- Número de determinantes
- Número de oraciones
- Número de párrafos
- Número de oraciones por párrafo
- Número de palabras por oración
- Número de caracteres por palabra
- Número de oraciones pasivas
- Número de párrafos organizados por tema
- Número de párrafos que comienzan con una oración del tema en cuestión

- Número de párrafos que necesitan ser divididos

En el caso de los hombres, sus párrafos mostraron tener una mejor organización que los de las mujeres, contar con una oración de tema principal y ser más largos. En cambio, los párrafos de las mujeres manejaban de mejor forma los temas y eran poco propensos a cambiar de tema y necesitar ser divididos.

En [Newman et al., 2008], con la finalidad de generalizar en la medida de lo posible su investigación, crearon un corpus recolectado a partir de 70 estudios diferentes consiguiendo así un aproximado de 46 millones de palabras. Dentro de los resultados, reportan que las mujeres usan más palabras relacionadas con procesos psicológicos y sociales, así como mayor cantidad de verbos. Por su parte, los hombres hablan más sobre las propiedades de los objetos, prefieren temas más impersonales, sus palabras son de una longitud mayor y utilizan más artículos, números y preposiciones.

2.3.5. Características lingüísticas de edad

En relación con la edad, diversas investigaciones han mostrado que la forma de utilizar la lengua varía dependiendo de ésta; esta variación se ve reflejada en elementos que brindan contenido al texto. Estudios reportan que, al incrementar la edad, las personas utilizan más palabras de sentimiento con connotación positiva y más tiempos futuros, a diferencia de la gente joven que utiliza más palabras de sentimiento con connotación negativa, más palabras que hacen referencia a ellos mismos, más tiempos pasados y en general, un vocabulario más sencillo [Pennebaker et al., 2003, Pennebaker and Stone, 2003].

Por su parte, Schler *et al.*, comparan el uso del vocabulario utilizado por tres grupos de edades: 10s, 20s y 30s obteniendo las palabras representativas mostradas en la Tabla 2.1 y los grupos de palabras mostrados en la Tabla 2.2 [Schler et al., 2006]. Finalmente, reportan haber observado un comportamiento similar al reportado por [Pennebaker et al., 2003, Pennebaker and Stone, 2003].

Tabla 2.1: Palabras por grupo de edad

10s	*20s*	*30s*
maths	semester	marriage
homework	apartment	development
bored	drunk	campaign
sis	beer	tax
boring	student	local
awesome	album	democratic
mum	college	son
crappy	someday	systems
mad	dating	provide
dumb	bar	workers

Tabla 2.2: Grupos de palabras por grupo de edad

10s	*20s*	*30s*
sports	tv	money
sleep	eating	job
sex		family
friends		
positive emotions		
negative emotions		

Nguyen *et al.* logran diferenciar entre grupos de palabras y palabras individuales de gente joven y gente adulta [Nguyen et al., 2011], (Tabla 2.3).

Tabla 2.3: Grupos de palabras y palabras individuales por edad

jóvenes	*adultos*
like	years
gender-male	Inclusive (grupo)
School (grupo)	granddaughter
just	grandchildren
Anger (grupo)	had
Causation (grupo)	daughter
mom	grandson
so	ah
definitely	
Negative Emotions (grupo)	

Capítulo 3

Trabajos relacionados

3.1. Uso de n-gramas para el PLN

Los n-gramas son un recurso bastante utilizado en el PLN, ya que permiten extraer del texto características de contenido y estilísticas que pueden ser utilizadas para diversas tareas como son: traducción automática, resumen automático y clasificación de texto.

Los n-gramas son series de la unidad de información textual seleccionada. Existen diferentes tipos de n-gramas. Es posible decidir la unidad de información textual que se desee dependiendo de la tarea a realizar y el tipo de de información que se desee extraer. En traducción automática es común utilizar n-gramas de palabras o incluso n-gramas de oraciones para crear lo que se conoce como Modelo del Lenguaje [Koehn, 2010]. Por otro lado, para tareas de resumen automático son utilizados n-gramas de palabras [Fernández et al., 2007, Giannakopoulos et al., 2008]. Dentro de la clasificación de texto, en tareas como detección de plagio, identificación de autor y perfilado de autor, diferentes tipos de n-gramas son utilizados: n-gramas de palabras, n-gramas de caracteres y n-gramas POS [González-Gallardo et al., 2015, Stamatatos et al., 2015].

3.2. Identificación de género y edad de una persona

El objetivo de la presente investigación se encuentra completamente relacionado con la identificación de género y edad haciendo uso del PLN; debido a esto, a continuación se presenta una síntesis de algunos trabajos relacionados con la misma. Como se podrá apreciar, es común encontrar investigaciones que aborden ambas tareas obteniendo resultados, en su mayoría, con una precisión arriba del 80 %.

Koppel *et al.* realizaron un estudio con un conjunto de 566 documentos del *British National Corpus (BNC)*[1] [Corpus, 2015]; corpus de 100 millones de palabras compuesto por muestras transcritas en inglés provenientes de diversas fuentes. El estudio consistió en identificar de forma automática el género del autor del texto, siendo las palabras funcionales y los n-gramas POS las características utilizadas. Finalmente los autores reportan haber obtenido una precisión del 80 % utilizando un modelo basado en un separador lineal [Koppel et al., 2002].

Doyle *et al.* se enfocaron en una colección de de 495 ensayos de estudiantes pertenecientes al corpus *BAWE*[2] [English, 2015]. Su sistema clasificador consistió en la construcción de dos perfiles (uno para hombres y otro para mujeres) compuestos por n-gramas de caracteres, n-gramas de palabras y n-gramas POS; la asignación de una nueva muestra a una de las dos clases se obtiene a partir de la distancia entre los n-gramas de la muestra y los n-gramas de cada perfil. El mejor resultado reportado obtuvo una precisión del 81 % [Doyle and Kešelj, 2005].

En [Schler et al., 2006], se analizaron entradas pertenecientes al sitio web Blogger[3] juntando alrededor de 71 mil 500 muestras. Para la clasificación de edad, optaron por separar el corpus en tres clases diferentes: *10s* (13-17 años), *20s* (23-27 años) y *30s* (33-42 años); dejando fuera los rangos intermedios y así hacer más clara la diferenciación de edades. Las características seleccionadas para representar los textos fueron una combinación de características de contenido (palabras de contenido y clases de palabras) y estilísticas (POS, palabras funcionales e hipervínculos), resultando estas últimas más informativas para la predicción de género. Los resultados reportados son similares a [Koppel et al., 2002] con un 80.1 % de precisión al clasificar género; en el caso de la clasificación de edad, las características que más información aportaron fueron las de contenido, llegando así a una precisión del 76.2 %.

El corpus creado en [Schler et al., 2006] fue reutilizado por Goswami *et al.* en donde utilizaron como características para representar los documentos palabras fuera del diccionario que son comunes en las entradas de blogs y la longitud promedio de las oraciones. Las clases creadas para la clasificación de edad fueron las mismas que en [Schler et al., 2006]. Utilizando un clasificador Bayes Ingenuo reportan una precisión del 89.3 % para género y 80.32 % para edad [Goswami et al., 2009].

Rao *et al.* se enfocaron en la detección de género y edad en usuarios de Twitter utilizando n-gramas de palabras y características sociolingüísticas (Tabla 3.1). La clasificación de edad decidieron abordarla como una clasificación binaria con únicamente dos clases: *+30* y *-30*. Haciendo uso de una Máquina de Vectores de Soporte, reportan haber obtenido una precisión del 72.33 % para la clasificación de género y 74.11 % para edad [Rao et al., 2010].

Por su parte, Mukherjee *et al.* trabajaron con un conjunto de 3100 blogs per-

[1] http://www.natcorp.ox.ac.uk/

[2] http://www2.warwick.ac.uk/fac/soc/al/research/collect/bawe/

[3] http://blogger.com/

Tabla 3.1: Grupos de palabras y palabras individuales por edad

característica	descripción/ejemplo
SIMLEYS	Lista de *emoticons* obtenidos de la Wikipedia
OMG	Abreviación de 'Oh My God'
ELLIPSES	'...'
POSSESIVE BIGRAMS	Ej: my_XXX, our_XXX
REPATED ALPHABETS	Ej: niceeeeee, nooooo waaay
SELF	Ej: I_xxx, Im_xxx
LAUGH	Ej: LOL, ROTFL, LMFAO, haha, hehe
SHOUT	Texto en MAYÚSCULAS
EXASPERATION	Ej: Ugh, mmmm, hmmm, ahh, grrr
AGREEMENT	Ej: yea, yeah, ohya
HONORIFICS	Ej: dude, man, bro, sir
AFFECTION	Ej: xoxo
EXCITEMENT	Una cadena de simbolos de exclamación (!!!!!)
SINGLE EXCLAIM	Una sola exclamación al final del tuit
PUZZLED PUNCT	Una combinación de cualquier número de '?' y '!'

tenecientes a diferentes sitios de Internet utilizando características estilísticas y de contenido (clases de palabras, f-measure y secuencias POS) para crear un selector automático de características (EFS). Los mejores resultados reportados, con una precisión del 88.56 %, combinaban EFS y una Máquina de Vectores de Soporte aplicada a una regresión [Mukherjee and Liu, 2010].

En [Zhang and Zhang, 2010], Zhang *et al.* utilizaron el corpus creado por [Mukherjee and Liu, 2010]. Las características utilizadas para representar el texto fueron palabras, signos de puntuación, longitud promedio de palabras y oraciones, etiquetas gramaticales y grupos de palabras. Realizaron pruebas con Bayes Ingenuo y con Máquinas de Vectores de Soporte, siendo este segundo clasificador el que mejores resultados mostró con una precisión del 72.1 %.

Peersman *et al.* crearon un corpus de 1.5 millones de muestras obtenido de la red social Netlog[4]. Cuatro clases fueron creadas para la clasificación de edad: *min16* (11 a 15 años), *plus16* (16 años en adelante), *plus18* (18 años en adelante) y *plus25* (25 años en adelante). El mejor resultado obtenido para la clasificación conjunta de edad y género *min16_male VS. min16_female VS. plus25_male VS. plus25_female* fue una precisión del 64.2 % se alcanzó mediante una Máquina de Vectores de Soporte y utilizando 1-gramas, 2-gramas y 3-gramas de palabras, así como 2-gramas, 3-gramas y 4-gramas de caracteres como características [Peersman et al., 2011].

Cheng *et al.* trabajaron con un corpus de 6769 notas publicadas por *reuters.com*

[4]https://www.twoo.com/

y con un corpus de 8970 correos electrónico utilizando características de contenido como grupos de palabras y palabras de contenido, así como diferentes características estilísticas. Para el primer corpus, reportan una precisión del 76.75 % y una precisión del 82.23 % para el segundo; ambos resultados fueron obtenidos utilizando una Máquina de Vectores de Soporte [Cheng et al., 2011].

Capítulo 4

Clasificación Automática de Texto

Si bien es posible intentar predecir el género de una persona con un algoritmo de regresión y tratar a la variable "género" como una variable continua para finalmente dar como resultado un valor real entre $[-1, +1]$ en donde -1 significa *mujer*, $+1$ significa *hombre* y los valores intermedios significan la tendencia a ser uno u otro; se optó por un algoritmo de clasificación sobre un algoritmo de regresión, debido a la premisa de que la forma en que una persona se comunica está totalmente definida por el género al que pertenece. La variable "género" se tomó como una variable discreta con únicamente dos valores excluyentes: *hombre* y *mujer*.

La predicción del grupo etario se abordó de igual forma como un problema de clasificación al definir grupos de edades posibles; las características de cada uno de los grupos son las siguientes:

- Cada grupo está compuesto por un rango de edades adyacentes, ya sean continuas o discretas
- Los grupos deben ser excluyentes entre sí

Este tipo de problemas en donde se tienen que predecir dos aspectos de un elemento o entidad pertenecen a un problema multiclase (discutido con aterioridad en el Capítulo 2) y es posible abordarlo de diferentes formas.

Una posibilidad es crear dos clasificadores (uno para género y otro para grupo etario), de esta forma el elemento al cuál se le va a predecir ambos aspectos se ingresa, por ejemplo, primeramente al clasificador de género y, posteriormente, al clasificador de grupo etario .Esto conlleva las siguientes ventajas y desventajas:

- Ventajas
    - Independencia de las tareas; permite clasificar únicamente por género o por grupo etario.

- Es posible establecer diferentes algoritmos de aprendizaje para cada clasificador.
- A partir del resultado arrojado por el clasificador de género es posible configurar el clasificador de grupo etario para que se adapte al género predicho.
- Independencia de las características para representar género y grupo etario.

- Desventajas
  - Al crear dos clasificadores se tiene que hacer el entrenamiento de dos algoritmos de aprendizaje; y en caso de contar con diferentes características para el grupo etario de los hombres y de las mujeres, es necesario entrenar tres algoritmos de aprendizaje: el primero, para la clasificación de género; el segundo, para la clasificación del grupo etario de los hombres; y el tercero, para la clasificación del grupo etario de las mujeres.
  - Si la clasificación por género es incorrecta y se cuenta con diferentes características para el grupo etario de los hombres y de las mujeres, es más probable que la clasificación por grupo etario resulte errónea.

Otra posibilidad es generar únicamente un clasificador que abarque tanto la clasificación de género como la clasificación de grupo etario; para lograr esto es necesario hacer la combinación entre las clases pertenecientes a cada clasificación. Supongamos que para la clasificación de género se cuenta con dos clases (*hombre* y *mujer*) y para la clasificación por grupo etario se cuenta con tres clases $\{$*20s* $\rightarrow [20, 30)$ años, *30s* $\rightarrow [30, 40)$ años, *40s* $\rightarrow [40, +\infty)$ años$\}$, las clases resultantes serían las siguientes:

- *hombre_20s*: Hombres con edad mayor o igual a 20 años pero menor a 30 años.
- *hombre_30s*: Hombres con edad mayor o igual a 30 años pero menor a 40 años.
- *hombre_40s*: Hombres con edad mayor o igual a 40 años.
- *mujer_20s*: Mujeres con edad mayor o igual a 20 años pero menor a 30 años.
- *mujer_30s*: Mujeres con edad mayor o igual a 30 años pero menor a 40 años.
- *mujer_40s*: Mujeres con edad mayor o igual a 40 años.

De la misma forma que con el esquema de crear dos clasificadores, al crear uno solo existen ventajas y desventajas que deben ser consideradas.

- Ventajas
  - Únicamente es necesario hacer el entrenamiento de un algoritmo de aprendizaje, lo que implica menor tiempo y menor uso de recursos de la computadora
  - Es posible identificar con más facilidad las diferencias existentes entre las clases de la clasificación conjunta. Por ejemplo: al analizar la matriz de confusión de los resultados es posible apreciar qué tan parecido escriben los "Hombres con edad mayor o igual a 20 años pero menor a 30 años" y las "Mujeres con edad mayor o igual a 20 años pero menor a 30 años".
- Desventajas
  - El corpus de entrenamiento debe estar etiquetado de tal forma que de cada muestra sea posible conocer ambas clasificaciones: el género al que pertenece y el grupo etario del que forma parte.
  - Es necesario contar con un corpus medianamente balanceado, ya que al combinar las clases es posible que alguna o algunas queden con muy pocas muestras. Por ejemplo: si se cuenta con 50 muestras de la clase *20s* pero 40 de esas muestras también están etiquetadas como *hombre*, la distribución quedaría como se muestra en la Tabla 4.1.

Tabla 4.1: Distribución no balanceada de muestras.

clases	*hombre*	*mujer*
20s	40	10
porcentaje	80 %	20 %

Esta distribución de las muestras implica que el algoritmo de aprendizaje se entrenará con bastantes datos de *hombre_20s* y con pocos datos de *mujer_20s*, haciendo que el algoritmo de aprendizaje sea bueno identificando "Hombres con edad mayor o igual a 20 años pero menor a 30 años" y malo identificando "Mujeres con edad mayor o igual a 20 años pero menor a 30 años" ya que no tuvo una muestra representativa de esa clase.

La presente investigación hace uso de la segunda aproximación para así crear un solo clasificador; esto se debe al interés que se tiene en analizar las diferencias entre las clases de la clasificación conjunta.

4.1. Algoritmo propuesto

Para lograr una descripción del algoritmo lo más general posible, no se hará mención de ninguno de los dos elementos a clasificar (género y grupo etario) y, por consiguiente, de ninguna de sus clases.

Al optar por la creación de un solo clasificador es necesario que el corpus esté etiquetado de tal forma que de cada muestra sea posible conocer ambas clasificaciones. Supongamos que el primer aspecto a clasificar A tiene definidas dos clases: U, V

$$A = \{U, V\}$$

y el segundo aspecto B tiene definidas cuatro clases: W, X, Y, Z

$$B = \{W, X, Y, Z\}$$

al combinar ambos aspectos, las ocho clases resultantes son las siguientes:

$$A \cup B = \{UW, UX, UY, UZ, VW, VX, VY, VZ\}$$

Cabe mencionar la independencia en el orden de los aspectos, dando la libertad de combinarlos de la siguiente forma sin repercutir en los resultados:

$$B \cup A = \{WU, WV, XU, XV, YU, YV, ZU, ZV\}$$

El algoritmo, al ser un clasificador de texto, sigue un enfoque supervisado por lo que consta de dos fases: la fase de entrenamiento y la fase de prueba. La fase de entrenamiento tiene como finalidad entrenar al sistema con muestras del conjunto S, conformado por los pares ordenados $(m_i\, c_j) \in MXC$ en donde M es el conjunto de muestras de entrenamiento y C el conjunto de clases $A \cup B$. Durante la fase de prueba el objetivo es lograr predecir de forma automática la clase $c \in C$ de una nueva muestra $n \notin M$.

4.1.1. Fase de entrenamiento

La fase de entrenamiento se encuentra dividida en cinco etapas, cada una encargada de una tarea en específico con la finalidad de que sea modular y de que se pueda modificar cada etapa sin alterar el resto de ellas: Extracción, Etiquetado, Generación de POS, Creación de vectores y Entrenamiento.

Figura 4.1: Etapas de la fase de entrenamiento

4.1.1.1. Extracción

En esta etapa, el corpus es dividido en cada una de las ocho clases existentes, generando los ocho *archivos de clase* siguientes:

$$archivos\_de\_clase = \{UW_{txt}, UX_{txt}, UY_{txt}, UZ_{txt}, VW_{txt}, VX_{txt}, VY_{txt}, VZ_{txt}\}$$

4.1.1.2. Etiquetado

Durante el etiquetado, cada *archivo de clase* creado durante la etapa de extracción es enviado a Freeling[1] [Padró and Stanilovsky, 2012] para ser analizado y etiquetado gramaticalmente (etiquetado POS). Freeling regresa una cadena de texto en formato JSON[2], la cual es almacenada, obteniendo los ocho *archivos JSON* siguientes:

$$archivos\_JSON = \{UW_{json}, UX_{json}, UY_{json}, UZ_{json}, VW_{json}, VX_{json}, VY_{json}, VZ_{json}\}$$

4.1.1.3. Generación de POS

Una vez que se tienen los *archivos JSON*, estos son procesados para crear ocho *archivos POS* que cuenten con la misma estructura que los archivos de clase:

$$archivos\_POS = \{UW_{POS}, UX_{POS}, UY_{POS}, UZ_{POS}, VW_{POS}, VX_{POS}, VY_{POS}, VZ_{POS}\}$$

Es necesario re etiquetar ciertos *tokens* de la cadena JSON con etiquetas dependientes del corpus en donde se esté aplicando el algoritmo para buscar maximizar su rendimiento; a este proceso se le denomina 'normalización dinámica dependiente del contexto'.

- Normalización dinámica dependiente del contexto

  Freeling logra extraer la información gramatical dentro del texto, proveyendo así de las estructuras gramaticales que son utilizadas con mayor frecuencia y la forma en que las mismas son utilizadas. El problema es que existen estructuras léxicas dentro de las redes sociales que Freeling no es capaz de comprender, por lo que el etiquetado gramatical de estas estructuras es erróneo. Debido a esto, es necesario realizar una normalización dependiente de la red social en cuestión.

  Para ejemplificar esto imaginemos un escenario en el cual se está trabajando con un corpus de Facebook llamado *Comentarios de la Ciudad de México en el tiempo*[3] y se tienen los siguientes comentarios:

  - Sección etiquetada como *masculino*
    - Suena interesante **@Lena** polii !!
    - **@Alejandro Campos**. ¿dirás 1554? Saludos.
    - how it's happnd ? pls write to me in english **@mera**

[1] http://nlp.lsi.upc.edu/freeling/

[2] Freeling es llamado utilizando una interfaz que convierte su salida en una cadena JSON. Esta interfaz fue desarrollada por el Grupo de Ingeniería Lingüística, Instituto de Ingeniería, UNAM.

[3] *Comentarios de la Ciudad de México en el tiempo (Sección clasificada por género, femenino y masculino)* es un corpus creado por el Grupo de Ingeniería Lingüística.

    - **@Enrique Gonzalez Gomar** ¿Donde puedo consultar esa información que mencionas?
    - ya vieron **@lahijadelosapaches**
- Sección etiquetada como *femenino*
    - **@Raul Garcia** .... ¿cómo se organizaban para las labores de recuperación de cuerpos?
    - **@javier** campuzano **@jorgesgvn**
    - Mira **@omar** Zamudio
    - **@Jesus Mejia Perez** Dinos que métodos, por favor!

En ambas secciones se observa el patrón *@texto*, este patrón es utilizado dentro de Facebook para etiquetar a un usuario de la red social teniendo como finalidad llamar su atención y mencionarlo dentro del comentario. Al ingresar cada una de estas cadenas al etiquetador POS, la sección correspondiente al patrón *@texto* es etiquetada de la siguiente forma: *[...{"lemma": "@", "token": "@", "tag": "Fz", "prob": "1"}, {"lemma": "texto", "token": "texto", "tag": "XXXX", "prob": "N"}...]*. Al igualar la estructura a la de los *archivos de clase*, las siguientes cadenas son generadas:

- Sección etiquetada como *masculino*
    - VMIP3S0 AQ0CS0 **Fz NP00000** NCFS000 Fat Fat
    - **Fz NP00000** Fp Fia VMIF2S0 Z Fit NP00000 Fp
    - RG VMIP3S0 NCMS000 Fit RG VMIP3S0 RG PP1CS000 AQ0CS0 NCMS000 **Fz AQ0FS0**
    - **Fz NP00000** Fia PR000000 VMIP1S0 VMN0000 DD0FS0 NCFS000 PR0CN000 VMIP2S0 Fit
    - RG VMIS3P0 **Fz NCFP000**
- Sección etiquetada como *femenino*
    - **Fz NP00000** Fz Fia PT000000 P00CN000 VMII3P0 SPS00 DA0FP0 NCFP000 SPS00 NCFS000 SPS00 NCMP000 Fit
    - **Fz VMN0000** NCMS000 **Fz NCFS000**
    - NP00000 **Fz VMN0000** NP00000
    - **Fz NP00000** CS NCMP000 Fc RG Fat

Como se puede apreciar, al hacer el etiquetado gramatical mediante Freeling, la información que las estructuras sintácticas de Facebook aporta se pierde. Las entidades que originalmente eran unidades de Facebook, debido al etiquetado gramatical son separados y sus elementos son etiquetados de forma independiente. Debido a esto, es necesario hacer un re-etiquetado del patrón *@texto* con el fin de mantener esa unidad; bajo esta premisa, el patrón *@texto* debe ser etiquetado como *REF#USER*; obteniendo las siguientes cadenas:

- Sección etiquetada como *masculino*

    - VMIP3S0 AQ0CS0 **REF#USER** NCFS000 Fat Fat
    - **REF#USER** Fp Fia VMIF2S0 Z Fit NP00000 Fp
    - RG VMIP3S0 NCMS000 Fit RG VMIP3S0 RG PP1CS000 AQ0CS0 NCMS000 **REF#USER**
    - **REF#USER** Fia PR000000 VMIP1S0 VMN0000 DD0FS0 NCFS000 PR0CN000 VMIP2S0 Fit
    - RG VMIS3P0 **REF#USER**
- Sección etiquetada como *femenino*
    - **REF#USER** Fz Fia PT000000 P00CN000 VMII3P0 SPS00 DA0FP0 NCFP000 SPS00 NCFS000 SPS00 NCMP000 Fit
    - **REF#USER** NCMS000 **REF#USER**
    - NP00000 **REF#USER** NP00000
    - **REF#USER** CS NCMP000 Fc RG Fat

Observemos el siguiente comentario etiquetado como *masculino*:

Frase:

Suena interesante **@Lena** polii !!

Etiquetado POS (Freeling):

VMIP3S0 AQ0CS0 **Fz NP00000** NCFS000 Fat Fat

Re-etiquetado (etiquetas dependientes del corpus):

VMIP3S0 AQ0CS0 **REF#USER** NCFS000 Fat Fat

Una vez hecho el análisis gramatical mediante Freeling, la frase *"Suena interesante @Lena polii !!"* es etiquetada de la siguiente forma: VMIP3S0 **(verbo principal indicativo presente de la tercera persona del singular)** AQ0CS0 **(adjetivo calificativo común singular)** Fz **(puntuación)** NP00000 **(nombre propio)** NCFS000 **(nombre común femenino singular)** Fat **(puntuación)** Fat**(puntuación)**.

Para el caso de *"@Lena"*, Freeling hace un etiquetado correcto indicando que el elemento *"@"* corresponde a un signo de puntuación y *"Lena"* a un nombre propio. El inconveniente al hacer esta separación, que para fines gramaticales es correcta, es que la información que puede ser proporcionada por la estructura sintáctica de Facebook se pierde totalmente dejando fuera un elemento muy importante en la comunicación y haciendo imposible saber qué tanto es utilizado y la forma en que interactúa con los otros elementos dentro del comentario.

Para contrarrestar el problema antes mencionado, el algoritmo debe encontrar una forma de reconocer que cualquier serie de caracteres que siga (sin espacio) a

un símbolo de "@" debe ser re-etiquetado de tal forma que sea posible mantener la información gramatical. En este ejemplo, el re-etiquetado se puede hacer por dos métodos distintos.

El primer método es ir analizando cada elemento de la cadena *"VMIP3S0 AQ0CS0* ***Fz NP00000*** *NCFS000 Fat Fat"* y al encontrar la etiqueta *"FZ"* verificar si la siguiente etiqueta corresponde a un nombre. En caso de que esto se cumpla, intercambiar las etiquetas *"Fz"* y *"NP00000"* por una que haga referencia al etiquetado de un usuario dentro de la red social: *"REF#USER"*. Haciendo esta sustitución se obtiene la siguiente cadena: *"VMIP3S0 AQ0CS0* ***REF#USER*** *NCFS000 Fat Fat"*; ésta parece ser una solución óptima pero tiene un problema con comentarios que gramaticalmente compartan la misma secuencia que el ejemplo expuesto. A continuación, se presenta un ejemplo de un comentario etiquetado como *mujer* del caso ya mencionado:

Frase:

Hector miraaaaa **=D**

Etiquetado POS (Freeling):

NP00000 VMIP3S0 **Fz NP00000**

Re-etiquetado (etiquetas dependientes del corpus):

NP00000 VMIP3S0 **REF#USER**

Al emplear el primer método de re-etiquetado dado, se puede observar que la etiqueta *REF#USER* no corresponde a una referencia a otro usuario dentro de la red social, generando así información gramatical falsa del comentario de Facebook.

El segundo método consiste en analizar la frase en vez de la cadena etiquetada por Freeling. De esta forma, al ir avanzando por la frase *"Suena interesante* ***@Lena*** *polii !!"* y al encontrarse con el símbolo *"@"* al inicio de una palabra (*token*), el método verifica la etiqueta gramatical del siguiente elemento y en caso de no pertenecer al grupo de los signos de puntuación (*Fx*) se genera la etiqueta dependiente del corpus *"REF#USER"*, obteniedo finalmete la siguiente cadena de etiquetas: *"VMIP3S0 AQ0CS0* ***REF#USER*** *NCFS000 Fat Fat"*.

Si la frase *"Hector miraaaaa =D"* es analizada por este método, la cadena de elementos gramaticales generada por Freeling y la cadena POS creada en el re-etiquetado será exactamente la misma: *"NP00000 VMIP3S0* ***FZ NP00000****"* respetando así la información gramatical contenida dentro del comentario.

4.1.1.4. Creación de vectores

Hasta la fase anterior, sólo se ha realizado la separación de las clases y el etiquetado gramatical de los textos teniendo en cuenta las etiquetas dependientes del corpus. Para la creación de los vectores de entrenamiento es necesario hablar sobre el tipo de características que utiliza el algoritmo y la justificación de éstas.

Como se mencionó en la Sección 2.3.3 *Características para la clasificación automática de texto*, es posible identificar dos grupos de características: las características de contenido y las características estilísticas [Argamon et al., 2009]. Estudios con los dos grupos de características han demostrado ser efectivos en textos tradicionales (textos literarios, documentales y ensayos) [Peersman et al., 2011] como en textos procedentes de Internet (blogs, sitios de reseñas, sitios de opinión y redes sociales) [Goswami et al., 2009, Mukherjee and Liu, 2010, Peersman et al., 2011, Schler et al., 2006].

Para los estudios realizados sobre textos tradicionales es común observar el uso de ambos tipos de características. En 2005, Doyle *et al.* trabajaron sobre una colección de ensayos en inglés de estudiantes del corpus *BAWE*[4] utilizando n-gramas de caracteres (característica de estilo), n-gramas de palabras (característica de estilo), n-gramas POS (característica de estilo) y palabras funcionales (característica de contenido) [Doyle and Kešelj, 2005].

En el caso de los textos procedentes de internet, también es posible observar el uso de las mismas características. Schler *et al.* en 2006 trabajaron sobre un corpus de 71 mil blogs procedentes de Blogger[5] utilizando una combinación de características para la clasificación de género y edad; características como etiquetas POS (característica de estilo), palabras del blog e hipervínculos (características de estilo), palabras funcionales (característica de estilo), palabras de contenido (característica de contenido) y clases de palabras (característica de contenido) [Schler et al., 2006].

Los resultados reportados muestran que, si bien es útil la combinación de dichas características, es posible que dependiendo de la tarea un grupo de ellas sea más eficiente que otro. En la clasificación de género resultó que las características de estilo fueron más informativas que las características de contenido mientras que en la clasificación de edad, las características de contenido resultaron ser más informativas.

Otro ejemplo es el caso de Nguyen *et al.*, que en 2011 utilizaron tres corpus (dos procedentes de Internet) para clasificar género y predecir la edad mediante una regresión: blogs, conversaciones telefónicas y posts de foros. Las características que utilizaron fueron unigramas de caracteres (característica de estilo), unigramas y bigramas POS (características de estilo) y clases de palabras (característica de contenido).

El presente algoritmo utiliza únicamente características de estilo para representar

[4] http://www2.warwick.ac.uk/fac/soc/al/research/collect/bawe/
[5] http://blogger.com/

vectorialmente los textos procedentes de la red social en cuestión; existen dos razones que motivan la postura de sólo utilizar éstas. En primer lugar, la intención de que el algoritmo sea aplicable a diferentes idiomas hace que se vuelva complicado utilizar características de contenido, ya que se debería tener un conocimiento amplio de la lengua a procesar y así lograr extraer características como palabras de contenido, clases de palabras o grupos de palabras.

En segundo lugar, al trabajar con textos cortos procedentes de Internet es poco práctico extraer palabras funcionales, palabras de contenido o clases de palabras, ya que los textos no se encuentran normalizados. Si bien es cierto que es posible normalizar el texto a procesar, esta normalización se tendría que hacer de forma supervisada. Es necesario mencionar que el costo de normalizar este tipo de textos es muy alto, debido a que al efectuar la normalización, la gran cantidad de información estilística que podría haber sido extraída del texto es eliminada.

Se tomó otra muestra del corpus *Comentarios de la Ciudad de México en el tiempo*. Al revisarla se puede considerar realizar una normalización (Tabla 4.2) bajo los siguientes criterios:

- En caso de encontrar el mismo signo de puntuación más de una vez de forma consecutiva, únicamente mostrar uno.
- Cambiar todas las letras mayúsculas por minúsculas.
- Si en una palabra, una letra existe más de una vez de forma consecutiva, únicamente mostrar una ocurrencia.
- Seguir las siguientes reglas de sustitución:
  - ke → que
  - k → que
  - x → por

Tabla 4.2: Comentarios normalizados

Clasificación	Comentarios	Comentarios normalizados
Hombre	wooooooooooooooooooooooooooooooooowwwwwwwwwwwwgeniaaaaaaaaaaaaaal!!!	wow genial!
	SOLO SE QUE ES... GRADIOSO ESE LUGAR...	solo se que es. gradioso ese lugar.
	Simplemente hermosaaaaaa <3	simplemente hermosa <3
	Aaaaaahhhh estoy a un ladooooooooo. Esos Jardines del sur.	ah estoy a un lado. esos jardines del sur.
	De primerisimo mundo!!!!!!!!!	de primerisimo mundo!
Mujer	Wooooow es hermosaaa!!!! Gracias Cosmopitufo Espacial vecino!!	wow es hermosa! gracias cosmopitufo espacial vecino!
	bonito lugar, y gracias x kompartir stas fotos. ;-) ;-)	bonito lugar, y gracias por kompartir stas fotos. ;-) ;-)
	hOy ese edificiO alberga Oficinas administrativas de la Secretaría de Salud, de ladO derechO esta el eificiO de la cOmisiÓn federal de luz, enfrente al mOnunemtO a cuauhtemOc , ahí ene ese edificiO trabaja mi mami :)	hoy ese edificio alberga oficinas administrativas de la secretaría de salud, de lado derecho esta el eificio de la comisión federal de luz, enfrente al monunemto a cuauhtemoc , ahí ene ese edificio trabaja mi mami :)
	Bonitos recuerdos de la voca 5 y la ciudadela :)	bonitos recuerdos de la voca 5 y la ciudadela :)
	durante mis estudios en la vocacional 5 Taxqueña (83-86) use la linea 1 y 2 del metro	durante mis estudios en la vocacional 5 taxqueña (83-86) use la linea 1 y 2 del metro

Debido a la normalización llevada a cabo es posible agrupar palabras funcionales y palabras de contenido, así como identificar clases de palabras. Estas características que se logran extraer nos aportan información como:

- Las expresiones utilizadas por los hombres y las mujeres para enlazar ideas (palabras funcionales)
- La finalidad de los comentarios escritos por hombres y mujeres (palabras de contenido)
- Las diferentes temáticas expresadas en los comentarios escritos por hombres y mujeres (clases de palabras)

El costo de efectuar la normalización mostrada en la Tabla 4.2 es la pérdida de información estilística como:

- Interacción entre letras mayúsculas y minúsculas:
  - SOLO SE QUE ES... GRADIOSO ESE LUGAR...
  - $\cdots$ de la Secretaría de Salud, de ladO derechO esta el eificiO de la cOmisiÓn federal de luz, enfrente al mOnunemtO a cuauhtemOc$\cdots$
- Diferencia en la cantidad de signos de puntuación utilizados de forma consecutiva:
  - De primerisimo mundo!!!!!!!!!
  - Wooooow es hermosaaa!!!! Gracias Cosmopitufo Espacial vecino!!
- Expresión de emociones al repetir de forma consecutiva caracteres en las palabras:
  - wooooooooooooooooooooooo oooooooooooowwwwwwwwwww geniaaaaaaaaaaaaaal!!!
  - Wooooow es hermosaaa!!!! Gracias Cosmopitufo Espacial vecino!!
- Tendencias en la forma de economizar el lenguaje utilizando abreviaciones fonéticas:
  - bonito lugar, y gracias x kompartir stas fotos. ;-) ;-)

Las características de estilo seleccionadas están compuestas por **n-gramas de caracteres** y **n-gramas de etiquetas gramaticales (n-gramas POS)**. El valor de n de los n-gramas de caracteres como de los n-gramas POS es totalmente ajustable.

Con los n-gramas de caracteres lo que se pretende es extraer la mayor cantidad de elementos estilísticos posible que la seriación de caracteres pueda proveer. Al utilizar $n = 1$, lo que extraen los 1-gramas de caracteres es la frecuencia de cada uno de los caracteres utilizados por hombres y mujeres dejando a un lado el contexto de los mismos; retomando el ejemplo mostrado en la Tabla 4.2, los 5 1-gramas más utilizados por hombres y mujeres son los mostrados en la Tabla 4.3

Para lograr extraer el contexto entre caracteres es necesario incrementar el tamaño de los gramas; por ejemplo, si se utilizan 2-gramas de caracteres es posible que dentro de la gran cantidad de combinaciones existentes se logren extraer sufijos de palabras que indiquen género y número (Tabla 4.4).

En el ejemplo dado es posible visualizar repeticiones de signos de puntuación contiguos, esto podría darnos información relacionada con la forma en que hombres y mujeres plasman ciertos sentimientos. Otra forma de plasmar sentimientos dentro de

Tabla 4.3: 1-gramas de caracteres

Hombres	Mujeres
o = 52	a = 47
a = 28	e = 43
w = 14	i = 31
! = 12	o = 23
e = 10	d = 19

Tabla 4.4: 2-gramas de caracteres

	Hombres		Mujeres	
	Masculino	Femenino	Masculino	Femenino
Singular	so, lo, un, do, el, mo	sa	no, to, iO, dO, hO, Ón, tO, ro	sa, ga, va, ta, ca, la
Plural	os, es	es	es, os	es, as

los comentarios es con emoticonos; utilizando 2-gramas de caracteres es posible extraer emoticonos de dos caracteres como: *¡3, :), :(, :D, D:, XD*, etc.. La siguiente tabla (Tabla 4.5) muestra el uso de signos de puntuación y ejemplifica lo ya comentado.

Tabla 4.5: 2-gramas de caracteres

Hombres	Mujeres
!! = 10	!! = 4
.. = 4	:) = 2
<3 = 1	;- = 2

Al utilizar 3-gramas de caracteres es posible la extracción de sufijos más largos; sufijos que indiquen tiempos verbales, diminutivos o superlativos.

Por su parte, al utilizar n-gramas POS es posible obtener desde la frecuencia de elementos gramaticales utilizados por cada una de las clases definidas, hasta la frecuencia de las estructuras gramaticales empleadas; esto a partir de la longitud de los n-gramas POS.

Para lograr el etiquetado gramatical y así obtener los n-gramas POS, se hace uso del *software* Freeling. Este *software* es capaz de etiquetar gramaticalmente las palabras a un nivel muy detallado; por ejemplo: la palabra “monumental” es etiquetada de la siguiente forma: *AQ0CS0* (Tabla 4.6).

Entre las ventajas de tener este nivel de detalle se encuentra la posibilidad de

Tabla 4.6: Etiquetado gramatical de la palabra *monumental*

monumental						
Atributo	Categoría	Tipo	Grado	Género	Número	Función
Código	A	Q	0	C	S	0
Valor	Adjetivo	Calificativo	N/A	Común	Singular	-

identificar con gran exactitud las estructuras gramaticales que son empleadas; el gran incoveniente es que a mayor detalle en los elementos gramaticales, mayor es la dispersión que se presenta al momento de obtener la frecuencia de los elementos. Debido a esto, únicamente se toma en cuenta el primer nivel de detalle que hace referencia a la categoría gramatical a la que pertenece la palabra o signo, quedando etiquetada la palabra "monumental" como: *A* (Tabla4.7).

Tabla 4.7: Etiquetado gramatical utilizado de la palabra *monumental*

monumental	
Atributo	Categoría
Código	A
Valor	Adjetivo

En el ejemplo dado, al hacer la extracción de los 1-gramas POS se obtienen las siguientes frecuencias (Tabla 4.8):

Tabla 4.8: Frecuencia de 1-gramas POS

Código	Valor	Frecuencia	
		Hombres	Mujeres
N	Nombre	7	25
F	Puntuación	18	23
D	Determinante	3	14
S	Preposición	3	10
V	Verbo	2	8
A	Adjetivo	2	7
Z	Numeral	0	5
C	Conjunción	0	3
R	Adverbio	2	2

Empleando 2-gramas POS es posible comenzar a descubrir el contexto de los elementos gramaticales así como su frecuencia dependiendo de la clase a la que per-

tenecen. En la siguiente tabla se muestran los cinco 2-gramas POS más frecuentes tanto de hombres como de mujeres (Tabla 4.9):

Tabla 4.9: Frecuencia de 2-gramas POS

Hombres		Mujeres	
2-gramas POS	Frecuencia	2-gramas POS	Frecuencia
F-F	11	F-F	12
N-F	5	D-N	10
D-N	3	N-F	9
F-N	3	S-D	7
S-D	2	N-S	5

Si se utilizan 3-gramas POS, se obtienen estructuras más complejas y menos frecuentes de elementos gramaticales. Los cinco 3-gramas POS más frecuentes tanto de hombres como de mujeres del ejemplo dado son los siguientes (Tabla 4.10):

Tabla 4.10: Frecuencia de 3-gramas POS

Hombres		Mujeres	
3-gramas POS	Frecuencia	3-gramas POS	Frecuencia
F-F-F	8	F-F-F	7
D-N-F	2	S-D-N	5
S-D-N	2	N-S-D	4
F-N-F	2	N-F-F	4
F-F-N	2	D-N-F	3

Una vez justificado el tipo de características que utiliza el algoritmo y la forma en que se extraen las mismas, es momento de retomar la muestra original del corpus de Facebook *Comentarios de la Ciudad de México en el tiempo* y su etiquetado gramatical para obtener las características que lo representan.

La Tabla 4.11 retoma los comentarios de la muestra original, el etiquetado gramatical que genera Freeling, el re-etiquetado dependiente del corpus y, finalmente, las etiquetas tomadas en cuenta por el algoritmo.

Tabla 4.11: Comentarios final

Hombres	Comentarios	• Suena interesante @Lena polii !! • @Alejandro Campos. ¿dirás 1554? Saludos. • how it's happnd ? pls write to me in english @mera • @Enrique Gonzalez Gomar ¿Donde puedo consultar esa información que mencionas? • ya vieron @lahijadelosapaches
	Etiquetado gramatical	• VMIP3S0 AQ0CS0 Fz NP00000 NCFS000 Fat Fat • Fz NP00000 Fp Fia VMIF2S0 Z Fit NP00000 Fp • RG VMIP3S0 NCMS000 Fit RG VMIP3S0 RG PP1CS000 AQ0CS0 NCMS000 Fz AQ0FS0 • Fz NP00000 Fia PR000000 VMIP1S0 VMN0000 DD0FS0 NCFS000 PR0CN000 VMIP2S0 Fit • RG VMIS3P0 Fz NCFP000
	Re-etiquetado	• VMIP3S0 AQ0CS0 REF#USER NCFS000 Fat Fat • REF#USER Fp Fia VMIF2S0 Z Fit NP00000 Fp • RG VMIP3S0 NCMS000 Fit RG VMIP3S0 RG PP1CS000 AQ0CS0 NCMS000 REF#USER • REF#USER Fia PR000000 VMIP1S0 VMN0000 DD0FS0 NCFS000 PR0CN000 VMIP2S0 Fit • RG VMIS3P0 REF#USER
	Etiquetas finales	• V A REF#USER N F F • REF#USER F F V Z F N F • R V N F R V R P A N REF#USER • REF#USER F P V V D N P V F • R V REF#USER
Mujeres	Comentarios	• @Raul Garcia .... ¿cómo se organizaban para las labores de recuperación de cuerpos? • @javier campuzano @jorgesgvn • Mira @omar Zamudio • @Jesus Mejia Perez Dinos que métodos, por favor!
	Etiquetado gramatical	• Fz NP00000 Fz Fia PT000000 P00CN000 VMII3P0 SPS00 DA0FP0 NCFP000 SPS00 NCFS000 SPS00 NCMP000 Fit

		• Fz VMN0000 NCMS000 Fz NCFS000 • NP00000 Fz VMN0000 NP00000 • Fz NP00000 CS NCMP000 Fc RG Fat
	Re-etiquetado	• REF#USER Fz Fia PT000000 P00CN000 VMII3P0 SPS00 DA0FP0 NCFP000 SPS00 NCFS000 SPS00 NCMP000 Fit • REF#USER NCMS000 REF#USER • NP00000 REF#USER NP00000 • REF#USER CS NCMP000 Fc RG Fat
	Etiquetas finales	• REF#USER F F P P V S D N S N S N F • REF#USER N REF#USER • N REF#USER N • REF#USER C N F R F

Tanto los n-gramas de caracteres como los n-gramas POS tienen la posibilidad de ser retroactivos; esto significa que si se elige utilizar 3-gramas retroactivos, implícitamente también se está haciendo uso de los 2-gramas y los 1-gramas de caracteres y POS. La decisión de hacer retroactivos los n-gramas repercute en gran medida en el rendimiento del algoritmo, pues el tamaño de las características obedece a la siguiente fórmula sumatoria:

$$L = \sum_{n=1}^{N}(A - N + 1) \tag{4.1}$$

En donde L es el número de características, N es el grama mayor y A es la cantidad de *tokens* (letras o etiquetas gramaticales).

Otro elemento que es posible notar es que el caracter de espacio en blanco en realidad no aporta información estilística alguna, por lo que se puede eliminar a la hora de procesar los comentarios. Esta acción implica que el número de características se vea reducido.

Al generar los 3-gramas de caracteres retroactivos del siguiente comentario: *"Suenainteresante@Lenapolii!!"*

$$\begin{aligned} L_{caracteres} &= \sum_{n=1}^{3}(28 - n + 1) \\ L_{caracteres} &= (28 - 1 + 1) + (28 - 2 + 1) + (28 - 3 + 1) \\ L_{caracteres} &= 81 \end{aligned}$$

Y al generar los 3-gramas POS retroactivos de su etiquetado gramatical:
V A REF#USER N F F

$$\begin{aligned} L_{POS} &= \sum_{n=1}^{3}(6-n+1) \\ L_{POS} &= (6-1+1)+(6-2+1)+(6-3+1) \\ L_{POS} &= 15 \\ L_{caracteres}+L_{POS} &= 81+15=96 \end{aligned}$$

Como se puede observar, una cadena de 32 caracteres es capaz de generar 96 características. Ese comportamiento en corpus de gran tamaño resulta en vectores de características de gran longitud.

Si no fuera necesario el uso de n-gramas retroactivos, el número de características sigue la siguiente función:

$$L = (A - N + 1) \tag{4.2}$$

En donde L es el número de características, N es el la longitud de los n-gramas y A es la cantidad de *tokens* (letras o etiquetas gramaticales).

Al generar los 3-gramas de caracteres no retroactivos del siguiente comentario:
"Suenainteresante@Lenapolii!!"

$$\begin{aligned} L_{caracteres} &= (28-3+1) \\ L_{caracteres} &= 26 \end{aligned}$$

Y al generar los 3-gramas POS no retroactivos de su etiquetado gramatical:
V A REF#USER N F F

$$\begin{aligned} L_{POS} &= (6-3+1) \\ L_{POS} &= 4 \\ L_{caracteres}+L_{POS} &= 26+4=30 \end{aligned}$$

A diferencia de los 3-gramas retroactivos, los 3-gramas no retroactivos generan 30 características. Dependiendo del corpus a procesar y de los recursos con los que se cuente, cualquiera de las dos opciones puede ser elegida.

Una vez explicadas las características de las que hace uso la presente propuesta, se detallará el algoritmo utilizado para la creación de vectores:

1. $\forall m \in M$:

   a) Extraer los n-gramas de caracteres como sus frecuencias y generar un vector de características $vCAR_m$

   $$vCAR_m = [(n-grama1, f_{n-grama1}), (n-grama2, f_{n-grama2}), \cdots, (n-gramaZ, f_{n-gramaZ})]$$

   b) Extraer los n-gramas POS como sus frecuencias y generar un vector de características $vPOS_m$

   $$vPOS_m = [(n-gramaPOS1, f_{n-gramaPOS1}), (n-gramaPOS2, f_{n-gramaPOS2}), \cdots, (n-gramaPOSZ, f_{n-gramaPOSZ})]$$

2. Generar un vector general de características $vCAR_{gral}$ asignando un cero a las frecuencias obtenidas
   $$vCAR_{gral} = \bigcup_{m \in M} vCAR_m$$

3. Generar un vector general de características $vPOS_{gral}$ asignando un cero a las frecuencias obtenidas
   $$vPOS_{gral} = \bigcup_{m \in M} vPOS_m$$

4. Generar el vector de características del sistema.
   $$vSistema = vCAR_{gral} \cup vPOS_{gral} \tag{4.3}$$

   Nota: La longitud de este vector indica la cantidad de características o *features* de cada muestra.

5. $\forall m \in M$:

   a) Generar un vector de características de la muestra.
   $$vMuestra_m = vSistema$$

   b) Asignar la frecuencia de los n-gramas de caracteres de $vCAR_m$ en $vMuestra_m$

   c) Asignar la frecuencia de los n-gramas POS de $vPOS_m$ en $vMuestra_m$

Una vez implementado el algoritmo, se cuenta con M vectores de características de una longitud $|vSistema|$; formando así una matriz de entrenamiento denominada *matrizEntrenamiento*.

$$matrizEntrenamiento = \{vMuestra_m | m \in M\} \times vSistema$$

Para ejemplificar la creación de vectores, se hará uso de los comentarios mostrados en la Tabla 4.2 y se utilizarán 2-gramas de caracteres y de 2-gramas POS no retroactivos. Los comentarios etiquetados como *hombre* poseen el subíndice h, mientras que los comentarios etiquetados como *mujer*, el m. Debido a la longitud de las tablas que se generan, el ejemplo se localiza en el Capítulo 8 *Anexo* Sección 8.1 *Creación de vectores*.

4.1.1.5. Entrenamiento

Esta última etapa se encarga del entrenamiento del clasificador ϕ. Debido a que el presente algoritmo es modular, la etapa de entrenamiento tiene la libertad de implementar cualquier algoritmo clasificador que acepte muestras de entrenamiento con el formato de la matriz de entrenamiento creada en la etapa anterior.

4.1.2. Fase de prueba

La finalidad de la fase de prueba es predecir de forma automática la clase $c \in C$ a la que pertenece una muestra $p \notin M$. Para lograr esto, se hace uso del clasificador ϕ entrenado en la última etapa de la fase de entrenamiento.

Esta fase se divide en cuatro etapas: Etiquetado, Generación de POS, Creación del vector general y Prueba.

Figura 4.2: Etapas de la fase de prueba

4.1.2.1. Etiquetado

La muestra p a clasificar es enviada a Freeling para ser analizada y etiquetada gramaticalmente. Obteniendo así una cadena en formato JSON[6] (p_{JSON}).

[6]Freeling es llamado utilizando una interfaz que convierte su salida en una cadena JSON. Esta interfaz fue desarrollada por el Grupo de Ingeniería Lingüística, Instituto de Ingeniería, UNAM.

4.1.2.2. Generación de POS

Una vez obtenida la cadena p_{JSON}, ésta es procesada siguiendo la lógica de la Normalización dinámica dependiente del contexto dando como resultado el vector p_{POS}.

4.1.2.3. Creación del vector general

1. Extraer de p los n-gramas de caracteres como sus frecuencias y generar un vector de características $vCAR_p$

   $$vCAR_p = [(n-grama1, f_{n-grama1}), (n-grama2, f_{n-grama2}), \cdots, (n-gramaZ, f_{n-gramaZ})]$$

2. Extraer de p_{POS} los n-gramas POS como sus frecuencias y generar un vector de características $vPOS_p$

   $$vPOS_p = [(n-gramaPOS1, f_{n-gramaPOS1}), (n-gramaPOS2, f_{n-gramaPOS2}), \cdots, (n-gramaPOSZ, f_{n-gramaPOSZ})]$$

3. Para crear el vector general, se retoma el vector de características del sistema ($vSistema$) definido en (4.3) y se le asignan las frecuencias existentes en los vectores $vCAR_p$ y $vPOS_p$.

   $$vGeneral_m = vSistema \leftarrow \forall v \in (vCAR_p \cup vPOS_p) \tag{4.4}$$

4.1.2.4. Prueba

El vector general ($vGeneral_m$) es ingresado al clasificador ϕ y éste le asigna una clase $c \in C$.

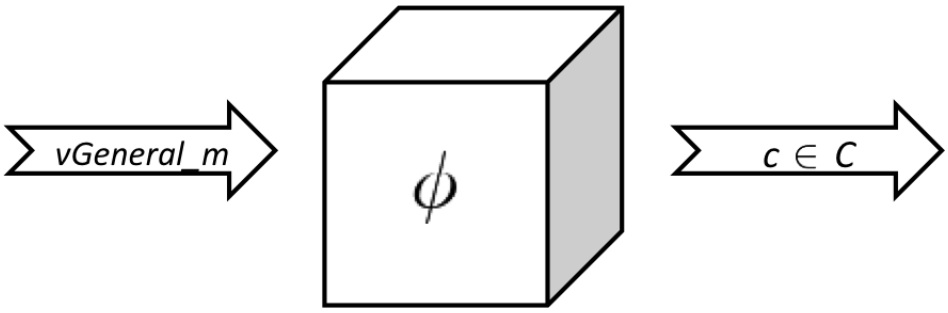

Figura 4.3: Clasificador

4.2. Protocolo experimental

4.2.1. Planteamiento

Cada año el Laboratorio de evaluación *Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN)* organiza un concurso a nivel internacional con temas relacionados a la detección de plagio y análisis de autoría; este año la competencia estuvo conformada por tres tareas: Identificación de Autor[7] [PAN, 2015a], Detección de Plagio[8] [PAN, 2015c] y Perfilado de Autor[9] [PAN, 2015b].

La finalidad de la Identificación de Autor es que al dar un documento, se logre predecir de manera correcta su autor. La Detección de Plagio tiene como objetivo identificar todas las fuentes plagiadas a partir de un documento dado y finalmente el Perfilado de Autor pretende que dado un documento se extraiga la información del autor.

La tarea de Perfilado de Autor de este año tuvo como objetivo identificar género, grupo etario y ciertos rasgos de personalidad (extrovertido, estable, agradable, concienzudo y abierto) de usuarios de Twitter, tarea muy apegada al objetivo de la presente investigación; por lo que se tomó la decisión de entrar al concurso y desarrollar la experimentación bajo las reglas del certamen.

Un aspecto importante a recalcar es que la tarea no consistió en identificar género, grupo etario y rasgos de personalidad a partir de un tuit, sino identificar género y edad a partir de un grupo de 100 tuits de un autor en particular; por lo que al hablar de *muestras* en realidad se hace mención al grupo de 100 tuits de un autor.

Cabe mencionar que los rasgos de personalidad se salen de los objetivos ya mencionados; por esta razón, y aunque sí se abordó el problema de forma satisfactoria en el concurso, se hará poco hincapié en ellos. En caso de interesarse por el tema se recomienda hacer la lectura del artículo de [González-Gallardo et al., 2015].

[7] http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-web/author-identification.html

[8] http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-web/plagiarism-detection.html

[9] http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-web/author-profiling.html

4.2.2. Descripción del corpus

Para realizar las pruebas pertinentes durante la fase de desarrollo del concurso, los organizadores liberaron un corpus (corpus de entrenamiento), el cuál fue necesario dividir para entrenar y probar el sistema desarrollado. Posteriormente, en la fase de evaluación un segundo corpus fue utilizado para medir el rendimiento de los sistemas (corpus de evaluación).

4.2.2.1. Corpus de entrenamiento

El corpus de entrenamiento se encuentra constituido por muestras de tuits en español, inglés, italiano y holandés. Con respecto a género, el corpus se encuentra balanceado en los cuatro idiomas; 50 % se encuentran etiquetados como *female* (mujer) y 50 %, como *male* (hombre) (Tabla 4.12).

Tabla 4.12: Distribución por género del corpus de entrenamiento

Idioma	*female*		*male*		Muestras totales
	Muestras	Porcentaje	Muestras	Porcentaje	
Español	50	50 %	50	50 %	100
Inglés	76	50 %	76	50 %	152
Italiano	19	50 %	19	50 %	38
Holandés	17	50 %	17	50 %	34

Para español e inglés la edad es definida en cuatro rangos diferentes: $18-24$, $25-34$, $35-49$ y $50-xx$; la distribución de estos rangos se muestra en la Tabla 4.13 y en la Figura 4.4. El italiano y el holandés no cuentan con rangos de edad definidos, por lo que sólo es posible hacer clasificación por género en estos idiomas.

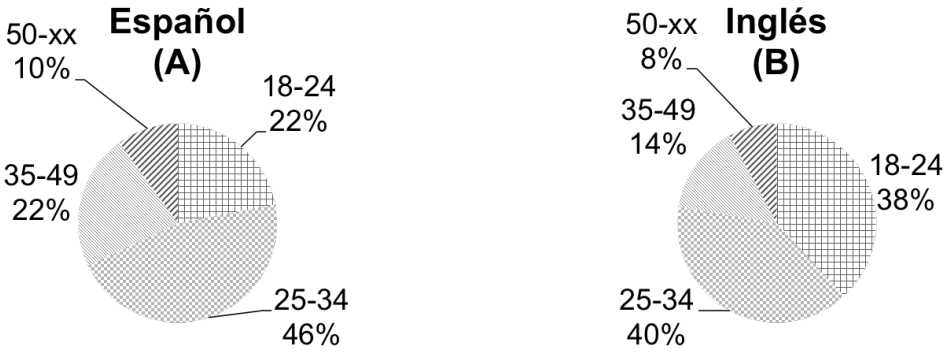

Figura 4.4: Distribución del corpus de entrenamiento edad

Tabla 4.13: Distribución por edad del corpus de entrenamiento

Grupo		Español	Inglés
18-24	Muestras	22	58
	Porcentaje	22 %	38 %
25-34	Muestras	46	60
	Porcentaje	46 %	40 %
35-49	Muestras	22	22
	Porcentaje	22 %	14 %
50-xx	Muestras	10	12
	Porcentaje	10 %	8 %
	Cantidad total de muestras	100	152

Cada idioma está constituido por una carpeta en donde se encuentran los archivos que contienen las muestras de tuits (un archivo por muestra); estos archivos tienen la extensión `xml` y su estructura es la siguiente:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<author id="user#" lang="xx">
    <document><![CDATA[tweet1]]></document>
    <document><![CDATA[tweet2]]></document>
    .
    .
    .
    <document><![CDATA[tweetN]]></document>
</author>
```

El atributo `id="user#"` hace referencia al autor de la muestra y corresponde al nombre del archivo. Si un archivo se llama `user22.xml`, el valor del atributo será `id="user22"`. El atributo `lang="xx"` indica el idioma de la muestra; los posibles valores son: `es` (español), `en` (inglés), `it` (italiano) y `nl` (holandés).

A parte de los archivos `xml`, la carpeta contiene un archivo llamado `truth.txt` que contiene la información de las clases a la que pertenece cada muestra; su estructura es la siguiente:

```
user#:::[M|F]:::[18-24|25-34|35-49|50-xx]|XX-XX:::{-0.5 - 0.5}:::\
{-0.5 - 0.5}:::{-0.5 - 0.5}:::{-0.5 - 0.5}:::{-0.5 - 0.5}
user#:::[M|F]:::[18-24|25-34|35-49|50-xx]|XX-XX:::{-0.5 - 0.5}:::\
{-0.5 - 0.5}:::{-0.5 - 0.5}:::{-0.5 - 0.5}:::{-0.5 - 0.5}
.
.
```

.

```
user#:::[M|F]:::[18-24|25-34|35-49|50-xx]|XX-XX:::{-0.5 - 0.5}:::\
{-0.5 - 0.5}:::{-0.5 - 0.5}:::{-0.5 - 0.5}:::{-0.5 - 0.5}
```

El primer elemento `user#` indica la muestra en cuestión; `[M|F]` es el género al que pertenece la muestra: `M` → *hombre*, `F` → *mujer*. Finalmente, el tercer elemento indica el grupo de edad al que pertenece la muestra: `18-24 o 25-34 o 35-49 o 50-xx` para las muestras en español e inglés, y `XX` para las muestras en italiano y holandés.

Para entrenar al sistema clasificador y, posteriormente probarlo, fue necesario dividir el corpus en dos partes. La primera parte, la de entrenamiento, se compone del 70 % de las muestras (Tabla 4.14 y 4.15); mientras que la segunda se compone del 30 % restante (Tabla 4.16 y 4.17).

Tabla 4.14: Distribución por género del corpus de entrenamiento (sección de entrenamiento)

Idioma	*female*		*male*		Muestras totales
	Muestras	Porcentaje	Muestras	Porcentaje	
Español	35	50 %	35	50 %	70
Inglés	53	50 %	53	50 %	106
Italiano	13	50 %	13	50 %	26
Holandés	12	50 %	12	50 %	24

Tabla 4.15: Distribución por edad del corpus edad entrenamiento (sección de entrenamiento)

Grupo		Español	Inglés
18-24	Muestras	16	40
	Porcentaje	22.86 %	37.74 %
25-34	Muestras	32	42
	Porcentaje	45.71 %	39.62 %
35-49	Muestras	16	16
	Porcentaje	22.86 %	15.09 %
50-xx	Muestras	6	8
	Porcentaje	8.57 %	7.55 %
	Cantidad total de muestras	70	106

Tabla 4.16: Distribución por género del corpus de entrenamiento (sección de evaluación)

Idioma	*female*		*male*		Muestras totales
	Muestras	Porcentaje	Muestras	Porcentaje	
Español	15	50 %	15	50 %	30
Inglés	23	50 %	23	50 %	46
Italiano	6	50 %	6	50 %	12
Holandés	5	50 %	5	50 %	10

Tabla 4.17: Distribución por edad del corpus de entrenamiento (sección de evaluación)

Grupo		Español	Inglés
18-24	Muestras	6	18
	Porcentaje	20 %	39.13 %
25-34	Muestras	14	18
	Porcentaje	46.67 %	39.13 %
35-49	Muestras	6	6
	Porcentaje	20 %	13.04 %
50-xx	Muestras	4	4
	Porcentaje	13.33 %	8.7 %
	Cantidad total de muestras	30	46

4.2.2.2. Corpus de evaluación

Debido a que es posible que el mismo corpus de evaluación sea utilizado en futuras ocasiones, no se contó con acceso al mismo; por lo que la información relativa a esta sección se obtuvo a partir del artículo publicado por [Rangel et al., 2015].

De la misma forma que el corpus de entrenamiento, el corpus de evaluación se encuentra constituido por muestras de tuits en español, inglés, italiano y holandés. Con respecto al género, el corpus se encuentra balanceado en los cuatro idiomas; 50 % se encuentran etiquetados como *female* (mujer) y 50 % como *male* (hombre) (Tabla 4.18).

Para español e inglés, la edad está definida en cuatro rangos diferentes: $18-24$, $25-34$, $35-49$ y $50-xx$; la distribución de estos rangos se muestra en la Tabla 4.19 y en la Figura 4.5. En concordancia con el corpus de entrenamiento, el italiano y el holandés no cuentan con rangos de edad definidos.

Cada idioma está constituido por una carpeta en donde se encuentran los archivos que contienen las muestras de tuits (un archivo por muestra); estos archivos tienen

Tabla 4.18: Distribución por género del corpus de evaluación

Idioma	*female*		*male*		Muestras totales
	Muestras	Porcentaje	Muestras	Porcentaje	
Español	44	50 %	44	50 %	88
Inglés	71	50 %	71	50 %	142
Italiano	18	50 %	18	50 %	36
Holandés	16	50 %	16	50 %	32

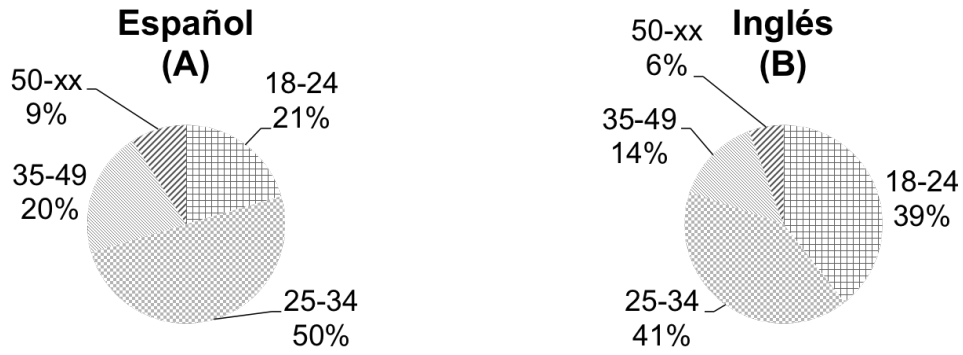

Figura 4.5: Distribución por edad del corpus de evaluación

la extensión `xml` y su estructura es la siguiente:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<author id="user#" lang="xx">
    <document><![CDATA[tweet1]]></document>
    <document><![CDATA[tweet2]]></document>
    .
    .
    .
    <document><![CDATA[tweetN]]></document>
</author>
```

El atributo `id="user#"` hace referencia al autor de la muestra y corresponde al nombre del archivo. Si un archivo se llama `user22.xml`, el valor del atributo será `id="user22"`. El atributo `lang="xx"` indica el idioma de la muestra; los posibles valores son: `es` (español), `en` (inglés), `it` (italiano) y `nl` (holandés).

Tabla 4.19: Distribución por edad del corpus de evaluación

Grupo		Español	Inglés
18-24	Muestras	18	56
	Porcentaje	20 %	39 %
25-34	Muestras	44	58
	Porcentaje	50 %	41 %
35-49	Muestras	18	20
	Porcentaje	20 %	14 %
50-xx	Muestras	8	8
	Porcentaje	10 %	6 %
	Cantidad total de muestras	88	142

4.2.3. Descripción del experimento

La implementación del algoritmo ya descrito se realizó en el lenguaje de programación Python[10]. Esta implementación está constituida por 12 *scripts* de Python; seis para la fase de entrenamiento y seis para la fase de prueba.

La experimentación abarca los idiomas español, inglés, italiano y holandés. Éstos fueron tratados de forma independiente.

- Entrenamiento
  - `PanEntrenamiento.py`
  - `extractorV2.py`
  - `etiquetadorPOSV2.py`
  - `generadorPOSV2.py`
  - `generadorVectoresV3.py`
  - `generadorNGramasCP.py`
- Prueba
  - `PanPrueba.py`
  - `extractorPrueba.py`
  - `etiquetadorPOSV2Prueba.py`
  - `generadorPOSV2Prueba.py`
  - `generadorVectoresV3Prueba.py`
  - `generadorNGramasCPPrueba.py`

[10]https://www.python.org/

4.2.3.1. Entrenamiento

El *script* `PanEntrenamiento.py` es el encargado de hacer las llamadas al resto de los *scripts* de la fase de entrenamiento, cumpliendo así con las cinco etapas de esta fase (Figura 4.1).

Extracción

`PanEntrenamiento.py` → `extractorV2.py`

En esta etapa, el *script* `extractorV2.py` se encarga de unir todas las muestras (archivos `.xml`) pertenecientes a cada una de las clases definidas: ocho, en el caso de inglés y español ($M\_20s, M\_30s, M\_40s, M\_50s, F\_20s, F\_30s, F\_40s, F\_50s$); y dos, en el caso de italiano y holandés (M, F), generando así los *archivos de clase*.

Al unir los archivos `.xml`, el *script* elimina todas las etiquetas XML. Así mismo, ejecuta las sustituciones mostradas en la Tabla 4.20.

Tabla 4.20: Sustituciones realizadas por `extractorV2.py`

Texto	Explicación	Sustitución
@username	Referencia a otro usuario de Twitter	@us
http[s]://...	Liga a un sitio externo	htt
`\n`	Salto de línea	[espacio]

Archivos de clase creados según el idioma a procesar:

- Español
  - `es_F_20s.txt`
  - `es_F_30s.txt`
  - `es_F_40s.txt`
  - `es_F_50s.txt`
  - `es_M_20s.txt`
  - `es_M_30s.txt`
  - `es_M_40s.txt`
  - `es_M_50s.txt`
- Inglés
  - `en_F_20s.txt`
  - `en_F_30s.txt`

- en_F_40s.txt
- en_F_50s.txt
- en_M_20s.txt
- en_M_30s.txt
- en_M_40s.txt
- en_M_50s.txt

- Italiano
  - it_F.txt
  - it_M.txt
- Holandés
  - nl_F.txt
  - nl_M.txt

Etiquetado

PanEntrenamiento.py → etiquetadorPOSV2.py

El script etiquetadorPOSV2.py ingresa cada uno de los *archivos de clase* a Freeling para ser etiquetados gramaticalmente, y así obtener los *archivos JSON* correspondientes. Dependiendo del idioma que se esté procesando, etiquetadorPOSV2.py carga una instancia de Freeling configurada para analizar un idioma en especial.

Para el caso de Holandés, Freeling no cuenta con el módulo correspondiente, por lo que se tomó la decisión de procesarlo como si fuera inglés, tomando en cuenta la siguiente noción: **Si se utiliza el módulo de inglés para analizar holandés, Freeling va a cometer errores; pero si estos errores siguen un patrón estable, es posible que cierta información gramatical se pueda extraer.**

Archivos JSON que contienen la información gramatical de las clases de acuerdo con el idioma a procesar:

- Español
  - es_F_20s_freeling.json
  - es_F_30s_freeling.json
  - es_F_40s_freeling.json
  - es_F_50s_freeling.json
  - es_M_20s_freeling.json
  - es_M_30s_freeling.json

- es_M_40s_freeling.json
- es_M_50s_freeling.json

- Inglés
  - en_F_20s_freeling.json
  - en_F_30s_freeling.json
  - en_F_40s_freeling.json
  - en_F_50s_freeling.json
  - en_M_20s_freeling.json
  - en_M_30s_freeling.json
  - en_M_40s_freeling.json
  - en_M_50s_freeling.json
- Italiano
  - it_F_freeling.json
  - it_M_freeling.json
- Holandés
  - nl_F_freeling.json
  - nl_M_freeling.json

Generación de POS

`PanEntrenamiento.py` → `generadorPOSV2.py`

El *script* `generadorPOSV2.py` convierte la cadena JSON de los *archivos JSON* en *archivos POS*, igualando el formato de los *archivos de clase*. En esta etapa, se tiene en cuenta la Normalización dinámica dependiente del contexto, planteada en la descripción del algoritmo. De esta manera se aplican las reglas de re-etiquetado señaladas en la Tabla 4.21.

Tabla 4.21: Reglas de re-etiquetado implementadas por `generadorPOSV2.py`

Texto	Etiqueta
@us	REF#USERNAME
htt	REF#LINK
#{*texto*}	REF#HASTAG

Archivos POS con etiquetas gramaticales y estructura similar a los *archivos de clase* con respecto al idioma a procesar:

- Español
  - es_F_20s_freeling.pos
  - es_F_30s_freeling.pos
  - es_F_40s_freeling.pos
  - es_F_50s_freeling.pos
  - es_M_20s_freeling.pos
  - es_M_30s_freeling.pos
  - es_M_40s_freeling.pos
  - es_M_50s_freeling.pos
- Inglés
  - en_F_20s_freeling.pos
  - en_F_30s_freeling.pos
  - en_F_40s_freeling.pos
  - en_F_50s_freeling.pos
  - en_M_20s_freeling.pos
  - en_M_30s_freeling.pos
  - en_M_40s_freeling.pos
  - en_M_50s_freeling.pos
- Italiano
  - it_F_freeling.pos
  - it_M_freeling.pos
- Holandés
  - nl_F_freeling.pos
  - nl_M_freeling.pos

Creación de vectores

PanEntrenamiento.py → generadorVectoresV3.py → generadorNGramasCP.py

Dependiendo del idioma que está siendo procesado, PanEntrenamiento.py establece los mejores *parámetros de extracción* (basándose en una serie de pruebas manuales) que maximicen el rendimiento del sistema. Los *parámetros de extracción* tienen en cuenta la longitud de los n-gramas de caracteres *(num_gramas)*, así como la longitud de los n-gramas POS *(num_POS)*, si son retroactivos *(retro_gramas y retro_POS)*, el

Tabla 4.22: Parámetros de extracción

Parámetro	Valores
num_gramas	$0 \leq n$
num_POS	$0 \leq n$
retro_gramas	$0 \rightarrow NO$ $1 \rightarrow SI$
retro_POS	$0 \rightarrow NO$ $1 \rightarrow SI$
modo	$frec \rightarrow FRECUENCIA$ $bin \rightarrow BINARIO$
frec_log	$0 \rightarrow NO$ $1 \rightarrow SI$

Tabla 4.23: Parámetros de extracción seleccionados

Idioma	*num_gramas*	*num_POS*	*retro_gramas*	*retro_POS*	*modo*	*frec_log*
Español	3	3	1	1	frec	1
Inglés	2	3	1	1	frec	1
Italiano	3	3	1	1	frec	1
Holandés	3	3	1	1	frec	1

modo de conteo *(modo)* y si los n-gramas deberían ser representados en una escala logarítmica *(frec_log)* (Tabla 4.22).

Los *parámetros de extracción* seleccionados se muestran en la Tabla 4.23.

Una vez establecidos los *parámetros de extracción*, `PanEntrenamiento.py` invoca al *script* `generadorVectoresV3.py` para crear la matriz de entrenamiento *(matrizEntrenamiento)*. Éste a su vez llama al *script* `generadorNGramasCP.py`, el cual se encarga de obtener los vectores de características $vCAR_m$ y $vPOS_m$ de cada una de las muestras.

Entrenamiento

`PanEntrenamiento.py`

La matriz *(matrizEntrenamiento)* obtenida en la etapa anterior es ingresada al algoritmo de aprendizaje **LinearSVC**[11] [Pedregosa et al., 2011]; el cual es una implementación de una SVM con un kernel lineal. Una vez entrenado el algoritmo, es necesario almacenar el modelo de aprendizaje entrenado y el vector de características del sistema *(vSistema)* para que en la fase de prueba puedan ser recuperados.

[11] http://scikit-learn.org/stable/

- Español
  - es_modelo__M_F.pkl
  - es_features__M_F.pkl
- Inglés
  - en_modelo__M_F.pkl
  - en_features__M_F.pkl
- Italiano
  - it_modelo__M_F.pkl
  - it_features__M_F.pkl
- Holandés
  - nl_modelo__M_F.pkl
  - nl_features__M_F.pkl

4.2.3.2. Prueba

La fase de prueba se constituye por las siguientes cuatro etapas: Etiquetado, Generación de POS, Generación de vector general y Prueba (Figura 4.2). El *script* `PanPrueba.py` controla al resto de los *scripts* de esta fase.

Etiquetado

`PanPrueba.py` → `extractorPrueba.py`

`PanPrueba.py` → `etiquetadorPOSV2Prueba.py`

En esta primera etapa de la fase de prueba, `extractorPrueba.py` se encarga de procesar los archivos `.xml` correspondientes a las muestras de prueba; eliminando todas las etiquetas XML y sustituyendo el texto bajo las reglas establecidas en la Tabla 4.20. Una vez procesado el archivo `.xml` de la muestra, se crea su correspondiente archivo `.txt`.

- Español
  - es_muestra1.txt
  - es_muestra2.txt
  - es_muestra3.txt
  - es_muestra4.txt

    - es_muestra5.txt

- Inglés
    - en_muestra6.txt
    - en_muestra7.txt
    - en_muestra8.txt
    - en_muestra9.txt
    - en_muestra10.txt
- Italiano
    - it_muestra11.txt
    - it_muestra12.txt
    - it_muestra13.txt
    - it_muestra14.txt
    - it_muestra15.txt
- Holandés
    - nl_muestra16.txt
    - nl_muestra17.txt
    - nl_muestra18.txt
    - nl_muestra19.txt
    - nl_muestra20.txt

El *script* `etiquetadorPOSV2Prueba.py` ingresa cada archivo de muestra `.txt` a Freeling para ser etiquetado gramaticalmente y así obtener su *archivo JSON* correspondiente. Dependiendo del idioma que se esté procesando, `etiquetadorPOSV2Prueba.py` carga una instancia de Freeling configurada para analizar un idioma en especial.

Para el caso de holandés, Freeling no cuenta con el módulo correspondiente, por lo que se tomó la decisión de procesarlo como si fuera inglés, tomando en cuenta noción ya mencionada.

Los *archivos JSON* correspondientes a la muestras de prueba son los siguientes:

- Español
    - es_muestra1_freeling.json
    - es_muestra2_freeling.json
    - es_muestra3_freeling.json

- es_muestra4_freeling.json
- es_muestra5_freeling.json

- Inglés
  - en_muestra6_freeling.json
  - en_muestra7_freeling.json
  - en_muestra8_freeling.json
  - en_muestra9_freeling.json
  - en_muestra10_freeling.json

- Italiano
  - it_muestra11_freeling.json
  - it_muestra12_freeling.json
  - it_muestra13_freeling.json
  - it_muestra14_freeling.json
  - it_muestra15_freeling.json

- Holandés
  - nl_muestra16_freeling.json
  - nl_muestra17_freeling.json
  - nl_muestra18_freeling.json
  - nl_muestra19_freeling.json
  - nl_muestra20_freeling.json

Generación de POS

`PanPrueba.py` → `generadorPOSV2Prueba.py`

El *script* `generadorPOSV2Prueba.py` convierte la cadena JSON de cada *archivo JSON* de las muestras de prueba en un *archivo POS*; durante este proceso, son aplicadas las reglas de re-etiquetado señaladas en la Tabla 4.21.

Los *archivos POS* correspondientes son los siguientes:

- Español
  - es_muestra1_freeling.pos
  - es_muestra2_freeling.pos
  - es_muestra3_freeling.pos

- es_muestra4_freeling.pos
- es_muestra5_freeling.pos

- Inglés
  - en_muestra6_freeling.pos
  - en_muestra7_freeling.pos
  - en_muestra8_freeling.pos
  - en_muestra9_freeling.pos
  - en_muestra10_freeling.pos

- Italiano
  - it_muestra11_freeling.pos
  - it_muestra12_freeling.pos
  - it_muestra13_freeling.pos
  - it_muestra14_freeling.pos
  - it_muestra15_freeling.pos

- Holandés
  - nl_muestra16_freeling.pos
  - nl_muestra17_freeling.pos
  - nl_muestra18_freeling.pos
  - nl_muestra19_freeling.pos
  - nl_muestra20_freeling.pos

Generación de vector general

PanPrueba.py → generadorVectoresV3Prueba.py → generadorNGramasCPPrueba.py

Dependiendo del idioma de las muestras a clasificar, PanPrueba.py selecciona los *parámetros de extracción* establecidos en la Tabla 4.23 y posteriormente, invoca al *script* generadorVectoresV3Prueba.py.

Por cada muestra de prueba, el *script* generadorVectoresV3Prueba.py se encarga de las siguientes tareas:

- Cargar el vector de características del sistema *(vSistema)*
- Invocar al *script* generadorNGramasCPPrueba.py para obtener los vectores de características $vCAR_m$ y $vPOS_m$ de la muestra

- Asignar los valores obtenidos de $vCAR_m$ y $vPOS_m$ en *vSistema*

Prueba

`PanPrueba.py`

Por cada muestra de prueba, el *script* `PanPrueba.py` realiza lo siguiente:

- Cargar el modelo de aprendizaje entrenado ϕ
- Ingresar el *vSistema* de la muestra al modelo ϕ para obtener la predicción del sistema
- Generar el archivo `.xml` de salida requerido para el concurso

Fue requerido que cada predicción realizada por el sistema se guardara en un archivo `.xml` indicando el nombre de la muestra, el tipo, su idioma y las predicciones realizadas, mostrándose la estructura del mismo a continuación:

```
<author id="muestra#"
    type="twitter"
    lang="[es|en|it|nl]"
    age_group="[18-24|25-34|35-49|50-xx]|XX-XX"
    gender="[male|female]"
    extroverted="#"
    stable="#"
    agreeable="#"
    conscientious="#"
    open="#"
/>
```

Una vez obtenidas las predicciones en sus correspondientes archivos, se procedió a compararlas con los valores reales de las clases a las que pertenecían, obteniendo los resultados mostrados en el Capítulo 5 *Resultados*.

Capítulo 5

Resultados

5.1. Resultados obtenidos

La medida de desempeño empleada por los organizadores del concurso fue la exactitud. Esta medida obtiene la relación entre la cantidad de muestras correctamente clasificadas y la cantidad total de muestras, obteniendo un valor entre 0 (0 % de exactitud) y 1 (100 % de exactitud).

$$exactitud = \frac{muestras\_correctamente\_clasificadas}{total\_de\_muestras} \tag{5.1}$$

Para ejemplificar la medida de *exactitud* se partirá de la Figura 5.1.

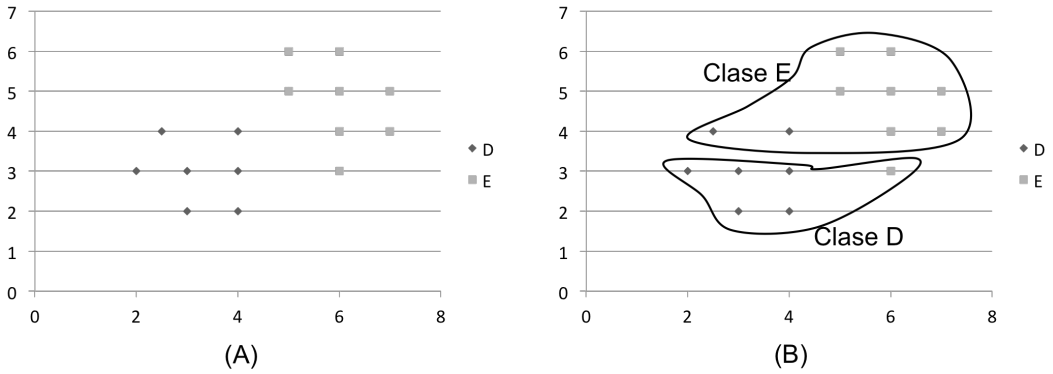

Figura 5.1: Ejemplo de exactitud

En (A) se observa un total de 15 muestras, siete pertenecientes a la clase D y ocho pertenecientes a la clase E. En (B) las muestras han sido clasificadas de forma automática a partir de un algoritmo de aprendizaje. Al analizar los resultados de la clasificación automática, es posible visualizar que de las 15 muestras, 12 fueron

clasificadas de forma correcta y tres, de forma incorrecta. La siguiente matriz de confusión[1] describe a detalle la relación entre las muestras clasificadas correctamente y las clasificadas incorrectamente.

	D	E
D	5	2
E	1	7

A partir de la matriz de confusión es posible visualizar aquellas muestras clasificadas correctamente (diagonal principal) y aquellas muestras clasificadas de forma incorrecta. Finalmente, la exactitud es calculada de la siguiente forma:

$$\begin{aligned} exactitud &= \frac{5+7}{5+2+1+7} \\ exactitud &= 0.8 \end{aligned}$$

Para automatizar la recolección de resultados se creó el *script* de Python `evaluador.py`, el cual lee todos los archivos `.xml` generados durante la fase de prueba y compara su contenido con los valores reales de las clases de las muestras analizadas; dando como resultado las matrices de confusión y la exactitud del sistema.

- Español

```
./es_salida
GENERO: 0.9
[[14  1]
 [ 2 13]]
......
EDAD: 0.8
[[ 6  0  0  0]
 [ 2 12  0  0]
 [ 0  2  4  0]
 [ 0  2  0  2]]
......
GENERO-EDAD: 0.733333333333
[[3 0 0 0 0 0 0 0]
 [1 5 0 0 0 1 0 0]
 [0 0 3 0 0 0 0 0]
 [0 1 0 1 0 0 0 0]
 [0 0 0 0 3 0 0 0]
```

[1]En el Capítulo 8 *Anexo* se describen las matrices de confusión

```
 [0 1 0 0 1 5 0 0]
 [0 1 0 0 0 1 1 0]
 [0 0 0 0 0 1 0 1]]
RMSE
EXTROVERTED: 0.106458129484
STABLE: 0.127801930085
OPEN: 0.137840487521
CONSCIENTIOUS: 0.164316767252
AGREEABLE: 0.158113883008
PROMEDIO RMSE: 0.13890623947
GLOBAL RANKING 0.797213546932
```

- Inglés

```
./en_salida/
GENERO: 0.826086956522
[[19  4]
 [ 4 19]]
......
EDAD: 0.847826086957
[[17  1  0  0]
 [ 2 15  1  0]
 [ 0  2  4  0]
 [ 0  1  0  3]]
......
GENERO-EDAD: 0.717391304348
[[7 0 0 0 2 0 0 0]
 [1 7 0 0 0 1 0 0]
 [0 1 1 0 0 1 0 0]
 [0 0 0 2 0 0 0 0]
 [3 0 0 0 5 1 0 0]
 [0 0 0 0 1 7 1 0]
 [0 0 0 0 0 0 3 0]
 [0 1 0 0 0 0 0 1]]
RMSE
EXTROVERTED: 0.181778651829
STABLE: 0.181778651829
OPEN: 0.16218615177
CONSCIENTIOUS: 0.123358790949
AGREEABLE: 0.149637242516
PROMEDIO RMSE: 0.159747897779
GLOBAL RANKING 0.778821703285
```

- Italiano

```
./salida_it/
GENERO: 1.0
[[6 0]
 [0 6]]
......
EDAD: 1.0
[[12]]
......
GENERO-EDAD: 1.0
[[6 0]
 [0 6]]
RMSE
EXTROVERTED: 0.0645497224368
STABLE: 0.19364916731
OPEN: 0.111803398875
CONSCIENTIOUS: 0.1
AGREEABLE: 0.0912870929175
PROMEDIO RMSE: 0.112257876308
GLOBAL RANKING 0.943871061846
```

- Holandés

```
./nl_salida/
GENERO: 0.9
[[4 1]
 [0 5]]
......
EDAD: 1.0
[[10]]
......
GENERO-EDAD: 0.9
[[4 1]
 [0 5]]
RMSE
EXTROVERTED: 0.118321595662
STABLE: 0.161245154966
OPEN: 0.118321595662
CONSCIENTIOUS: 0.0316227766017
AGREEABLE: 0.144913767462
PROMEDIO RMSE: 0.114884978071
GLOBAL RANKING 0.892557510965
```

5.2. Análisis de resultados

Las matrices de confusión mostradas en la sección anterior son de gran utilidad para poder lograr un análisis de los resultados obtenidos. A continuación, se muestra una interpretación de los resultados obtenidos:

- Español
  Número de muestras a probar: 30
  Como se ve en la matriz de confusión referente a la predicción de género en español,

	F	M
F	14	1
M	2	13

  el sistema es bastante bueno para identificar el género de un usuario de Twitter ya que al calcular su exactitud, se obtiene que en un 90 % de las veces realiza una predicción correcta.

$$\begin{aligned} genero\_esp &= \frac{14+13}{14+1+2+13} \\ genero\_esp &= 0.9 \end{aligned}$$

  Obteniendo la exactitud de forma independiente para mujeres y hombres,

$$\begin{aligned} & mujeres & \\ genero\_es_F &= & \frac{14}{14+1} \\ genero\_es_F &= & 0.933 \\ & hombres & \\ genero\_es_M &= & \frac{13}{13+2} \\ genero\_es_M &= & 0.86 \end{aligned}$$

  se observa que el sistema es ligeramente mejor reconociendo mujeres *(93.3 %)* que hombres *(86 %)*.

  Con respecto a la matriz de confusión referente a la predicción de edad en español,

	20s	*30s*	*40s*	*50s*
20s	6	0	0	0
30s	2	12	0	0
40s	0	2	4	0
50s	0	2	0	2

el sistema muestra que en un 80 % de las veces realiza una predicción correcta:

$$\begin{aligned} edad\_es &= \frac{6+12+4+2}{6+2+12+2+4+2+2} \\ edad\_es &= 0.8 \end{aligned}$$

Observando a detalle el desempeño en cada una de las cuatro clases pertenecientes a la edad,

$$\begin{aligned} & 20s \\ edad\_es_{20s} &= \frac{6}{6} \\ edad\_es_{20s} &= 1.0 \\ & 30s \\ edad\_es_{30s} &= \frac{12}{2+12} \\ edad\_es_{30s} &= 0.857 \\ & 40s \\ edad\_es_{40s} &= \frac{4}{2+4} \\ edad\_es_{40s} &= 0.667 \\ & 50s \\ edad\_es_{50s} &= \frac{2}{2+2} \\ edad\_es_{50s} &= 0.5 \end{aligned}$$

es posible concluir que el sistema es muy bueno identificando personas con un rango de 18 a 24 años; pero es muy malo al predecir personas con más de 49 años. En el caso de las personas con una edad entre los 35 y los 49 años, una de cada tres veces comete el error de clasificarlos dentro del rango de los 25 a los 34 años. Finalmente, una de cada siete veces comete el error de clasificar como *20s* a personas con un rango de edad de entre los 24 y los 34 años *(30s)*.

Al analizar la matriz de confusión, en donde se combinan ambos rasgos,

	F_20s	*F_30s*	*F_40s*	*F_50s*	*M_20s*	*M_30s*	*M_40s*	*M_50s*
F_20s	3	0	0	0	0	0	0	0
F_30s	1	5	0	0	0	1	0	0
F_40s	0	0	3	0	0	0	0	0
F_50s	0	1	0	1	0	0	0	0
M_20s	0	0	0	0	3	0	0	0
M_30s	0	1	0	0	1	5	0	0
M_40s	0	1	0	0	0	1	1	0
M_50s	0	0	0	0	0	1	0	1

se obtiene una exactitud de 0.73, lo que significa que el sistema hace una predicción correcta al indicar el género y el rango de edad de una persona el 73 % de las veces.

Analizando cada una de las ocho clases existentes,

$$
\begin{aligned}
& F\_20s \\
genero\_edad\_es_{F\_20s} &= \frac{3}{3} \\
genero\_edad\_es_{F\_20s} &= 1.0 \\
& F\_30s \\
genero\_edad\_es_{F\_30s} &= \frac{5}{1+5+1} \\
genero\_edad\_es_{F\_30s} &= 0.714 \\
& F\_40s \\
genero\_edad\_es_{F\_40s} &= \frac{3}{3} \\
genero\_edad\_es_{F\_40s} &= 1.0 \\
& F\_50s \\
genero\_edad\_es_{F\_50s} &= \frac{1}{1+1} \\
genero\_edad\_es_{F\_50s} &= 0.5 \\
& M\_20s \\
genero\_edad\_es_{M\_20s} &= \frac{3}{3} \\
genero\_edad\_es_{M\_20s} &= 1.0 \\
& M\_30s \\
genero\_edad\_es_{M\_30s} &= \frac{5}{1+5+1} \\
genero\_edad\_es_{M\_30s} &= 0.714 \\
& M\_40s \\
genero\_edad\_es_{M\_40s} &= \frac{1}{1+1+1} \\
genero\_edad\_es_{M\_40s} &= 0.333
\end{aligned}
$$

$$
\begin{aligned}
& & M\_50s & \\
genero\_edad\_es_{M\_50s} & = & \frac{1}{1+1} & \\
genero\_edad\_es_{M\_50s} & = & 0.5 &
\end{aligned}
$$

se observa que el sistema acierta el 100 % de las veces al identificar hombres y mujeres con un rango de edad de entre los 18 y los 24 años; así como mujeres entre los 35 y los 49 años. Es importante recalcar la confusión que se presenta al identificar personas de entre 25 y 34 años, ya que tanto en hombres como en mujeres una de cada siete veces se predice un rango de edad menor; así como el género opuesto. A partir de esta observación, podría concluirse que existen elementos que hombres y mujeres pertenecientes a este rango de edad comparten de forma representativa.

- Inglés

  Número de muestras a probar: 46
  El primer elemento a analizar es la matriz de confusión referente a la predicción de género:

	F	M
F	19	4
M	4	19

A partir de la matriz es posible visualizar que el desempeño del sistema es bueno, ya que el grueso de los valores se encuentra en la diagonal principal. Al obtener su exactitud,

$$
\begin{aligned}
genero\_en & = & \frac{19+19}{19+4+4+19} \\
genero\_en & = & 0.826
\end{aligned}
$$

resulta que el 82.6 % de las veces realiza una predicción de género de forma correcta. Obteniendo la exactitud de forma independiente para mujeres y hombres,

$$
\begin{aligned}
& & mujeres & \\
genero\_en_F & = & \frac{19}{19+4} & \\
genero\_en_F & = & 0.826 & \\
& & hombres & \\
genero\_en_M & = & \frac{19}{19+4} & \\
genero\_en_M & = & 0.826 &
\end{aligned}
\tag{5.2}
$$

se observa que el sistema es bueno identificando hombres *(82.6 %)* y mujeres *(82.6 %)*.

Con relación a la predicción de edad, al observar su matriz de confusión,

	20s	*30s*	*40s*	*50s*
20s	17	1	0	0
30s	2	15	1	0
40s	0	2	4	0
50s	0	1	0	3

y posteriormente, calcular su exactitud,

$$\begin{aligned} edad\_en &= \frac{17+15+4+3}{17+1+2+15+1+2+4+1+3} \\ edad\_en &= 0.848 \end{aligned}$$

se aprecia que el sistema realiza una predicción correcta sobre la edad de las personas en un 85 % de las ocasiones.

Al obtener la exactitud de las cuatro clases que componen a la edad y analizando su matriz de confusión,

$$\begin{aligned} & 20s \\ edad\_en_{20s} &= \frac{17}{17+1} \\ edad\_en_{20s} &= 0.944 \\ & 30s \\ edad\_en_{30s} &= \frac{15}{2+15+1} \\ edad\_en_{30s} &= 0.833 \\ & 40s \\ edad\_en_{40s} &= \frac{4}{2+4} \\ edad\_en_{40s} &= 0.667 \\ & 50s \\ edad\_en_{50s} &= \frac{3}{1+3} \\ edad\_en_{50s} &= 0.75 \end{aligned}$$

se puede apreciar que la mayoría de los errores (6 de 7) son cometidos en clases adyacentes a la clase real.

Observando la matriz de confusión de la clasificación conjunta y sus valores de exactitud,

	F_20s	*F_30s*	*F_40s*	*F_50s*	*M_20s*	*M_30s*	*M_40s*	*M_50s*
F_20s	7	0	0	0	2	0	0	0
F_30s	1	7	0	0	0	1	0	0
F_40s	0	1	1	0	0	1	0	0
F_50s	0	0	0	2	0	0	0	0
M_20s	3	0	0	0	5	1	0	0
M_30s	0	0	0	0	1	7	1	0
M_40s	0	0	0	0	0	0	3	0
M_50s	0	1	0	0	0	0	0	1

$$
\begin{aligned}
& & F\_20s & \\
genero\_edad\_en_{F\_20s} & = & \frac{7}{7+2} & \\
genero\_edad\_en_{F\_20s} & = & 0.778 & \\
& & F\_30s & \\
genero\_edad\_en_{F\_30s} & = & \frac{7}{1+7+1} & \\
genero\_edad\_en_{F\_30s} & = & 0.778 & \\
& & F\_40s & \\
genero\_edad\_en_{F\_40s} & = & \frac{1}{1+1+1} & \\
genero\_edad\_en_{F\_40s} & = & 0.333 & \\
& & F\_50s & \\
genero\_edad\_en_{F\_50s} & = & \frac{2}{2} & \\
genero\_edad\_en_{F\_50s} & = & 1.0 & \\
& & M\_20s & \\
genero\_edad\_en_{M\_20s} & = & \frac{5}{3+5+1} & \\
genero\_edad\_en_{M\_20s} & = & 0.556 & \\
& & M\_30s & \\
genero\_edad\_en_{M\_30s} & = & \frac{7}{1+7+1} & \\
genero\_edad\_en_{M\_30s} & = & 0.778 & \\
& & M\_40s & \\
genero\_edad\_en_{M\_40s} & = & \frac{3}{3} & \\
genero\_edad\_en_{M\_40s} & = & 1.0 & \\
& & M\_50s &
\end{aligned}
$$

$$
\begin{aligned}
genero\_edad\_en_{M\_50s} &= \frac{1}{1+1} \\
genero\_edad\_en_{M\_50s} &= 0.5
\end{aligned}
$$

se observa que hombres y mujeres con un rango de edad entre los 18 y los 24 años comparten características que hacen que el sistema falle, ya que una tercera parte de las predicciones en este rango de edad (5 de 18) son clasificadas con el género incorrecto, comportamiento similar al presentado en español y la confusión al predecir personas entre 25 y 34 años.

- Italiano
  Número de muestras clasificadas: 12

  A diferencia de las dos lenguas ya presentadas, el corpus del italiano se encuentra etiquetado únicamente por género, razón por la cual sólo puede ser posible el análisis de este rasgo.

	F	M
F	6	0
M	0	6

  La matriz de confusión mostrada refleja que el sistema es muy bueno diferenciando hombres y mujeres con un valor de exactitud igual a 1.0. Es posible que este valor sea tan elevado debido a las pocas muestras por clasificar.

- Holandés

  Número de muestras clasificadas: 10

  El caso del holandés es similar al italiano; el corpus sólo se encuentra etiquetado por género, por lo cual sólo es posible realizar el análisis respectivo.

	F	M
F	4	1
M	0	5

$$
\begin{aligned}
genero\_nl &= \frac{4+5}{4+1+5} \\
genero\_nl &= 0.9
\end{aligned}
$$

Revisando la matriz de confusión, se puede apreciar que el sistema es ligeramente mejor al identificar hombres, pero esta apreciación podría no ser correcta debido a la poca cantidad de muestras.

Los resultados presentados muestran el rendimiento de la presente propuesta en diferentes lenguas, reflejando un desempeño extremadamente alto en el caso del italiano y del holandés, con una exactitud del 100 % y del 90 % respectivamente. La razón de esos valores de exactitud se debe a dos elementos: en primer lugar, para estos dos idiomas únicamente se realizó una clasificación por género, teniendo así únicamente dos clases (*hombre* y *mujer*) a diferencia de las ocho clases existentes para español e inglés; en segundo lugar la cantidad de muestras disponibles fue muy poca, haciendo que el sistema aprendiera patrones muy específicos durante la etapa de entrenamiento, patrones que las muestras de prueba conservaron, dando como resultado predicciones acertadas.

Es preciso prestar atención a los rangos de edad de 18 a 24 años ($20s$) y de 25 a 39 años ($30s$), ya que en esas clases es donde el sistema llega a clasificar erróneamente el género de la persona.

Para determinar el rendimiento general del sistema se calculó el promedio de las exactitudes obtenidas en los diferentes idiomas. La exactitud promedio en la predicción de género es obtenida a partir de los cuatro idiomas. En la Tabla 5.1 se muestra la exactitud de cada lengua, así como el promedio general.

Tabla 5.1: Exactitud promedio en la predicción de género

Idioma	*Exactitud*
español	0.9
inglés	0.826
italiano	1.0
holandés	0.9
PROMEDIO	0.907

De forma similar, se calculó la exactitud promedio en la predicción de edad. En este caso, y por los motivos ya mencionados, únicamente se realizó el cálculo sobre español e inglés, como se aprecia en la Tabla 5.2.

Tabla 5.2: Exactitud promedio en la predicción de edad

Idioma	*Exactitud*
español	0.8
inglés	0.848
PROMEDIO	0.824

Finalmente, la exactitud promedio del sistema es obtenida a partir de la clasificación conjunta de edad y género en el caso de español e inglés, y de la clasificación de género, en el caso de italiano y holandés.

Tabla 5.3: Exactitud promedio del sistema

Idioma	*Exactitud*
español	0.733
inglés	0.717
italiano	1.0
holandés	0.9
PROMEDIO	0.838

Capítulo 6

Participación en PAN2015

Los resultados ya mencionados hacen referencia a la evaluación que fue realizada para la presente investigación, la cual se obtuvo a partir de la segmentación del corpus de entrenamiento descrito en el Capítulo 4 *Clasificación automática de textos*. Durante la fase de evaluación del concurso, el rendimiento de los sistemas fue medido utilizando un corpus de evaluación (también descrito en el Capítulo 4 *Clasificación automática de textos*). En esta fase, la presente propuesta obtuvo el segundo lugar general de 22 participantes; siendo tercer lugar en español, segundo en inglés, primero en italiano y segundo en holandés.

A continuación, se presenta una breve descripción de los artículos realizados por los otros siete mejores competidores que componen, junto con la presente propuesta, las cinco mejores propuestas de cada uno de los cuatro idiomas. Posteriormente, se muestra una comparación de resultados entre estas siete propuestas y la presente.

6.1. INAOE's participation at PAN'15: Author Profiling task. Notebook for PAN at CLEF 2015

La propuesta realizada por Álvarez-Carmona *et al.* [Carmona et al., 2015] basa su funcionamiento en la representación de características de alto nivel de los tuits. Para ello, se hace una división de las características del texto, separándolas en discriminatorias y descriptivas.

Por características discriminatorias se refieren a aquellas que proveen pistas relacionadas con el estilo del autor del texto. A diferencia del enfoque tradicional de obtener palabras funcionales o signos de puntuación, se extraen Atributos de Segundo Orden (SOA) [Lopez-Monroy, 2013]. La idea detrás de SOA es representar vectores de documentos en un espacio de perfiles; bajo esa representación, cada valor representa la relación entre cada documento con cada perfil.

Las características descriptivas hacen relación a la información temática relevante que se obtiene a partir de ciertos términos que están contenidos en el texto. Para la extracción de estos términos, se utiliza un método llamado Análisis de Semántica Latente (LSA) [Wiemer-Hastings et al., 2004]; este análisis reduce la dimensionalidad de las características, identificando aquellas palabras que mejor representan los textos para generar conceptos. A partir de estas dos representaciones, un vector general de características es generado para hacer el entrenamiento del sistema.

Para realizar la caracterización de los tuits, siete clasificadores son entrenados: uno para identificar género, un segundo para detectar edad, y cinco para predecir los diferentes rasgos de personalidad. El algoritmo de aprendizaje seleccionado para los siete clasificadores es SVM.

6.2. Author profiling using stylometric and structural feature groupings. Notebook for PAN at CLEF 2015

En esta propuesta, realizada por Grivas *et al.* [Grivas et al., 2015], se aborda el problema del perfilado de autor diferenciando dos grupos de características: estructurales y estilométricas. Las características estructurales son extraídas de los tuits sin pre-procesar; mientras que para las características estilométricas, los tuits son preprocesados, eliminando todas las etiquetas `html` y toda información sintáctica provista por Twitter (*hashtags*, referencias a otros usuarios, *urls*, etc.).

La intención del grupo de características estructurales es capturar aquellas características del texto que tengan interdependencia con el uso de Twitter (la frecuencia de referencias a otros usuarios, *hashtags*, *urls*, etc.). Por otro lado, el grupo de características estilométricas intenta capturar aquellas características de contexto que un usuario genera de forma no automática; en este grupo se encuentran los 3-gramas Tf-idf, bolsa de 3-gramas, bolsa de palabras, Tf-idf de palabras, frecuencia de palabras en mayúsculas y frecuencia de palabras de entre 1 y 20 caracteres.

La detección de edad, género y personalidad es abordada de forma independiente. Edad y género son considerados problemas de clasificación, mientras que la detección de los rasgos de personalidad es considerado un problema de regresión. En ambos casos el algoritmo de aprendizaje utilizado es SVM.

6.3. UniNE at CLEF 2015: Author Profiling. Notebook for PAN at CLEF 2015

Kocher [Kocher and Savoy, 2015] propone la implementación de un modelo de perfilado de autor basado en distancias relativas entre los perfiles de entrenamiento y los perfiles a predecir. La cantidad de perfiles de entrenamiento es la cantidad de clases existentes.

El grupo de características utilizado pertenece al grupo estilístico, siendo los 200 términos más frecuentes los utilizados para caracterizar los documentos (tuits). Para disminuir el número de términos con una sola ocurrencia, todos los *hashtags* diferentes son tomados como uno solo, al igual que las *urls*. Cada perfil está constituido por la probabilidad de ocurrencia de los 200 términos más frecuentes.

El modelo propuesto basa su funcionamiento en la distancia que existe entre los 200 términos del perfil de entrenamiento a comparar y los 200 términos del perfil a predecir. A continuación, se presenta la fórmula empleada:

$$\Delta(Q, A) = \sum_{i=1}^{k} |P_Q[t_i] - P_A[t_i]|$$

en donde k es el número de términos (200), $P_Q[t_i]$ la probabilidad del término t_i en el perfil a predecir y $P_A[t_i]$ la probabilidad del término t_i en el perfil de entrenamiento a comparar. Para obtener las probabilidades se divide la frecuencia del término entre la cantidad de términos en el perfil.

6.4. Automatic Profiling of Twitter Users Based on Their Tweets. Notebook for PAN at CLEF 2015

Sulea *et al.* [Sulea and Dichiu, 2015] abordan el problema de perfilado de autor proponiendo una combinación de características: Tf-idf a nivel de caracteres y la relación tipo/*token* de un autor. La detección de edad y género es tratado como un problema de clasificación utilizando SVM, mientras que los cinco rasgos de personalidad son considerados problemas de regresión, haciendo uso de un regresor lineal.

Para la detección de edad y género, una matriz Tf-idf de caracteres es creada, en esta matriz, las columnas corresponden a los n-gramas de caracteres de todos los tuits y cada renglón corresponde al valor Tf-idf de todos los tuits pertenecientes a un

autor. De forma similar, para la predicción de los rasgos de personalidad, se crea una matriz Tf-idf, con la diferencia de que los renglones corresponden a tuits aislados.

La relación tipo/*token* de un autor es obtenida a partir de la relación entre el número de raíces únicas y el número de palabras usadas una vez aplicado *steming*; siendo excluidas las palabras funcionales.

6.5. What do your look-alikes say about you? Exploiting strong and weak similarities for author profiling. Notebook for PAN at CLEF 2015

Przybyla *et al.* [Przybyla and Teisseyre, 2015] se basan en la premisa de que aquellos autores que utilizan vocabulario similar tienden a tener rasgos similares. Los autores dividen las características utilizadas en dos grupos: basadas en palabra (*word-based*) y basadas en texto (*text-based*).

Las características basadas en palabra hacen referencia al número de ocurrencias de los lemas de los tuits, mientras que las basadas en texto contienen aquellas características obtenidas de forma estadística del texto en general: longitud promedio del tuit (número de caracteres), longitud promedio de las palabras, número promedio de *urls*, *hashtags*, referencia a otros usuarios, letras mayúsculas, signos de exclamación e interrogación, emoticonos positivos y negativos, letras repetidas, signos de exclamación e interrogación repetidos, números y errores de ortografía por tuit. Así como lexicones asociados a sentimientos y emociones.

La predicción de género, edad y rasgos de personalidad es realizada en dos fases: en primera instancia, en el corpus de entrenamiento se busca la muestra que sea más parecida a la muestra a predecir. Si la similitud obtenida es mayor a cierto umbral, la muestra en cuestión es asignada a las mismas clases que la muestra de entrenamiento. En caso de que la similitud no sea mayor al umbral establecido, el género y la edad son tratados de forma independiente utilizando árboles de decisión, mientras que para la predicción de los rasgos de personalidad se utilizan árboles de regresión.

6.6. XRCE Personal Language Analytics Engine for Multilingual Author Profiling. Notebook for PAN at CLEF 2015

Nowson *et al.* [Nowson et al., 2015] separan su sistema en cuatro etapas diferentes: pre-procesado, extracción de características, descomposición de valor singular y mo-

delos de clasificación.

Durante la etapa de pre-procesado, Nowson *et al.* implementan todo un conjunto de herramientas que efectúan un análisis lingüístico de los tuits: tokenización, análisis morfosintáctico, etiquetado POS, detección de entidades nombradas y extracción de relaciones de dependencias. Este análisis lingüístico es efectuado sobre los cuatro idiomas (español, inglés, italiano y holandés), siendo más efectivo en inglés debido a que el desarrollo de las herramientas para esta lengua está más avanzado. Para el caso de inglés, se utiliza un diccionario para normalizar el vocabulario dentro de la red social así como un lexicón de polaridad y una capa para realizar análisis de sentimientos. De la mano con el análisis lingüístico, un análisis enfocado a Twitter es efectuado para lograr extraer *hashtags*, referencias a otros usuarios y emoticonos con su respectiva polaridad.

La extracción de características considera dos tipos de características: a nivel palabra y basadas en la clase. Dentro de las características a nivel palabra se incluyen los 1-gramas, 2-gramas y 3-gramas de palabras y lemas; etiquetas POS de palabras y lemas; negaciones de palabras; palabras con al menos tres letras repetidas y 2-gramas, 3-gramas y 4-gramas de caracteres repetidos. Las características basadas en la clase considera entidades nombradas; 1-gramas, 2-gramas y 3-gramas de etiquetas POS; emoticonos positivos, negativos y otros; *hashtags*, referencias a otros usuarios y *urls*; uso de nombres y pronombres femeninos y masculinos y palabras con mayúsculas.

La traducción automática es utilizada para aquellos idiomas en donde hay pocos datos de entrenamiento, y así buscar aumentar el rendimiento del sistema en ellos. El inconveniente con este proceso es que los marcadores sociodemográficos tienden a perderse al realizar la traducción automática.

Finalmente, la descomposición de valor singular es utilizada para comprimir la cantidad de características que compone cada vector, posteriormente un conjunto de diez clasificadores es entrenado y usado para realizar la predicción de cada rasgo.

6.7. Statistical Learning Methods for Profiling Analysis. Notebook for PAN at CLEF 2015

Miculicich [Werlen, 2015] aborda el problema de perfilado de autor tratando la identificación de género, edad y rasgos de personalidad como problemas de clasificación. Es posible separar las características utilizadas por Miculicich en tres grupos: categorías de palabras, signos de puntuación y elementos propios de Twitter.

Las categorías de palabras son obtenidas mediante LIWC para saber qué tan frecuentemente un autor utiliza cada categoría de palabra y con esos datos estimar cierta información sobre el mismo. Los signos de puntuación componen el segundo grupo de categorías y está conformado por signos de interrogación y de exclamación, puntos,

comas, puntos y comas y otros signos de puntuación. El tercer grupo está compuesto por aquellos elementos propios de Twitter como los emoticonos, *urls*, *hashtags* y referencias a otros usuarios.

Para hacer la selección de características, Miculicich utiliza Eliminación Recursiva de características mediante *SVM Recursive Feature Elimination* en el caso de la predicción de género y edad; para los rasgos de personalidad *Forward-Backward Feature Selection* es implementado.

Las SVM son utilizadas como clasificadores para la predicción de género, edad y rasgos de personalidad. Adicionalmente, para este último LDA es implementado como un segundo clasificador.

6.8. Comparación de resultados

La tarea de perfilado de autor consistió en predecir el género, la edad y cinco rasgos de personalidad de un usuario de Twitter [Rangel et al., 2015]. Para evaluar el desempeño en la predicción de género y edad, se utilizó la medida de exactitud, mientras que *RMSE* fue utilizado para los cinco rasgos de personalidad. Finalmente, una medida global fue empleada para agrupar todas las predicciones:

$$GLOBAL = \frac{(1 - RMSE) + exactitud\_edad\_y\_genero}{2}$$

A continuación, se presentan los resultados de los cinco mejores resultados de cada idioma en la tarea de *Author Profiling PAN2015*, y posteriormente, una tabla de resultados generales.

- Español

  En la Tabla 6.1 es posible apreciar que ninguna propuesta supera el 77 % de exactitud en la predicción de edad y género, siendo *Alvarez-Carmona* la propuesta que alcanza una mayor exactitud. En cuanto a los rasgos de personalidad, la propuesta de *Kocher* obtuvo el menor valor de RMSE (0.1235), mientras que *Sulea* obtuvo el mayor. Es preciso recordar que un menor valor de RMSE significa que las predicciones estuvieron menos alejadas de la diagonal principal dentro de la matriz de confusión, lo que se traduce en un sistema con errores menos pronunciados.

  Álvarez-Carmona obtuvo el mejor resultado global con un valor de 0.8215 y un tiempo de procesamiento de 44 segundos. La presente propuesta obtuvo un valor global de 0.7745 y un tiempo de procesamiento de 4 minutos y 25 segundos.

Tabla 6.1: Resultados PAN2015 para español

lugar	*participante*	*edad&genero*	*personalidad*	*GLOBAL*	*tiempo*
1	*Alvarez-Carmona*	0.7727	0.1297	0.8215	00:00:44
2	*Kiprov*	0.7273	0.1495	0.7889	00:02:46
3	*Gonzalez-Gallardo*	0.7045	0.1555	0.7745	00:04:25
4	*Kocher*	0.6705	0.1235	0.7735	00:00:02
5	*Sulea*	0.6591	0.1599	0.7496	00:01:39

- Inglés

  Para el caso de inglés, *Alvarez-Carmona* obtuvo el mejor rendimiento de todos los participantes, con un valor global de 0.7906 obtenido a partir de una exactitud del 72.5 % en la predicción de edad y género, y un valor de 0.1442 en los rasgos de personalidad. De igual forma, el tiempo de procesamiento fue el menor siendo de casi un minuto. En segundo lugar, se encuentra la presente propuesta con una exactitud en la predicción de género y edad del 69.7 % y un valor RMSE ligeramente mayor al obtenido por *Alvarez-Carmona*; los seis minutos y medio del tiempo de procesamiento se deben muy probablemente a la gran cantidad de tuits que tienen que ser analizados por Freeling (Tabla 6.2).

Tabla 6.2: Resultados PAN2015 para inglés

lugar	*participante*	*edad&genero*	*personalidad*	*GLOBAL*	*tiempo*
1	*Alvarez-Carmona*	0.7254	0.1442	0.7906	00:00:59
2	*Gonzalez-Gallardo*	0.6972	0.1491	0.7740	00:06:29
3	*Teisseyre*	0.6479	0.1500	0.7489	00:03:15
4	*Grivas*	0.6690	0.1716	0.7487	00:01:53
5	*Sulea*	0.6197	0.1442	0.7378	00:02:44

- Italiano

  Los resultados globales para italiano (Tabla 6.3) oscilan entre 0.8089 y 0.8658, siendo la presente propuesta la que mejor rendimiento refleja. *Grivas*, *Nowson* y *Kocher* muestran resultados globales similares entre sí (alrededor del 0.82). Es importante recalcar el tiempo de procesamiento de la propuesta presentada por *Kocher* basada en distancias relativas: 1 segundo.

- Holandés

  Los resultados de holandés (Tabla 6.4) son en general bastante buenos, siendo la propuesta de *Grivas* la que obtuvo mayor exactitud (96.88 %) en la predicción de género, pero el valor más alto de RMSE (0.1571). El valor global mayor (0.9406) fue obtenido por *Alvarez-Carmona*, compartiendo la misma exactitud en la predicción de género con la presente propuesta (93.75 %).

Tabla 6.3: Resultados PAN2015 para italiano

lugar	*participante*	*genero*	*personalidad*	*GLOBAL*	*tiempo*
1	*Gonzalez-Gallardo*	0.8611	0.1294	0.8658	00:01:31
2	*Grivas*	0.8333	0.1743	0.8295	00:00:29
3	*Nowson*	0.8056	0.1515	0.8270	00:01:59
4	*Kocher*	0.7778	0.1259	0.8260	00:00:01
5	*Alvarez-Carmona*	0.7222	0.1044	0.8089	00:00:25

Tabla 6.4: Resultados PAN2015 para holandés

lugar	*participante*	*genero*	*personalidad*	*GLOBAL*	*tiempo*
1	*Alvarez-Carmona*	0.9375	0.0563	0.9406	00:00:24
2	*Gonzalez-Gallardo*	0.9375	0.0890	0.9242	00:01:20
3	*Grivas*	0.9688	0.1571	0.9058	00:00:29
4	*Sulea*	0.8438	0.1164	0.8637	00:00:31
5	*Miculicich*	0.8125	0.1175	0.8475	00:00:10

- Resultados Generales

  A partir de los valores globales de los cuatro idiomas, un resultado general es obtenido por cada participante siendo los mejores cinco los que se muestran en la Tabla 6.5. *Alvarez-Carmona* obtuvo el primer lugar, con un valor global de 0.8404; seguido por la presente propuesta, presentando un valor de 0.8346. Para visualizar los resultados del resto de los participantes, es posible consultar el artículo [Rangel et al., 2015].

Tabla 6.5: Resultados Generales PAN2015

lugar	*participante*	*GLOBAL*
1	*Alvarez-Carmona*	0.8404
2	*Gonzalez-Gallardo*	0.8346
3	*Grivas*	0.8075
4	*Kocher*	0.7875
5	*Sulea*	0.7755

Capítulo 7

Conclusiones y trabajo futuro

El perfilado de autor ha sido utilizado en gran medida en textos literarios, documentos y ensayos; buscando predecir si un texto pertenece o no a un grupo de autores que comparten ciertos atributos. A partir de las redes sociales que contienen texto con características diferentes a los textos ya mencionados (longitud menor o igual a 250 palabras, vocabulario no estandarizado, libre uso de mayúsculas y de signos de puntuación, uso de emoticonos y vocabulario propio de la red social) ha surgido la necesidad de buscar aproximaciones que logren extraer la mayor cantidad de información bajo estas nuevas características.

Otro elemento a destacar es la gran cantidad de lenguas que son utilizadas en las redes sociales, por lo que es importante que los algoritmos desarrollados sean aplicables a diferentes lenguas sin tener que hacer cambios mayores.

Estos dos elementos dieron pie para desarrollar un prototipo de *software* que permitiera la clasificación automática de textos cortos mediante un algoritmo de aprendizaje, haciendo uso de características relacionadas con el género y grupo etario de una persona, *software* que debiera ser independiente del idioma y pudiera ser aplicado a corpus de diferentes redes sociales.

Para lograr que el *software* fuera independiente del idioma, fue necesario optar por características que dieran una mayor libertad, y por lo tanto, fueran independientes del idioma; estas características fueron n-gramas de caracteres y n-gramas de etiquetas gramaticales (POS).

Adicional a la selección de estas características, se decidió aprovechar la información extra que pudiese aportar la red social (en este caso Twitter); información extra como los *hashtags*, referencias a otros usuarios (*@usuario*) y *urls*. Al ser estructuras léxicas no estandarizadas, fue necesario realizar un re-etiquetado (Normalización dinámica dependiente del contexto) de las etiquetas gramaticales obtenidas mediante Freeling y así, mantener las secuencias gramaticales creadas por los usuarios.

SVM demostró ser un algoritmo de Aprendizaje de Máquina bastante acertado

para la tarea, realizando los entrenamientos necesarios de forma rápida y clasificando las muestras de prueba de forma eficiente. Merece la pena recordar que el algoritmo de Aprendizaje de Máquina es implementado en la etapa de *Entrenamiento* dentro de la *Fase de entrenamiento*, siendo el resto de las etapas independientes a ésta; por lo que es posible la utilización de otros algoritmos de Aprendizaje de Máquina sin afectar el funcionamiento del resto de las fases.

A partir de los resultados obtenidos, tanto de la evaluación realizada localmente, como de la evaluación del *PAN2015*, se puede concluir que es posible desarrollar un algoritmo capaz de clasificar automáticamente (con cierta independencia del idioma) textos cortos, aprovechando la estructura y la información sintáctica proporcionada por la red social.

El presente proyecto de investigación contribuye desde un aspecto teórico al perfilado de autor, al comprobar la utilidad del re-etiquetado gramatical que se obtiene a partir de los recursos léxicos que las redes sociales proveen. Por otro, lado existe una contribución práctica a las redes sociales al brindar una posible herramienta que ayude a detectar perfiles no deseables dentro de las mismas.

Como trabajo futuro se plantea probar con otras redes sociales y con *córpora* de mayor tamaño, así como experimentar con otros analizadores gramaticales para intentar aumentar la velocidad del sistema. Por otro lado, se tiene la intención de realizar experimentos con algún método de selección de variables, con la finalidad de comprimir el tamaño de los vectores y desechar aquellas características que no estén aportando información relevante para la clasificación.

Referencias

[Alpaydin, 2010] Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, 2nd edition.

[Argamon et al., 2009] Argamon, S., Koppel, M., Pennebaker, J. W., and Schler, J. (2009). Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119–123.

[Carmona et al., 2015] Carmona, M. Á. Á., López-Monroy, A. P., Montes-y-Gómez, M., Pineda, L. V., and Escalante, H. J. (2015). Inaoe's participation at pan'15: Author profiling task. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Cheng et al., 2011] Cheng, N., Chandramouli, R., and Subbalakshmi, K. (2011). Author gender identification from text. *Digital Investigation*, 8(1):78–88.

[Corpus, 2015] Corpus, B. N. (2015). About the bnc.

[Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

[Doyle and Kešelj, 2005] Doyle, J. and Kešelj, V. (2005). Automatic categorization of author gender via n-gram analysis. In *The 6th Symposium on Natural Language Processing, SNLP*.

[English, 2015] English, B. A. W. (2015). Bawe (british academic written english) and bawe plus collections.

[Facebook, 2015] Facebook (2015). Facebook newsroom.

[Felice and Specia, 2012] Felice, M. and Specia, L. (2012). Linguistic features for quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 96–103. Association for Computational Linguistics.

[Fernández et al., 2007] Fernández, S., SanJuan, E., and Torres-Moreno, J. M. (2007). Textual energy of associative memories: Performant applications of enertex algorithm in text summarization and topic segmentation. In *MICAI 2007: Advances in Artificial Intelligence*, pages 861–871. Springer.

[Gelbukh and Sidorov, 2006] Gelbukh, A. and Sidorov, G. (2006). *Procesamiento automático del español con enfoque en recursos léxicos grandes*. Centro de Investigación en Computación, Instituto Politécnico Nacional.

[Giannakopoulos et al., 2008] Giannakopoulos, G., Karkaletsis, V., and Vouros, G. (2008). Testing the use of n-gram graphs in summarization sub-tasks. In *Proceedings of the text analysis conference (TAC)*.

[González-Gallardo et al., 2015] González-Gallardo, C. E., Montes, A., Sierra, G., Nuñez-Juárez, J. A., Salinas-López, A. J., and Ek, J. (2015). Tweets classification using corpus dependent tags, character and POS n-grams. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Goswami et al., 2009] Goswami, S., Sarkar, S., and Rustagi, M. (2009). Stylometric analysis of bloggers' age and gender. In *Third International AAAI Conference on Weblogs and Social Media*.

[Grivas et al., 2015] Grivas, A., Krithara, A., and Giannakopoulos, G. (2015). Author profiling using stylometric and structural feature groupings. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Holmes and Meyerhoff, 2008] Holmes, J. and Meyerhoff, M. (2008). *The handbook of language and gender*, volume 25. John Wiley & Sons.

[Jones and Myhill, 2007] Jones, S. and Myhill, D. (2007). Discourses of difference? examining gender differences in linguistic characteristics of writing. *Canadian Journal of Education/Revue canadienne de l'éducation*, pages 456–482.

[Kanaris, 1999] Kanaris, A. (1999). Gendered journeys: Children's writing and the construction of gender. *Language and Education*, 13(4):254–268.

[Kocher and Savoy, 2015] Kocher, M. and Savoy, J. (2015). Unine at CLEF 2015 author identification: Notebook for PAN at CLEF 2015. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Koehn, 2010] Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

[Koppel et al., 2002] Koppel, M., Argamon, S., and Shimoni, A. R. (2002). Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412.

[Lopez-Monroy, 2013] Lopez-Monroy, A. (2013). Montes-y-gomez, m., escalante, hj, villasenor-pineda, l., villatoro-tello, e.: Inaoe's participation at pan'13: Author profiling task. *Notebook Papers of CLEF*.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

[Metsis et al., 2006] Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes? In *CEAS*, pages 27–28.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

[Mukherjee and Liu, 2010] Mukherjee, A. and Liu, B. (2010). Improving gender classification of blog authors. In *Proceedings of the 2010 conference on Empirical Methods in natural Language Processing*, pages 207–217. Association for Computational Linguistics.

[Newman et al., 2008] Newman, M. L., Groom, C. J., Handelman, L. D., and Pennebaker, J. W. (2008). Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*, 45(3):211–236.

[Nguyen et al., 2011] Nguyen, D., Smith, N. A., and Rosé, C. P. (2011). Author age prediction from text using linear regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 115–123. Association for Computational Linguistics.

[Nowson et al., 2015] Nowson, S., Perez, J., Brun, C., Mirkin, S., and Roux, C. (2015). XRCE personal language analytics engine for multilingual author profiling: Notebook for PAN at CLEF 2015. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Padró and Stanilovsky, 2012] Padró, L. and Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey. ELRA.

[PAN, 2015a] PAN (2015a). Pan author identification.

[PAN, 2015b] PAN (2015b). Pan author profiling.

[PAN, 2015c] PAN (2015c). Pan plagiarism detection.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

[Peersman et al., 2011] Peersman, C., Daelemans, W., and Van Vaerenbergh, L. (2011). Predicting age and gender in online social networks. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, pages 37–44. ACM.

[Pennebaker et al., 2003] Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.

[Pennebaker and Stone, 2003] Pennebaker, J. W. and Stone, L. D. (2003). Words of wisdom: language use over the life span. *Journal of personality and social psychology*, 85(2):291.

[Przybyla and Teisseyre, 2015] Przybyla, P. and Teisseyre, P. (2015). What do your look-alikes say about you? exploiting strong and weak similarities for author profiling. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Rangel et al., 2015] Rangel, F., Rosso, P., Potthast, M., Stein, B., and Daelemans, W. (2015). Overview of the 3rd author profiling task at pan 2015. In *CLEF*.

[Rao et al., 2010] Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM.

[Russell and Norvig, 2003] Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition.

[Saussure, 2008] Saussure, F. (2008). Curso de lingüística general.

[Schler et al., 2006] Schler, J., Koppel, M., Argamon, S., and Pennebaker, J. W. (2006). Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 6, pages 199–205.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

[Sra et al., 2012] Sra, S., Nowozin, S., and Wright, S. J. (2012). *Optimization for machine learning*. Mit Press.

[Stamatatos et al., 2015] Stamatatos, E., Potthast, M., Rangel, F., Rosso, P., and Stein, B. (2015). Overview of the pan/clef 2015 evaluation lab. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 518–538. Springer.

[Stein and Niggemann, 1999] Stein, B. and Niggemann, O. (1999). On the nature of structure and its identification. In *Graph-Theoretic Concepts in Computer Science*, pages 122–134. Springer.

[Sulea and Dichiu, 2015] Sulea, O. and Dichiu, D. (2015). Automatic profiling of twitter users based on their tweets: Notebook for PAN at CLEF 2015. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Talbot, 2010] Talbot, M. (2010). *Language and gender*. Polity.

[Twitter, 2015] Twitter (2015). Twitter newsroom.

[Werlen, 2015] Werlen, L. M. (2015). Statistical learning methods for profiling analysis: Notebook for PAN at CLEF 2015. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.*

[Wiemer-Hastings et al., 2004] Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (2004). Latent semantic analysis. In *Proceedings of the 16th international joint conference on Artificial intelligence*, pages 1–14. Citeseer.

[Zhang and Zhang, 2010] Zhang, C. and Zhang, P. (2010). Predicting gender from blog posts. Technical report.

Capítulo 8

Anexo

8.1. Creación de vectores

1. $\forall m \in M$:

   a) Extraer los n-gramas de caracteres como sus frecuencias y generar un vector de características $vCAR_m$

   $$vCAR_m = [(n-grama1, f_{n-grama1}), (n-grama2, f_{n-grama2}), \cdots, (n-gramaZ, f_{n-gramaZ})]$$

   b) Extraer los n-gramas POS como sus frecuencias y generar un vector de características $vPOS_m$

   $$vPOS_m = [(n-gramaPOS1, f_{n-gramaPOS1}), (n-gramaPOS2, f_{n-gramaPOS2}), \cdots, (n-gramaPOSZ, f_{n-gramaPOSZ})]$$

Tabla 8.1: Creación de vectores paso 1

$h1$	wooooooooooooooooooooooooooooooooooowwwwwwwwwwwww geniaaaaaaaaaaaaaal!!!
$vCAR_{h1}$	[('oo', 36), ('aa', 13), ('ww', 12), ('!!', 2), ('wg', 1), ('en', 1), ('ni', 1), ('wo', 1), ('l!', 1), ('al', 1), ('ia', 1), ('ge', 1), ('ow', 1)]
$vPOS_{h1}$	[('F F', 2), ('R A', 1), ('A F', 1)]
$h2$	SOLO SE QUE ES... GRADIOSO ESE LUGAR...
$vCAR_{h2}$	[('..', 4), ('ES', 2), ('SO', 2), ('OS', 2), ('SE', 2), ('GA', 1), ('DI', 1), ('S.', 1), ('AD', 1), ('LO', 1), ('LU', 1), ('RA', 1), ('EL', 1), ('GR', 1), ('EE', 1), ('.G', 1), ('AR', 1), ('IO', 1), ('R.', 1), ('EQ', 1), ('OL', 1), ('QU', 1), ('OE', 1), ('UE', 1), ('UG', 1)]
$vPOS_{h2}$	[('N F', 2), ('F N', 1)]
$h3$	Simplemente hermosaaaaaa <3
$vCAR_{h3}$	[('aa', 5), ('em', 1), ('me', 1), ('le', 1), ('eh', 1), ('¡3', 1), ('os', 1), ('mo', 1), ('Si', 1), ('en', 1), ('im', 1), ('mp', 1), ('er', 1), ('rm', 1), ('sa', 1), ('te', 1), ('nt', 1), ('a¡', 1), ('pl', 1), ('he', 1)]

$vPOS_{h3}$	[('V F', 1), ('F F', 1), ('R V', 1)]
$h4$	Aaaaaahhhh estoy a un ladooooooooooo. Esos Jardines del sur.
$vCAR_{h4}$	[('oo', 9), ('aa', 4), ('hh', 3), ('es', 2), ('el', 1), ('ad', 1), ('ah', 1), ('ar', 1), ('au', 1), ('in', 1), ('ur', 1), ('o.', 1), ('nl', 1), ('la', 1), ('ne', 1), ('rd', 1), ('to', 1), ('ls', 1), ('do', 1), ('di', 1), ('ya', 1), ('de', 1), ('.E', 1), ('sJ', 1), ('he', 1), ('Aa', 1), ('su', 1), ('st', 1), ('un', 1), ('so', 1), ('oy', 1), ('r.', 1), ('sd', 1), ('os', 1), ('Es', 1), ('Ja', 1)]
$vPOS_{h4}$	[('D N', 3), ('S D', 2), ('N F', 2), ('N V', 1), ('N S', 1), ('F D', 1), ('V S', 1)]
$h5$	De primerisimo mundo!!!!!!!!!
$vCAR_{h5}$	[('!!', 8), ('ri', 2), ('im', 2), ('pr', 1), ('me', 1), ('om', 1), ('mo', 1), ('do', 1), ('is', 1), ('De', 1), ('nd', 1), ('o!', 1), ('mu', 1), ('si', 1), ('un', 1), ('ep', 1), ('er', 1)]
$vPOS_{h5}$	[('F F', 8), ('S A', 1), ('N F', 1), ('A N', 1)]
$m6$	Wooooow es hermosaaa!!!! Gracias Cosmopitufo Espacial vecino!!
$vCAR_{m6}$	[('!!', 4), ('oo', 4), ('ci', 3), ('aa', 2), ('ac', 2), ('ia', 2), ('mo', 2), ('os', 2), ('ve', 1), ('al', 1), ('ec', 1), ('as', 1), ('in', 1), ('es', 1), ('er', 1), ('no', 1), ('Wo', 1), ('tu', 1), ('o!', 1), ('lv', 1), ('pa', 1), ('ra', 1), ('fo', 1), ('rm', 1), ('pi', 1), ('a!', 1), ('we', 1), ('oE', 1), ('it', 1), ('sC', 1), ('he', 1), ('Co', 1), ('Gr', 1), ('!G', 1), ('sp', 1), ('sh', 1), ('sm', 1), ('ow', 1), ('sa', 1), ('uf', 1), ('Es', 1), ('op', 1)]
$vPOS_{m6}$	[('F F', 4), ('N N', 1), ('N F', 1), ('V A', 1), ('N V', 1), ('A F', 1), ('F N', 1)]
$m7$	bonito lugar, y gracias x kompartir stas fotos. ;-) ;-)
$vCAR_{m7}$	[('as', 2), ('ar', 2), ('to', 2), ('-)', 2), (';-', 2), ('xk', 1), ('ac', 1), ('gr', 1), ('s.', 1), ('ir', 1), ('ci', 1), ('ga', 1), ('ia', 1), ('rt', 1), ('ni', 1), ('rs', 1), ('pa', 1), ('lu', 1), ('ra', 1), ('ti', 1), ('.;', 1), ('ta', 1), ('bo', 1), ('yg', 1), ('it', 1), ('sx', 1), ('fo', 1), ('on', 1), ('om', 1), ('ol', 1), (');', 1), ('ko', 1), ('st', 1), ('mp', 1), ('r,', 1), ('ug', 1), ('ot', 1), ('os', 1), ('sf', 1), (',y', 1)]
$vPOS_{m7}$	[('F F', 6), ('A N', 2), ('N F', 2), ('F C', 1), ('N V', 1), ('N N', 1), ('C N', 1), ('V A', 1)]
$m8$	hOy ese edificiO alberga Oficinas administrativas de la Secretaría de Salud, de ladO derechO esta el eificiO de la cOmisiÓn federal de luz, enfrente al mOnunemtO a cuauhtemOc , ahí ene ese edificiO trabaja mi mami :)

$vCAR_{m8}$	[('de', 7), ('el', 5), ('ci', 4), ('mi', 4), ('al', 4), ('ic', 4), ('fi', 4), ('la', 3), ('en', 3), ('ee', 3), ('ed', 3), ('es', 3), ('er', 3), ('re', 3), ('ra', 3), ('ad', 3), ('if', 3), ('iO', 3), ('nf', 2), ('tr', 2), ('lu', 2), ('te', 2), ('ta', 2), ('di', 2), ('Od', 2), ('Oa', 2), ('em', 2), ('ec', 2), ('mO', 2), ('hO', 2), ('ac', 2), ('is', 2), ('am', 2), ('as', 2), ('in', 2), ('ne', 2), ('st', 2), ('se', 2), ('d,', 1), (':)', 1), ('tO', 1), ('ga', 1), ('rí', 1), ('ld', 1), ('le', 1), ('lb', 1), ('lm', 1), ('ti', 1), ('On', 1), ('Om', 1), ('dm', 1), ('Of', 1), ('Oe', 1), ('ye', 1), ('Oy', 1), ('Ot', 1), ('dO', 1), ('ei', 1), ('ea', 1), ('i:', 1), ('Ón', 1), ('et', 1), ('íe', 1), ('ía', 1), ('rg', 1), ('eS', 1), ('uh', 1), ('z,', 1), ('be', 1), ('ba', 1), ('ja', 1), ('ch', 1), ('cr', 1), ('cu', 1), ('cO', 1), ('iÓ', 1), ('c,', 1), ('ht', 1), ('Sa', 1), ('Se', 1), ('ma', 1), ('uz', 1), ('mt', 1), ('un', 1), ('ud', 1), ('ua', 1), ('va', 1), ('ab', 1), ('ae', 1), ('ah', 1), ('aj', 1), ('iv', 1), ('ar', 1), ('au', 1), ('at', 1), ('ni', 1), ('na', 1), ('aO', 1), ('aS', 1), ('nt', 1), ('nu', 1), ('fr', 1), ('Oc', 1), ('fe', 1), (',d', 1), (',e', 1), ('im', 1), (',a', 1), ('si', 1), ('hí', 1), ('sa', 1), ('sd', 1)]
$vPOS_{m8}$	[('D N', 7), ('N F', 4), ('S D', 3), ('N S', 3), ('S N', 3), ('N V', 2), ('N A', 2), ('V D', 1), ('V N', 1), ('F V', 1), ('R D', 1), ('F S', 1), ('F R', 1), ('F F', 1), ('R N', 1), ('N N', 1), ('D D', 1), ('A D', 1), ('V S', 1), ('N D', 1), ('A S', 1)]
$m9$	Bonitos recuerdos de la voca 5 y la ciudadela :)
$vCAR_{m9}$	[('la', 3), ('el', 2), ('de', 2), ('os', 2), ('ac', 1), ('ad', 1), ('ca', 1), ('iu', 1), ('it', 1), ('ec', 1), ('vo', 1), ('av', 1), ('cu', 1), ('er', 1), ('ni', 1), ('5y', 1), ('ci', 1), ('Bo', 1), ('rd', 1), ('to', 1), ('do', 1), ('yl', 1), ('da', 1), ('a5', 1), ('re', 1), ('a:', 1), (':)', 1), ('on', 1), ('sr', 1), ('oc', 1), ('ue', 1), ('ud', 1), ('sd', 1)]
$vPOS_{m9}$	[('A N', 1), ('D N', 1), ('Z C', 1), ('S D', 1), ('N S', 1), ('D V', 1), ('F F', 1), ('N F', 1), ('C D', 1), ('V Z', 1)]
$m10$	durante mis estudios en la vocacional 5 Taxqueña (83-86) use la linea 1 y 2 del metro
$vCAR_{m10}$	[('se', 3), ('el', 2), ('io', 2), ('la', 2), ('al', 2), ('em', 1), ('ac', 1), ('en', 1), ('oc', 1), ('is', 1), ('in', 1), ('ea', 1), ('ci', 1), ('an', 1), ('et', 1), ('6)', 1), ('av', 1), ('ax', 1), ('st', 1), ('86', 1), ('es', 1), ('Ta', 1), ('xq', 1), ('nl', 1), ('1y', 1), ('na', 1), ('tu', 1), ('tr', 1), ('ne', 1), ('eñ', 1), ('li', 1), ('vo', 1), ('ra', 1), ('5T', 1), ('ña', 1), ('te', 1), ('nt', 1), ('ca', 1), ('y2', 1), ('(8', 1), ('di', 1), ('83', 1), ('a(', 1), ('de', 1), (')u', 1), ('a1', 1), ('du', 1), ('me', 1), ('on', 1), ('lm', 1), ('ro', 1), ('qu', 1), ('mi', 1), ('us', 1), ('ur', 1), ('2d', 1), ('l5', 1), ('3-', 1), ('ue', 1), ('ud', 1), ('os', 1), ('-8', 1)]
$vPOS_{m10}$	[('S D', 3), ('D N', 2), ('V D', 1), ('C Z', 1), ('F V', 1), ('D A', 1), ('Z N', 1), ('Z C', 1), ('N S', 1), ('F Z', 1), ('A Z', 1), ('D V', 1), ('Z F', 1), ('Z S', 1), ('N F', 1), ('V Z', 1)]

2. Generar un vector general de características $vCAR_{gral}$ asignando un cero a las frecuencias obtenidas

$$vCAR_{gral} = \bigcup_{m \in M} vCAR_m$$

Tabla 8.2: Creación de vectores paso 2

$vCAR_{gral}$	[('d,', 0), ('gr', 0), ('5T', 0), ('íe', 0), (');', 0), ('ía', 0), ('tO', 0), ('ge', 0), ('ch', 0), ('aO', 0), ('ga', 0), ('y2', 0), (':)', 0), ('ld', 0), ('le', 0), ('lb', 0), ('la', 0), ('Wo', 0), ('tu', 0), ('tr', 0), ('li', 0), ('lv', 0), ('to', 0), ('lu', 0), ('av', 0), ('RA', 0), ('ti', 0), ('te', 0), ('Ja', 0), ('ta', 0), ('do', 0), ('Om', 0), ('dm', 0), ('GR', 0), ('di', 0), ('ya', 0), ('Of', 0), ('Oe', 0), ('de', 0), ('ye', 0), ('yg', 0), ('da', 0), ('GA', 0), ('ma', 0), ('R.', 0), ('Ot', 0), ('dO', 0), ('yl', 0), ('OL', 0), ('qu', 0), ('Gr', 0), ('l!', 0), ('OE', 0), ('-)', 0), ('2d', 0), ('l5', 0), ('Bo', 0), ('uh', 0), ('OS', 0), ('-8', 0), ('em', 0), ('el', 0), ('en', 0), ('ei', 0), ('eh', 0), ('be', 0), ('ve', 0), ('ed', 0), ('fe', 0), ('ea', 0), ('ow', 0), ('ec', 0), ('i:', 0), ('et', 0), ('iu', 0), ('86', 0), ('ep', 0), ('es', 0), ('er', 0), ('rt', 0), ('o.', 0), ('rs', 0), ('o!', 0), ('rd', 0), ('wo', 0), ('rg', 0), ('On', 0), ('ra', 0), ('a(', 0), ('it', 0), ('rm', 0), ('hí', 0), ('ri', 0), ('eS', 0), ('hO', 0), ('wg', 0), ('EL', 0), ('we', 0), ('ba', 0), ('EE', 0), ('oE', 0), ('bo', 0), ('Oy', 0), ('ww', 0), ('ud', 0), ('re', 0), ('EQ', 0), ('ja', 0), ('ES', 0), ('oo', 0), ('on', 0), ('om', 0), ('ol', 0), ('rí', 0), ('oc', 0), ('oy', 0), ('UE', 0), ('r.', 0), ('ot', 0), ('os', 0), ('Es', 0), ('op', 0), ('!!', 0), ('xk', 0), ('ci', 0), ('lm', 0), ('5y', 0), ('6)', 0), ('ca', 0), ('S.', 0), ('cr', 0), ('xq', 0), ('cu', 0), ('ad', 0), ('pr', 0), ('cO', 0), ('..', 0), ('1y', 0), ('r,', 0), ('ee', 0), ('pa', 0), (';-', 0), ('eñ', 0), ('iÓ', 0), ('pi', 0), ('pl', 0), ('ls', 0), ('aj', 0), ('an', 0), ('c,', 0), ('(8', 0), ('UG', 0), ('mO', 0), ('z,', 0), (')u', 0), ('.E', 0), ('ht', 0), ('.G', 0), ('3-', 0), ('hh', 0), (',e', 0), ('Sa', 0), ('vo', 0), ('Se', 0), ('he', 0), ('me', 0), ('Co', 0), ('!G', 0), ('uz', 0), ('mo', 0), ('mi', 0), ('us', 0), ('ur', 0), ('mu', 0), ('mt', 0), ('un', 0), ('SO', 0), ('mp', 0), ('ue', 0), ('au', 0), ('ug', 0), ('uf', 0), ('ua', 0), ('SE', 0), ('aa', 0), ('va', 0), ('ac', 0), ('ab', 0), ('ae', 0), ('<3', 0), ('DI', 0), ('ah', 0), ('is', 0), ('ir', 0), ('am', 0), ('al', 0), ('iv', 0), ('as', 0), ('ar', 0), ('im', 0), ('at', 0), ('io', 0), ('in', 0), ('ia', 0), ('ax', 0), ('ic', 0), ('if', 0), ('ni', 0), ('Ón', 0), ('nl', 0), ('no', 0), ('LO', 0), ('De', 0), ('nd', 0), ('ne', 0), ('nf', 0), ('aS', 0), ('LU', 0), ('s.', 0), ('iO', 0), ('.;', 0), ('Ta', 0), ('nt', 0), ('nu', 0), ('a!', 0), ('fr', 0), ('Od', 0), ('AD', 0), ('83', 0), ('Oc', 0), ('ña', 0), ('a1', 0), ('sJ', 0), ('du', 0), ('AR', 0), ('a5', 0), ('Oa', 0), ('IO', 0), ('sC', 0), ('fi', 0), ('ro', 0), ('a:', 0), ('a<', 0), ('fo', 0), ('Aa', 0), (',d', 0), ('sx', 0), ('QU', 0), (',a', 0), ('sr', 0), ('sp', 0), ('ko', 0), ('Si', 0), ('su', 0), ('st', 0), ('si', 0), ('sh', 0), ('so', 0), ('sm', 0), ('na', 0), ('sa', 0), (',y', 0), ('sf', 0), ('se', 0), ('sd', 0)]

3. Generar un vector general de características $vPOS_{gral}$ asignando un cero a las frecuencias obtenidas

$$vPOS_{gral} = \bigcup_{m \in M} vPOS_m$$

Tabla 8.3: Creación de vectores paso 3

$vPOS_{gral}$	[('F V', 0), ('F S', 0), ('F R', 0), ('D N', 0), ('N V', 0), ('N S', 0), ('F Z', 0), ('F D', 0), ('N N', 0), ('F N', 0), ('F C', 0), ('N D', 0), ('N F', 0), ('N A', 0), ('V D', 0), ('V F', 0), ('V A', 0), ('D A', 0), ('V N', 0), ('D D', 0), ('V S', 0), ('V Z', 0), ('R A', 0), ('R D', 0), ('S N', 0), ('Z N', 0), ('S A', 0), ('Z C', 0), ('S D', 0), ('R N', 0), ('R V', 0), ('Z S', 0), ('Z F', 0), ('C Z', 0), ('A N', 0), ('D V', 0), ('A F', 0), ('A D', 0), ('A Z', 0), ('F F', 0), ('C N', 0), ('A S', 0), ('C D', 0)]

4. Generar el vector de características del sistema.

$$vSistema = vCAR_{gral} \cup vPOS_{gral}$$

Tabla 8.4: Creación de vectores paso 4

$vSistema$	[('d,', 0), ('gr', 0), ('5T', 0), ('íe', 0), ('N V', 0), (');', 0), ('ía', 0), ('tO', 0), ('ge', 0), ('ch', 0), ('aO', 0), ('ga', 0), ('N D', 0), ('y2', 0), ('N A', 0), (':)', 0), ('ld', 0), ('le', 0), ('lb', 0), ('la', 0), ('Wo', 0), ('tu', 0), ('tr', 0), ('li', 0), ('lv', 0), ('to', 0), ('lu', 0), ('av', 0), ('RA', 0), ('ti', 0), ('C N', 0), ('te', 0), ('F R', 0), ('Ja', 0), ('ta', 0), ('do', 0), ('Om', 0), ('dm', 0), ('GR', 0), ('di', 0), ('ya', 0), ('Of', 0), ('Oe', 0), ('de', 0), ('ye', 0), ('yg', 0), ('da', 0), ('N S', 0), ('GA', 0), ('ma', 0), ('R.', 0), ('Ot', 0), ('A F', 0), ('dO', 0), ('yl', 0), ('OL', 0), ('qu', 0), ('Gr', 0), ('l!', 0), ('OE', 0), ('-)', 0), ('2d', 0), ('l5', 0), ('A D', 0), ('Bo', 0), ('uh', 0), ('OS', 0), ('-8', 0), ('em', 0), ('el', 0), ('en', 0), ('ei', 0), ('eh', 0), ('F S', 0), ('be', 0), ('ve', 0), ('ed', 0), ('fe', 0), ('ea', 0), ('ow', 0), ('ec', 0), ('F Z', 0), ('F D', 0), ('i:', 0), ('F C', 0), ('et', 0), ('iu', 0), ('F N', 0), ('86', 0), ('ep', 0), ('es', 0), ('er', 0), ('rt', 0), ('o.', 0), ('N F', 0), ('rs', 0), ('o!', 0), ('rd', 0), ('wo', 0), ('rg', 0), ('On', 0), ('ra', 0), ('a(', 0), ('it', 0), ('rm', 0), ('hí', 0), ('ri', 0), ('eS', 0), ('hO', 0), ('wg', 0), ('EL', 0), ('we', 0), ('ba', 0), ('EE', 0), ('oE', 0), ('bo', 0), ('Oy', 0), ('R N', 0), ('ww', 0), ('ud', 0), ('re', 0), ('V A', 0), ('EQ', 0), ('ja', 0), ('ES', 0), ('oo', 0), ('on', 0), ('om', 0), ('ol', 0), ('rí', 0), ('oc', 0), ('D V', 0), ('oy', 0), ('UE', 0), ('r.', 0), ('ot', 0), ('os', 0), ('Es', 0), ('op', 0), ('!!', 0), ('xk', 0), ('ci', 0), ('lm', 0), ('5y', 0),

	('6)', 0), ('ca', 0), ('R V', 0), ('R A', 0), ('F V', 0), ('D D', 0), ('S.', 0), ('cr', 0), ('xq', 0), ('Z S', 0), ('cu', 0), ('ad', 0), ('pr', 0), ('V D', 0), ('R D', 0), ('V F', 0), ('cO', 0), ('D A', 0), ('..', 0), ('1y', 0), ('V N', 0), ('r,', 0), ('ee', 0), ('pa', 0), (';-', 0), ('eñ', 0), ('D N', 0), ('iÓ', 0), ('pi', 0), ('pl', 0), ('ls', 0), ('aj', 0), ('an', 0), ('V S', 0), ('c,', 0), ('(8', 0), ('UG', 0), ('mO', 0), ('z,', 0), (')u', 0), ('.E', 0), ('ht', 0), ('.G', 0), ('3-', 0), ('hh', 0),(',e', 0), ('A N', 0), ('Sa', 0), ('vo', 0), ('Se', 0), ('he', 0), ('me', 0), ('Co', 0), ('!G', 0), ('uz', 0), ('mo', 0), ('mi', 0), ('us', 0), ('ur', 0), ('mu', 0), ('mt', 0), ('un', 0), ('SO', 0), ('mp', 0), ('ue', 0), ('au', 0), ('ug', 0), ('uf', 0), ('ua', 0), ('SE', 0), ('aa', 0), ('va', 0), ('ac', 0), ('ab', 0), ('ae', 0), ('¡3', 0), ('DI', 0), ('F F', 0), ('ah', 0), ('is', 0), ('ir', 0), ('am', 0), ('al', 0), ('iv', 0), ('as', 0), ('ar', 0), ('im', 0), ('at', 0), ('io', 0), ('in', 0), ('ia', 0), ('ax', 0), ('ic', 0),('if', 0), ('ni', 0), ('Ón', 0), ('nl', 0), ('N N', 0), ('no', 0), ('LO', 0), ('De', 0), ('nd', 0), ('ne', 0), ('nf', 0), ('aS', 0), ('LU', 0), ('s.', 0), ('iO', 0), ('.;', 0), ('Ta', 0), ('nt', 0), ('nu', 0), ('a!', 0), ('fr', 0), ('Od', 0), ('AD', 0), ('S N', 0), ('Z N', 0), ('83', 0), ('Oc', 0), ('ña', 0), ('Z C', 0), ('S D', 0), ('S A', 0), ('Z F', 0), ('a1', 0), ('sJ', 0), ('du', 0), ('AR', 0), ('a5', 0),('Oa', 0), ('IO', 0), ('sC', 0), ('fi', 0), ('ro', 0), ('a:', 0), ('a¡', 0), ('fo', 0), ('Aa', 0), ('C Z', 0), (',d', 0), ('sx', 0), ('QU', 0), (',a', 0), ('sr', 0), ('sp', 0), ('ko', 0), ('Si', 0), ('su', 0), ('st', 0), ('A Z', 0), ('si', 0), ('sh', 0), ('so', 0), ('V Z', 0), ('sm', 0), ('na', 0), ('A S', 0), ('sa', 0), (',y', 0), ('C D', 0), ('sf', 0), ('se', 0), ('sd', 0)]

5. $\forall m \in M$:

   a) Generar un vector de características de la muestra.

   $$vMuestra_m = vSistema$$

   b) Asignar la frecuencia de los n-gramas de caracteres de $vCAR_m$ en $vMuestra_m$

   c) Asignar la frecuencia de los n-gramas POS de $vPOS_m$ en $vMuestra_m$

Tabla 8.5: Creación de vectores paso 5

$vMuestra_{h1}$	[('d,', 0), ('gr', 0), ('5T', 0), ('íe', 0), ('N V', 0), (');', 0), ('ía', 0), ('tO', 0), **('ge', 1)**, ('ch', 0), ('aO', 0), ('ga', 0), ('N D', 0), ('y2', 0), ('N A', 0), (':)', 0), ('ld', 0), ('le', 0), ('lb', 0), ('la', 0), ('Wo', 0), ('tu', 0), ('tr', 0), ('li', 0), ('lv', 0), ('to', 0), ('lu', 0), ('av', 0), ('RA', 0), ('ti', 0), ('C N', 0), ('te', 0), ('F R', 0), ('Ja', 0), ('ta', 0), ('do', 0), ('Om', 0), ('dm', 0), ('GR', 0), ('di', 0), ('ya', 0), ('Of', 0), ('Oe', 0), ('de', 0), ('ye', 0), ('yg', 0), ('da', 0), ('N S', 0), ('GA', 0), ('ma', 0), ('R.', 0), ('Ot', 0), **('A F', 1)**, ('dO', 0), ('yl', 0), ('OL', 0), ('qu', 0), ('Gr', 0), **('l!', 1)**, ('OE', 0),

	('-)', 0), ('2d', 0), ('l5', 0), ('A D', 0), ('Bo', 0), ('uh', 0), ('OS', 0), ('-8', 0), ('em', 0), ('el', 0), **('en', 1)**, ('ei', 0), ('eh', 0), ('F S', 0), ('be', 0), ('ve', 0), ('ed', 0), ('fe', 0), ('ea', 0), **('ow', 1)**, ('ec', 0), ('F Z', 0), ('F D', 0), ('i:', 0), ('F C', 0), ('et', 0), ('iu', 0), ('F N', 0), ('86', 0), ('ep', 0), ('es', 0), ('er', 0), ('rt', 0), ('o.', 0), ('N F', 0), ('rs', 0), ('o!', 0), ('rd', 0), **('wo', 1)**, ('rg', 0), ('On', 0), ('ra', 0), ('a(', 0), ('it', 0), ('rm', 0), ('hí', 0), ('ri', 0), ('eS', 0), ('hO', 0), **('wg', 1)**, ('EL', 0), ('we', 0), ('ba', 0),('EE', 0), ('oE', 0), ('bo', 0), ('Oy', 0), ('R N', 0), **('ww', 12**), ('ud', 0), ('re', 0), ('V A', 0), ('EQ', 0), ('ja', 0), ('ES', 0), **('oo', 36**), ('on', 0), ('om', 0), ('ol', 0), ('rí', 0), ('oc', 0), ('D V', 0), ('oy', 0), ('UE', 0), ('r.', 0), ('ot', 0), ('os', 0), ('Es', 0), ('op', 0), **('!!', 2)**, ('xk', 0), ('ci', 0), ('lm', 0), ('5y', 0), ('6)', 0), ('ca', 0), ('R V', 0), **('R A', 1)**, ('F V', 0), ('D D', 0), ('S.', 0), ('cr', 0), ('xq', 0), ('Z S', 0), ('cu', 0), ('ad', 0), ('pr', 0), ('V D', 0), ('R D', 0), ('V F', 0), ('cO', 0), ('D A', 0), ('..', 0), ('1y', 0), ('V N', 0), ('r,', 0), ('ee', 0), ('pa', 0), (';-', 0), ('eñ', 0), ('D N', 0), ('iÓ', 0), ('pi', 0), ('pl', 0), ('ls', 0), ('aj', 0), ('an', 0), ('V S', 0), ('c,', 0), ('(8', 0), ('UG', 0), ('mO', 0), ('z,', 0), (')u', 0), ('.E', 0), ('ht', 0), ('.G', 0), ('3-', 0), ('hh', 0), (',e', 0), ('A N', 0), ('Sa', 0), ('vo', 0), ('Se', 0), ('he', 0), ('me', 0), ('Co', 0), ('!G', 0), ('uz', 0), ('mo', 0), ('mi', 0), ('us', 0), ('ur', 0), ('mu', 0), ('mt', 0), ('un', 0), ('SO', 0), ('mp', 0), ('ue', 0), ('au', 0), ('ug', 0), ('uf', 0), ('ua', 0), ('SE', 0), **('aa', 13**), ('va', 0), ('ac', 0), ('ab', 0), ('ae', 0), ('¡3', 0), ('DI', 0), **('F F', 2)**, ('ah', 0), ('is', 0), ('ir', 0), ('am', 0), **('al', 1)**, ('iv', 0), ('as', 0), ('ar', 0), ('im', 0), ('at', 0), ('io', 0), ('in', 0), **('ia', 1)**, ('ax', 0), ('ic', 0), ('if', 0), **('ni', 1)**, ('Ón', 0), ('nl', 0), ('N N', 0), ('no', 0), ('LO', 0), ('De', 0), ('nd', 0), ('ne', 0), ('nf', 0), ('aS', 0), ('LU', 0), ('s.', 0), ('iO', 0), ('.;', 0), ('Ta', 0), ('nt', 0), ('nu', 0), ('a!', 0), ('fr', 0), ('Od', 0), ('AD', 0), ('S N', 0), ('Z N', 0), ('83', 0), ('Oc', 0), ('ña', 0), ('Z C', 0), ('S D', 0), ('S A', 0), ('Z F', 0), ('a1', 0), ('sJ', 0), ('du', 0), ('AR', 0), ('a5', 0), ('Oa', 0), ('IO', 0), ('sC', 0), ('fi', 0), ('ro', 0), ('a:', 0), ('a¡', 0), ('fo', 0), ('Aa', 0), ('C Z', 0), (',d', 0), ('sx', 0), ('QU', 0), (',a', 0), ('sr', 0), ('sp', 0), ('ko', 0), ('Si', 0), ('su', 0), ('st', 0), ('A Z', 0), ('si', 0), ('sh', 0), ('so', 0), ('V Z', 0), ('sm', 0), ('na', 0), ('A S', 0), ('sa', 0), (',y', 0), ('C D', 0), ('sf', 0), ('se', 0), ('sd', 0)]

$vMuestra_{h2}$	[('d,', 0),('gr', 0),('5T', 0),('íe', 0),('N V', 0),(');', 0),('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),('ga', 0),('N D', 0),('y2', 0),('N A', 0),(':)', 0),('ld', 0),('le', 0),('lb', 0),('la', 0),('Wo', 0),('tu', 0),('tr', 0),('li', 0),('lv', 0),('to', 0),('lu', 0),('av', 0),**('RA', 1)**,('ti', 0),('C N', 0),('te', 0),('F R', 0),('Ja', 0),('ta', 0),('do', 0),('Om', 0),('dm', 0),**('GR', 1)**,('di', 0),('ya', 0),('Of', 0),('Oe', 0),('de', 0),('ye', 0),('yg', 0),('da', 0),('N S', 0),**('GA', 1)**,('ma', 0),**('R.', 1)**,('Ot', 0),('A F', 0),('dO', 0),('yl', 0),**('OL', 1)**,('qu', 0),('Gr', 0),('l!', 0),**('OE', 1)**,('-)', 0),('2d', 0),('l5', 0),('A D', 0),('Bo', 0),('uh', 0),**('OS', 2)**,('-8', 0),('em', 0),('el', 0),('en', 0),('ei', 0),('eh', 0),('F S', 0),('be', 0),('ve', 0),('ed', 0),('fe', 0),('ea', 0),('ow', 0),('ec', 0),('F Z', 0),('F D', 0),('i:', 0),('F C', 0),('et', 0),('iu', 0),**('F N', 1)**,('86', 0),('ep', 0),('es', 0),('er', 0),('rt', 0),('o.', 0),**('N F', 2)**,('rs', 0),('o!', 0),('rd', 0),('wo', 0),('rg', 0),('On', 0),('ra', 0),('a(', 0),('it', 0),('rm', 0),('hí', 0),('ri', 0),('eS', 0),('hO', 0),('wg', 0),**('EL', 1)**,('we', 0),('ba', 0),**('EE', 1)**,('oE', 0),('bo', 0),('Oy', 0),('R N', 0),('ww', 0),('ud', 0),('re', 0),('V A', 0),**('EQ', 1)**,('ja', 0),**('ES', 2)**,('oo', 0),('on', 0),('om', 0),('ol', 0),('rí', 0),('oc', 0),('D V', 0),('oy', 0),**('UE', 1)**,('r.', 0),('ot', 0),('os', 0),('Es', 0),('op', 0),('!!', 0),('xk', 0),('ci', 0),('lm', 0),('5y', 0),('6)', 0),('ca', 0),('R V', 0),('R A', 0),('F V', 0),('D D', 0),**('S.', 1)**,('cr', 0),('xq', 0),('Z S', 0),('cu', 0),('ad', 0),('pr', 0),('V D', 0),('R D', 0),('V F', 0),('cO', 0),('D A', 0),**('..', 4)**,('1y', 0),('V N', 0),('r,', 0),('ee', 0),('pa', 0),(';-', 0),('eñ', 0),('D N', 0),('iÓ', 0),('pi', 0),('pl', 0),('ls', 0),('aj', 0),('an', 0),('V S', 0),('c,', 0),('(8', 0),**('UG', 1)**,('mO', 0),('z,', 0),(')u', 0),('.E', 0),('ht', 0),**('.G', 1)**,('3-', 0),('hh', 0),(',e', 0),('A N', 0),('Sa', 0),('vo', 0),('Se', 0),('he', 0),('me', 0),('Co', 0),('!G', 0),('uz', 0),('mo', 0),('mi', 0),('us', 0),('ur', 0),('mu', 0),('mt', 0),('un', 0),**('SO', 2)**,('mp', 0),('ue', 0),('au', 0),('ug', 0),('uf', 0),('ua', 0),**('SE', 2)**,('aa', 0),('va', 0),('ac', 0),('ab', 0),('ae', 0),('<3', 0),**('DI', 1)**,('F F', 0),('ah', 0),('is', 0),('ir', 0),('am', 0),('al', 0),('iv', 0),('as', 0),('ar', 0),('im', 0),('at', 0),('io', 0),('in', 0),('ia', 0),('ax', 0),('ic', 0),('if', 0),('ni', 0),('Ón', 0),('nl', 0),('N N', 0), ('no', 0),**('LO', 1)**,('De', 0),('nd', 0),('ne', 0),('nf', 0),('aS', 0),**('LU', 1)**,('s.', 0),('iO', 0),('.;', 0),('Ta', 0),('nt', 0),('nu', 0),('a!', 0),('fr', 0),('Od', 0),**('AD', 1)**,('S N', 0),('Z N', 0),('83', 0),('Oc', 0),('ña', 0),('Z C', 0),('S D', 0),('S A', 0),('Z F', 0),('a1', 0),('sJ', 0),('du', 0),**('AR', 1)**,('a5', 0),('Oa', 0),**('IO', 1)**,('sC', 0),('fi', 0),('ro', 0),('a:', 0),('a<', 0),('fo', 0),('Aa', 0),('C Z', 0),(',d', 0),('sx', 0),**('QU', 1)**,(',a', 0),('sr', 0),('sp', 0),('ko', 0),('Si', 0),('su', 0),('st', 0),('A Z', 0),('si', 0),('sh', 0),('so', 0),('V Z', 0),('sm', 0),('na', 0),('A S', 0),('sa', 0),(',y', 0),('C D', 0),('sf', 0),('se', 0),('sd', 0)]

$vMuestra_{h3}$	[('d,', 0),('gr', 0),('5T', 0),('íe', 0),('N V', 0),(');', 0),('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),('ga', 0),('N D', 0),('y2', 0),('N A', 0),(':)', 0),('ld', 0),**('le', 1)**,('lb', 0),('la', 0),('Wo', 0),('tu', 0),('tr', 0),('li', 0),('lv', 0),('to', 0),('lu', 0),('av', 0),('RA', 0),('ti', 0),('C N', 0),**('te', 1)**,('F R', 0),('Ja', 0),('ta', 0),('do', 0),('Om', 0),('dm', 0),('GR', 0),('di', 0),('ya', 0),('Of', 0),('Oe', 0),('de', 0),('ye', 0),('yg', 0),('da', 0),('N S', 0),('GA', 0),('ma', 0),('R.', 0),('Ot', 0),('A F', 0),('dO', 0),('yl', 0),('OL', 0),('qu', 0),('Gr', 0),('l!', 0),('OE', 0),('-)', 0),('2d', 0),('l5', 0),('A D', 0),('Bo', 0),('uh', 0),('OS', 0),('-8', 0),**('em', 1)**,('el', 0),**('en', 1)**,('ei', 0),**('eh', 1)**,('F S', 0),('be', 0),('ve', 0),('ed', 0),('fe', 0),('ea', 0),('ow', 0),('ec', 0),('F Z', 0),('F D', 0),('i:', 0),('F C', 0),('et', 0),('iu', 0),('F N', 0),('86', 0),('ep', 0),('es', 0),**('er', 1)**,('rt', 0),('o.', 0),('N F', 0),('rs', 0),('o!', 0),('rd', 0),('wo', 0),('rg', 0),('On', 0),('ra', 0),('a(', 0),('it', 0),**('rm', 1)**,('hí', 0),('ri', 0),('eS', 0),('hO', 0),('wg', 0),('EL', 0),('we', 0),('ba', 0),('EE', 0),('oE', 0),('bo', 0),('Oy', 0),('R N', 0),('ww', 0),('ud', 0),('re', 0),('V A', 0),('EQ', 0),('ja', 0),('ES', 0),('oo', 0),('on', 0),('om', 0),('ol', 0),('rí', 0),('oc', 0),('D V', 0),('oy', 0),('UE', 0),('r.', 0),('ot', 0),**('os', 1)**,('Es', 0),('op', 0),('!!', 0),('xk', 0),('ci', 0),('lm', 0),('5y', 0),('6)', 0),('ca', 0),**('R V', 1)**,('R A', 0),('F V', 0),('D D', 0),('S.', 0),('cr', 0),('xq', 0),('Z S', 0),('cu', 0),('ad', 0),('pr', 0),('V D', 0),('R D', 0),**('V F', 1)**,('cO', 0),('D A', 0),('..', 0),('1y', 0),('V N', 0),('r,', 0),('ee', 0),('pa', 0),(';-', 0),('eñ', 0),('D N', 0),('iÓ', 0),('pi', 0),**('pl', 1)**,('ls', 0),('aj', 0),('an', 0),('V S', 0),('c,', 0),('(8', 0),('UG', 0),('mO', 0),('z,', 0),(')u', 0),('.E', 0),('ht', 0),('.G', 0),('3-', 0),('hh', 0),(',e', 0),('A N', 0),('Sa', 0),('vo', 0),('Se', 0),**('he', 1)**,**('me', 1)**,('Co', 0),('!G', 0),('uz', 0),**('mo', 1)**,('mi', 0),('us', 0),('ur', 0),('mu', 0),('mt', 0),('un', 0),('SO', 0),**('mp', 1)**,('ue', 0),('au', 0),('ug', 0),('uf', 0),('ua', 0),('SE', 0),**('aa', 5)**,('va', 0),('ac', 0),('ab', 0),('ae', 0),**('<3', 1)**,('DI', 0),**('F F', 1)**,('ah', 0),('is', 0),('ir', 0),('am', 0),('al', 0),('iv', 0),('as', 0),('ar', 0),**('im', 1)**,('at', 0),('io', 0),('in', 0),('ia', 0),('ax', 0),('ic', 0),('if', 0),('ni', 0),('Ón', 0),('nl', 0),('N N', 0),('no', 0),('LO', 0),('De', 0),('nd', 0),('ne', 0),('nf', 0),('aS', 0),('LU', 0),('s.', 0),('iO', 0),('.;', 0),('Ta', 0),**('nt', 1)**,('nu', 0),('a!', 0),('fr', 0),('Od', 0),('AD', 0),('S N', 0),('Z N', 0),('83', 0),('Oc', 0),('ña', 0),('Z C', 0),('S D', 0),('S A', 0),('Z F', 0),('a1', 0),('sJ', 0),('du', 0),('AR', 0),('a5', 0),('Oa', 0),('IO', 0),('sC', 0),('fi', 0),('ro', 0),('a:', 0),**('a<', 1)**,('fo', 0),('Aa', 0),('C Z', 0),(',d', 0),('sx', 0),('QU', 0),(',a', 0),('sr', 0),('sp', 0),('ko', 0),**('Si', 1)**,('su', 0),('st', 0),('A Z', 0),('si', 0),('sh', 0),('so', 0),('V Z', 0),('sm', 0),('na', 0),('A S', 0),**('sa', 1)**,(',y', 0),('C D', 0),('sf', 0),('se', 0),('sd', 0)]

$vMuestra_{h4}$	[('d,', 0),('gr', 0),('5T', 0),('íe', 0),**('N V', 1)**,(');', 0),('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),('ga', 0),('N D', 0),('y2', 0),('N A', 0),(':)', 0),('ld', 0),('le', 0),('lb', 0),**('la', 1)**,('Wo', 0),('tu', 0),('tr', 0),('li', 0),('lv', 0),**('to', 1)**,('lu', 0),('av', 0),('RA', 0),('ti', 0),('C N', 0),('te', 0),('F R', 0),**('Ja', 1)**,('ta', 0),**('do', 1)**,('Om', 0),('dm', 0),('GR', 0),**('di', 1)**,**('ya', 1)**,('Of', 0),('Oe', 0),**('de', 1)**,('ye', 0),('yg', 0),('da', 0),**('N S', 1)**,('GA', 0),('ma', 0),('R.', 0),('Ot', 0),('A F', 0),('dO', 0),('yl', 0),('OL', 0),('qu', 0),('Gr', 0),('l!', 0),('OE', 0),('-)', 0),('2d', 0),('l5', 0),('A D', 0),('Bo', 0),('uh', 0),('OS', 0),('-8', 0),('em', 0),**('el', 1)**,('en', 0),('ei', 0),('eh', 0),('F S', 0),('be', 0),('ve', 0),('ed', 0),('fe', 0),('ea', 0),('ow', 0),('ec', 0),('F Z', 0),**('F D', 1)**,('i:', 0),('F C', 0),('et', 0),('iu', 0),('F N', 0),('86', 0),('ep', 0),**('es', 2)**,('er', 0),('rt', 0),**('o.', 1)**,**('N F', 2)**,('rs', 0),('o!', 0),**('rd', 1)**,('wo', 0),('rg', 0),('On', 0),('ra', 0),('a(', 0),('it', 0),('rm', 0),('hí', 0),('ri', 0),('eS', 0),('hO', 0),('wg', 0),('EL', 0),('we', 0),('ba', 0),('EE', 0),('oE', 0),('bo', 0),('Oy', 0),('R N', 0),('ww', 0),('ud', 0),('re', 0),('V A', 0),('EQ', 0),('ja', 0),('ES', 0),**('oo', 9)**,('on', 0),('om', 0),('ol', 0),('rí', 0),('oc', 0),('D V', 0),**('oy', 1)**,('UE', 0),**('r.', 1)**,('ot', 0),**('os', 1)**,**('Es', 1)**,('op', 0),('!!', 0),('xk', 0),('ci', 0),('lm', 0),('5y', 0),('6)', 0),('ca', 0),('R V', 0),('R A', 0),('F V', 0),('D D', 0),('S.', 0),('cr', 0),('xq', 0),('Z S', 0),('cu', 0),**('ad', 1)**,('pr', 0),('V D', 0),('R D', 0),('V F', 0),('cO', 0),('D A', 0),('..', 0),('1y', 0),('V N', 0),('r,', 0),('ee', 0),('pa', 0),(';-', 0),('eñ', 0),**('D N', 3)**,('iÓ', 0),('pi', 0),('pl', 0),**('ls', 1)**,('aj', 0),('an', 0),**('V S', 1)**,('c,', 0),('(8', 0),('UG', 0),('mO', 0),('z,', 0),(')u', 0),**('.E', 1)**,('ht', 0),('.G', 0),('3-', 0),**('hh', 3)**,(',e', 0),('A N', 0),('Sa', 0),('vo', 0),('Se', 0),**('he', 1)**,('me', 0),('Co', 0),('!G', 0),('uz', 0),('mo', 0),('mi', 0),('us', 0),**('ur', 1)**,('mu', 0),('mt', 0),**('un', 1)**,('SO', 0),('mp', 0),('ue', 0),**('au', 1)**,('ug', 0),('uf', 0),('ua', 0),('SE', 0),**('aa', 4)**,('va', 0),('ac', 0),('ab', 0),('ae', 0),('<3', 0),('DI', 0),('F F', 0),**('ah', 1)**,('is', 0),('ir', 0),('am', 0),('al', 0),('iv', 0),('as', 0),**('ar', 1)**,('im', 0),('at', 0),('io', 0),**('in', 1)**,('ia', 0),('ax', 0),('ic', 0),('if', 0),('ni', 0),('Ón', 0),**('nl', 1)**,('N N', 0),('no', 0),('LO', 0),('De', 0),('nd', 0),**('ne', 1)**,('nf', 0),('aS', 0),('LU', 0),('s.', 0),('iO', 0),('.;', 0),('Ta', 0),('nt', 0),('nu', 0),('a!', 0),('fr', 0),('Od', 0),('AD', 0),('S N', 0),('Z N', 0),('83', 0),('Oc', 0),('ña', 0),('Z C', 0),**('S D', 2)**,('S A', 0),('Z F', 0),('a1', 0),**('sJ', 1)**,('du', 0),('AR', 0),('a5', 0),('Oa', 0),('IO', 0),('sC', 0),('fi', 0),('ro', 0),('a:', 0),('a<', 0),('fo', 0),**('Aa', 1)**,('C Z', 0),(',d', 0),('sx', 0),('QU', 0),(',a', 0),('sr', 0),('sp', 0),('ko', 0),('Si', 0),**('su', 1)**,**('st', 1)**,('A Z', 0),('si', 0),('sh', 0),**('so', 1)**,('V Z', 0),('sm', 0),('na', 0),('A S', 0),('sa', 0),(',y', 0),('C D', 0),('sf', 0),('se', 0),**('sd', 1)**]

$vMuestra_{h5}$	[('d,', 0),('gr', 0),('5T', 0),('íe', 0),('N V', 0),(');', 0),('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),('ga', 0),('N D', 0),('y2', 0),('N A', 0),(':)', 0),('ld', 0),('le', 0),('lb', 0),('la', 0),('Wo', 0),('tu', 0),('tr', 0),('li', 0),('lv', 0),('to', 0),('lu', 0),('av', 0),('RA', 0),('ti', 0),('C N', 0),('te', 0),('F R', 0),('Ja', 0),('ta', 0),**('do', 1)**,('Om', 0),('dm', 0),('GR', 0),('di', 0),('ya', 0),('Of', 0),('Oe', 0),('de', 0),('ye', 0),('yg', 0),('da', 0),('N S', 0),('GA', 0),('ma', 0),('R.', 0),('Ot', 0),('A F', 0),('dO', 0),('yl', 0),('OL', 0),('qu', 0),('Gr', 0),('l!', 0),('OE', 0),('-)', 0),('2d', 0),('l5', 0),('A D', 0),('Bo', 0),('uh', 0),('OS', 0),('-8', 0),('em', 0),('el', 0),('en', 0),('ei', 0),('eh', 0),('F S', 0),('be', 0),('ve', 0),('ed', 0),('fe', 0),('ea', 0),('ow', 0),('ec', 0),('F Z', 0),('F D', 0),('i:', 0),('F C', 0),('et', 0),('iu', 0),('F N', 0),('86', 0),**('ep', 1)**,('es', 0),**('er', 1)**,('rt', 0),('o.', 0),**('N F', 1)**,('rs', 0),**('o!', 1)**,('rd', 0),('wo', 0),('rg', 0),('On', 0),('ra', 0),('a(', 0),('it', 0),('rm', 0),('hí', 0),**('ri', 2)**,('eS', 0),('hO', 0),('wg', 0),('EL', 0),('we', 0),('ba', 0),('EE', 0),('oE', 0),('bo', 0),('Oy', 0),('R N', 0),('ww', 0),('ud', 0),('re', 0),('V A', 0),('EQ', 0),('ja', 0),('ES', 0),('oo', 0),('on', 0),**('om', 1)**,('ol', 0),('rí', 0),('oc', 0),('D V', 0),('oy', 0),('UE', 0),('r.', 0),('ot', 0),('os', 0),('Es', 0),('op', 0),**('!!', 8)**,('xk', 0),('ci', 0),('lm', 0),('5y', 0),('6)', 0),('ca', 0),('R V', 0),('R A', 0),('F V', 0),('D D', 0),('S.', 0),('cr', 0),('xq', 0),('Z S', 0),('cu', 0),('ad', 0),**('pr', 1)**,('V D', 0),('R D', 0),('V F', 0),('cO', 0),('D A', 0),('..', 0),('1y', 0),('V N', 0),('r,', 0),('ee', 0),('pa', 0),(';-', 0),('eñ', 0),('D N', 0),('iÓ', 0),('pi', 0),('pl', 0),('ls', 0),('aj', 0),('an', 0),('V S', 0),('c,', 0),('(8', 0),('UG', 0),('mO', 0),('z,', 0),(')u', 0),('.E', 0),('ht', 0),('.G', 0),('3-', 0),('hh', 0),(',e', 0),**('A N', 1)**,('Sa', 0),('vo', 0),('Se', 0),('he', 0),**('me', 1)**,('Co', 0),('!G', 0),('uz', 0),**('mo', 1)**,('mi', 0),('us', 0),('ur', 0),**('mu', 1)**,('mt', 0),**('un', 1)**,('SO', 0),('mp', 0),('ue', 0),('au', 0),('ug', 0),('uf', 0),('ua', 0),('SE', 0),('aa', 0),('va', 0),('ac', 0),('ab', 0),('ae', 0),('<3', 0),('DI', 0),**('F F', 8)**,('ah', 0),**('is', 1)**,('ir', 0),('am', 0),('al', 0),('iv', 0),('as', 0),('ar', 0),**('im', 2)**,('at', 0),('io', 0),('in', 0),('ia', 0),('ax', 0),('ic', 0),('if', 0),('ni', 0),('Ón', 0),('nl', 0),('N N', 0),('no', 0),('LO', 0),**('De', 1)**,**('nd', 1)**,('ne', 0),('nf', 0),('aS', 0),('LU', 0),('s.', 0),('iO', 0),('.;', 0),('Ta', 0),('nt', 0),('nu', 0),('a!', 0),('fr', 0),('Od', 0),('AD', 0),('S N', 0),('Z N', 0),('83', 0),('Oc', 0),('ña', 0),('Z C', 0),('S D', 0),**('S A', 1)**,('Z F', 0),('a1', 0),('sJ', 0),('du', 0),('AR', 0),('a5', 0),('Oa', 0),('IO', 0),('sC', 0),('fi', 0),('ro', 0),('a:', 0),('a<', 0),('fo', 0),('Aa', 0),('C Z', 0),(',d', 0),('sx', 0),('QU', 0),(',a', 0),('sr', 0),('sp', 0),('ko', 0),('Si', 0),('su', 0),('st', 0),('A Z', 0),**('si', 1)**,('sh', 0),('so', 0),('V Z', 0),('sm', 0),('na', 0),('A S', 0),('sa', 0),(',y', 0),('C D', 0),('sf', 0),('se', 0),('sd', 0)]

$vMuestra_{m6}$	[('d,', 0),('gr', 0),('5T', 0),('íe', 0),**('N V', 1)**,(');', 0),('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),('ga', 0),('N D', 0),('y2', 0),('N A', 0),(':)', 0),('ld', 0),('le', 0),('lb', 0),('la', 0),**('Wo', 1)**,**('tu', 1)**,('tr', 0),('li', 0),**('lv', 1)**,('to', 0),('lu', 0),('av', 0),('RA', 0),('ti', 0),('C N', 0),('te', 0),('F R', 0),('Ja', 0),('ta', 0),('do', 0),('Om', 0),('dm', 0),('GR', 0),('di', 0),('ya', 0),('Of', 0),('Oe', 0),('de', 0),('ye', 0),('yg', 0),('da', 0),('N S', 0),('GA', 0),('ma', 0),('R.', 0),('Ot', 0),**('A F', 1)**,('dO', 0),('yl', 0),('OL', 0),('qu', 0),**('Gr', 1)**,('l!', 0),('OE', 0),('-)', 0),('2d', 0),('l5', 0),('A D', 0),('Bo', 0),('uh', 0),('OS', 0),('-8', 0),('em', 0),('el', 0),('en', 0),('ei', 0),('eh', 0),('F S', 0),('be', 0),**('ve', 1)**,('ed', 0),('fe', 0),('ea', 0),**('ow', 1)**,**('ec', 1)**,('F Z', 0),('F D', 0),('i:', 0),('F C', 0),('et', 0),('iu', 0),**('F N', 1)**,('86', 0),('ep', 0),**('es', 1)**,**('er', 1)**,('rt', 0),('o.', 0),**('N F', 1)**,('rs', 0),**('o!', 1)**,('rd', 0),('wo', 0),('rg', 0),('On', 0),**('ra', 1)**,('a(', 0),**('it', 1)**,**('rm', 1)**,('hí', 0),('ri', 0),('eS', 0),('hO', 0),('wg', 0),('EL', 0),**('we', 1)**,('ba', 0),('EE', 0),**('oE', 1)**,('bo', 0),('Oy', 0),('R N', 0),('ww', 0),('ud', 0),('re', 0),**('V A', 1)**,('EQ', 0),('ja', 0),('ES', 0),**('oo', 4)**,('on', 0),('om', 0),('ol', 0),('rí', 0),('oc', 0),('D V', 0),('oy', 0),('UE', 0),('r.', 0),('ot', 0),**('os', 2)**,**('Es', 1)**,**('op', 1)**,**('!!', 4)**,('xk', 0),**('ci', 3)**,('lm', 0),('5y', 0),('6)', 0),('ca', 0),('R V', 0),('R A', 0),('F V', 0),('D D', 0),('S.', 0),('cr', 0),('xq', 0),('Z S', 0),('cu', 0),('ad', 0),('pr', 0),('V D', 0),('R D', 0),('V F', 0),('cO', 0),('D A', 0),('..', 0),('1y', 0),('V N', 0),('r,', 0),('ee', 0),**('pa', 1)**,(';-', 0),('eñ', 0),('D N', 0),('iÓ', 0),**('pi', 1)**,('pl', 0),('ls', 0),('aj', 0),('an', 0),('V S', 0),('c,', 0),('(8', 0),('UG', 0),('mO', 0),('z,', 0),(')u', 0),('.E', 0),('ht', 0),('.G', 0),('3-', 0),('hh', 0),(',e', 0),('A N', 0),('Sa', 0),('vo', 0),('Se', 0),**('he', 1)**,('me', 0),**('Co', 1)**,**('!G', 1)**,('uz', 0),**('mo', 2)**,('mi', 0),('us', 0),('ur', 0),('mu', 0),('mt', 0),('un', 0),('SO', 0),('mp', 0),('ue', 0),('au', 0),('ug', 0),**('uf', 1)**,('ua', 0),('SE', 0),**('aa', 2)**,('va', 0),**('ac', 2)**,('ab', 0),('ae', 0),('<3', 0),('DI', 0),**('F F', 4)**,('ah', 0),('is', 0),('ir', 0),('am', 0),**('al', 1)**,('iv', 0),**('as', 1)**,('ar', 0),('im', 0),('at', 0),('io', 0),**('in', 1)**,**('ia', 2)**,('ax', 0),('ic', 0),('if', 0),('ni', 0),('Ón', 0),('nl', 0),**('N N', 1)**,**('no', 1)**,('LO', 0),('De', 0),('nd', 0),('ne', 0),('nf', 0),('aS', 0),('LU', 0),('s.', 0),('iO', 0),('.;', 0),('Ta', 0),('nt', 0),('nu', 0),**('a!', 1)**,('fr', 0),('Od', 0),('AD', 0),('S N', 0),('Z N', 0),('83', 0),('Oc', 0),('ña', 0),('Z C', 0),('S D', 0),('S A', 0),('Z F', 0),('a1', 0),('sJ', 0),('du', 0),('AR', 0),('a5', 0),('Oa', 0),('IO', 0),**('sC', 1)**,('fi', 0),('ro', 0),('a:', 0),('a<', 0),**('fo', 1)**,('Aa', 0),('C Z', 0),(',d', 0),('sx', 0),('QU', 0),(',a', 0),('sr', 0),**('sp', 1)**,('ko', 0),('Si', 0),('su', 0),('st', 0),('A Z', 0),('si', 0),**('sh', 1)**,('so', 0),('V Z', 0),**('sm', 1)**,('na', 0),('A S', 0),**('sa', 1)**,(',y', 0),('C D', 0),('sf', 0),('se', 0),('sd', 0)]

$vMuestra_{m7}$	[('d,', 0),**('gr', 1)**,('5T', 0),('íe', 0),**('N V', 1)**,**(');', 1)**,('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),**('ga', 1)**,('N D', 0),('y2', 0),('N A', 0),(':)', 0),('ld', 0),('le', 0),('lb', 0),('la', 0),('Wo', 0),('tu', 0),('tr', 0),('li', 0),('lv', 0),**('to', 2)**,**('lu', 1)**,('av', 0),('RA', 0),**('ti', 1)**,**('C N', 1)**,('te', 0),('F R', 0),('Ja', 0),**('ta', 1)**,('do', 0),('Om', 0),('dm', 0),('GR', 0),('di', 0),('ya', 0),('Of', 0),('Oe', 0),('de', 0),('ye', 0),**('yg', 1)**,('da', 0),('N S', 0),('GA', 0),('ma', 0),('R.', 0),('Ot', 0),('A F', 0),('dO', 0),('yl', 0),('OL', 0),('qu', 0),('Gr', 0),('l!', 0),('OE', 0),**('-)', 2)**,('2d', 0),('l5', 0),('A D', 0),('Bo', 0),('uh', 0),('OS', 0),('-8', 0),('em', 0),('el', 0),('en', 0),('ei', 0),('eh', 0),('F S', 0),('be', 0),('ve', 0),('ed', 0),('fe', 0),('ea', 0),('ow', 0),('ec', 0),('F Z', 0),('F D', 0),('i:', 0),**('F C', 1)**,('et', 0),('iu', 0),('F N', 0),('86', 0),('ep', 0),('es', 0),('er', 0),**('rt', 1)**,('o.', 0),**('N F', 2)**,**('rs', 1)**,('o!', 0),('rd', 0),('wo', 0),('rg', 0),('On', 0),**('ra', 1)**,('a(', 0),**('it', 1)**,('rm', 0),('hí', 0),('ri', 0),('eS', 0),('hO', 0),('wg', 0),('EL', 0),('we', 0),('ba', 0),('EE', 0),('oE', 0),**('bo', 1)**,('Oy', 0),('R N', 0),('ww', 0),('ud', 0),('re', 0),**('V A', 1)**,('EQ', 0),('ja', 0),('ES', 0),('oo', 0),**('on', 1)**,**('om', 1)**,**('ol', 1)**,('rí', 0),('oc', 0),('D V', 0),('oy', 0),('UE', 0),('r.', 0),**('ot', 1)**,**('os', 1)**,('Es', 0),('op', 0),('!!', 0),**('xk', 1)**,**('ci', 1)**,('lm', 0),('5y', 0),('6)', 0),('ca', 0),('R V', 0),('R A', 0),('F V', 0),('D D', 0),('S.', 0),('cr', 0),('xq', 0),('Z S', 0),('cu', 0),('ad', 0),('pr', 0),('V D', 0),('R D', 0),('V F', 0),('cO', 0),('D A', 0),('..', 0),('1y', 0),('V N', 0),**('r,', 1)**,('ee', 0),**('pa', 1)**,**(';-', 2)**,('eñ', 0),('D N', 0),('iÓ', 0),('pi', 0),('pl', 0),('ls', 0),('aj', 0),('an', 0),('V S', 0),('c,', 0),('(8', 0),('UG', 0),('mO', 0),('z,', 0),(')u', 0),('.E', 0),('ht', 0),('.G', 0),('3-', 0),('hh', 0),(',e', 0),**('A N', 2)**,('Sa', 0),('vo', 0),('Se', 0),('he', 0),('me', 0),('Co', 0),('!G', 0),('uz', 0),('mo', 0),('mi', 0),('us', 0),('ur', 0),('mu', 0),('mt', 0),('un', 0),('SO', 0),**('mp', 1)**,('ue', 0),('au', 0),**('ug', 1)**,('uf', 0),('ua', 0),('SE', 0),('aa', 0),('va', 0),**('ac', 1)**,('ab', 0),('ae', 0),('<3', 0),('DI', 0),**('F F', 6)**,('ah', 0),('is', 0),**('ir', 1)**,('am', 0),('al', 0),('iv', 0),**('as', 2)**,**('ar', 2)**,('im', 0),('at', 0),('io', 0),('in', 0),**('ia', 1)**,('ax', 0),('ic', 0),('if', 0),**('ni', 1)**,('Ón', 0),('nl', 0),**('N N', 1)**,('no', 0),('LO', 0),('De', 0),('nd', 0),('ne', 0),('nf', 0),('aS', 0),('LU', 0),**('s.', 1)**,('iO', 0),**('.;', 1)**,('Ta', 0),('nt', 0),('nu', 0),('a!', 0),('fr', 0),('Od', 0),('AD', 0),('S N', 0),('Z N', 0),('83', 0),('Oc', 0),('ña', 0),('Z C', 0),('S D', 0),('S A', 0),('Z F', 0),('a1', 0),('sJ', 0),('du', 0),('AR', 0),('a5', 0),('Oa', 0),('IO', 0),('sC', 0),('fi', 0),('ro', 0),('a:', 0),('a<', 0),**('fo', 1)**,('Aa', 0),('C Z', 0),(',d', 0),**('sx', 1)**,('QU', 0),(',a', 0),('sr', 0),('sp', 0),**('ko', 1)**,('Si', 0),('su', 0),**('st', 1)**,('A Z', 0),('si', 0),('sh', 0),('so', 0),('V Z', 0),('sm', 0),('na', 0),('A S', 0),('sa', 0),**(',y', 1)**,('C D', 0),**('sf', 1)**,('se', 0),('sd', 0)]

$vMuestra_{m8}$	[**('d,', 1)**,('gr', 0),('5T', 0),**('íe', 1)**,**('N V', 2)**,(');', 0),**('ía', 1)**,**('tO', 1)**,('ge', 0),**('ch', 1)**,**('aO', 1)**,**('ga', 1)**,**('N D', 1)**,('y2', 0),**('N A', 2)**,**(':)', 1)**,**('ld', 1)**,**('le', 1)**,**('lb', 1)**,**('la', 3)**,('Wo', 0),('tu', 0),**('tr', 2)**,('li', 0),('lv', 0),('to', 0),**('lu', 2)**,('av', 0),('RA', 0),**('ti', 1)**,('C N', 0),**('te', 2)**,**('F R', 1)**,('Ja', 0),**('ta', 2)**,('do', 0),**('Om', 1)**,**('dm', 1)**,('GR', 0),**('di', 2)**,('ya', 0),**('Of', 1)**,**('Oe', 1)**,**('de', 7)**,**('ye', 1)**,('yg', 0),('da', 0),**('N S', 3)**,('GA', 0),**('ma', 1)**,('R.', 0),**('Ot', 1)**,('A F', 0),**('dO', 1)**,('yl', 0),('OL', 0),('qu', 0),('Gr', 0),('l!', 0),('OE', 0),('-)', 0),('2d', 0),('l5', 0),**('A D', 1)**,('Bo', 0),**('uh', 1)**,('OS', 0),('-8', 0),**('em', 2)**,**('el', 5)**,**('en', 3)**,**('ei', 1)**,('eh', 0),**('F S', 1)**,**('be', 1)**,('ve', 0),**('ed', 3)**,**('fe', 1)**,**('ea', 1)**,('ow', 0),**('ec', 2)**,('F Z', 0),('F D', 0),**('i:', 1)**,('F C', 0),**('et', 1)**,('iu', 0),('F N', 0),('86', 0),('ep', 0),**('es', 3)**,**('er', 3)**,('rt', 0),('o.', 0),**('N F', 4)**,('rs', 0),('o!', 0),('rd', 0),('wo', 0),**('rg', 1)**,**('On', 1)**,**('ra', 3)**,('a(', 0),('it', 0),('rm', 0),**('hí', 1)**,('ri', 0),**('eS', 1)**,**('hO', 2)**,('wg', 0),('EL', 0),('we', 0),**('ba', 1)**,('EE', 0),('oE', 0),('bo', 0),**('Oy', 1)**,**('R N', 1)**,('ww', 0),**('ud', 1)**,**('re', 3)**,('V A', 0),('EQ', 0),**('ja', 1)**,('ES', 0),('oo', 0),('on', 0),('om', 0),('ol', 0),**('rí', 1)**,('oc', 0),('D V', 0),('oy', 0),('UE', 0),('r.', 0),('ot', 0),('os', 0),('Es', 0),('op', 0),('!!', 0),('xk', 0),**('ci', 4)**,**('lm', 1)**,('5y', 0),('6)', 0),('ca', 0),('R V', 0),('R A', 0),**('F V', 1)**,**('D D', 1)**,('S.', 0),**('cr', 1)**,('xq', 0),('Z S', 0),**('cu', 1)**,**('ad', 3)**,('pr', 0),**('V D', 1)**,**('R D', 1)**,('V F', 0),**('cO', 1)**,('D A', 0),('..', 0),('1y', 0),**('V N', 1)**,('r,', 0),**('ee', 3)**,('pa', 0),(';-', 0),('eñ', 0),**('D N', 7)**,**('iÓ', 1)**,('pi', 0),('pl', 0),('ls', 0),**('aj', 1)**,('an', 0),**('V S', 1)**,**('c,', 1)**,('(8', 0),('UG', 0),**('mO', 2)**,**('z,', 1)**,(')u', 0),('.E', 0),**('ht', 1)**,('.G', 0),('3-', 0),('hh', 0),**(',e', 1)**,('A N', 0),**('Sa', 1)**,('vo', 0),**('Se', 1)**,('he', 0),('me', 0),('Co', 0),('!G', 0),**('uz', 1)**,('mo', 0),**('mi', 4)**,('us', 0),('ur', 0),('mu', 0),**('mt', 1)**,**('un', 1)**,('SO', 0),('mp', 0),('ue', 0),**('au', 1)**,('ug', 0),('uf', 0),**('ua', 1)**,('SE', 0),('aa', 0),**('va', 1)**,**('ac', 2)**,**('ab', 1)**,**('ae', 1)**,('<3', 0),('DI', 0),**('F F', 1)**,**('ah', 1)**,**('is', 2)**,('ir', 0),**('am', 2)**,**('al', 4)**,**('iv', 1)**,**('as', 2)**,**('ar', 1)**,**('im', 1)**,**('at', 1)**,('io', 0),**('in', 2)**,('ia', 0),('ax', 0),**('ic', 4)**,**('if', 3)**,**('ni', 1)**,**('Ón', 1)**,('nl', 0),**('N N', 1)**,('no', 0),('LO', 0),('De', 0),('nd', 0),**('ne', 2)**,**('nf', 2)**,**('aS', 1)**,('LU', 0),('s.', 0),**('iO', 3)**,('.;', 0),('Ta', 0),**('nt', 1)**,**('nu', 1)**,('a!', 0),**('fr', 1)**,**('Od', 2)**,('AD', 0),**('S N', 3)**,('Z N', 0),('83', 0),**('Oc', 1)**,('ña', 0),('Z C', 0),**('S D', 3)**,('S A', 0),('Z F', 0),('a1', 0),('sJ', 0),('du', 0),('AR', 0),('a5', 0),**('Oa', 2)**,('IO', 0),('sC', 0),**('fi', 4)**,('ro', 0),('a:', 0),('a<', 0),('fo', 0),('Aa', 0),('C Z', 0),**(',d', 1)**,('sx', 0),('QU', 0),**(',a', 1)**,('sr', 0),('sp', 0),('ko', 0),('Si', 0),('su', 0),**('st', 2)**,('A Z', 0),**('si', 1)**,('sh', 0),('so', 0),('V Z', 0),('sm', 0),**('na', 1)**,**('A S', 1)**,**('sa', 1)**,(',y', 0),('C D', 0),('sf', 0),**('se', 2)**,**('sd', 1)**]

$vMuestra_{m9}$	[('d,', 0),('gr', 0),('5T', 0),('íe', 0),('N V', 0),(');', 0),('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),('ga', 0),('N D', 0),('y2', 0),('N A', 0),**(':)', 1)**,('ld', 0),('le', 0),('lb', 0),**('la', 3)**,('Wo', 0),('tu', 0),('tr', 0),('li', 0),('lv', 0),**('to', 1)**,('lu', 0),**('av', 1)**,('RA', 0),('ti', 0),('C N', 0),('te', 0),('F R', 0),('Ja', 0),('ta', 0),**('do', 1)**,('Om', 0),('dm', 0),('GR', 0),('di', 0),('ya', 0),('Of', 0),('Oe', 0),**('de', 2)**,('ye', 0),('yg', 0),**('da', 1)**,**('N S', 1)**,('GA', 0),('ma', 0),('R.', 0),('Ot', 0),('A F', 0),('dO', 0),**('yl', 1)**,('OL', 0),('qu', 0),('Gr', 0),('l!', 0),('OE', 0),('-)', 0),('2d', 0),('l5', 0),('A D', 0),**('Bo', 1)**,('uh', 0),('OS', 0),('-8', 0),('em', 0),**('el', 2)**,('en', 0),('ei', 0),('eh', 0),('F S', 0),('be', 0),('ve', 0),('ed', 0),('fe', 0),('ea', 0),('ow', 0),**('ec', 1)**,('F Z', 0),('F D', 0),('i:', 0),('F C', 0),('et', 0),**('iu', 1)**,('F N', 0),('86', 0),('ep', 0),('es', 0),**('er', 1)**,('rt', 0),('o.', 0),**('N F', 1)**,('rs', 0),('o!', 0),**('rd', 1)**,('wo', 0),('rg', 0),('On', 0),('ra', 0),('a(', 0),**('it', 1)**,('rm', 0),('hí', 0),('ri', 0),('eS', 0),('hO', 0),('wg', 0),('EL', 0),('we', 0),('ba', 0),('EE', 0),('oE', 0),('bo', 0),('Oy', 0),('R N', 0),('ww', 0),**('ud', 1)**,**('re', 1)**,('V A', 0),('EQ', 0),('ja', 0),('ES', 0),('oo', 0),**('on', 1)**,('om', 0),('ol', 0),('rí', 0),**('oc', 1)**,**('D V', 1)**,('oy', 0),('UE', 0),('r.', 0),('ot', 0),**('os', 2)**,('Es', 0),('op', 0),('!!', 0),('xk', 0),**('ci', 1)**,('lm', 0),**('5y', 1)**,('6)', 0),**('ca', 1)**,('R V', 0),('R A', 0),('F V', 0),('D D', 0),('S.', 0),('cr', 0),('xq', 0),('Z S', 0),**('cu', 1)**,**('ad', 1)**,('pr', 0),('V D', 0),('R D', 0),('V F', 0),('cO', 0),('D A', 0),('..', 0),('1y', 0),('V N', 0),('r,', 0),('ee', 0),('pa', 0),(';-', 0),('eñ', 0),**('D N', 1)**,('iÓ', 0),('pi', 0),('pl', 0),('ls', 0),('aj', 0),('an', 0),('V S', 0),('c,', 0),('(8', 0),('UG', 0),('mO', 0),('z,', 0),(')u', 0),('.E', 0),('ht', 0),('.G', 0),('3-', 0),('hh', 0),(',e', 0),**('A N', 1)**,('Sa', 0),**('vo', 1)**,('Se', 0),('he', 0),('me', 0),('Co', 0),('!G', 0),('uz', 0),('mo', 0),('mi', 0),('us', 0),('ur', 0),('mu', 0),('mt', 0),('un', 0),('SO', 0),('mp', 0),**('ue', 1)**,('au', 0),('ug', 0),('uf', 0),('ua', 0),('SE', 0),('aa', 0),('va', 0),**('ac', 1)**,('ab', 0),('ae', 0),('<3', 0),('DI', 0),**('F F', 1)**,('ah', 0),('is', 0),('ir', 0),('am', 0),('al', 0),('iv', 0),('as', 0),('ar', 0),('im', 0),('at', 0),('io', 0),('in', 0),('ia', 0),('ax', 0),('ic', 0),('if', 0),**('ni', 1)**,('Ón', 0),('nl', 0),('N N', 0),('no', 0),('LO', 0),('De', 0),('nd', 0),('ne', 0),('nf', 0),('aS', 0),('LU', 0),('s.', 0),('iO', 0),('.;', 0),('Ta', 0),('nt', 0),('nu', 0),('a!', 0),('fr', 0),('Od', 0),('AD', 0),('S N', 0),('Z N', 0),('83', 0),('Oc', 0),('ña', 0),**('Z C', 1)**,**('S D', 1)**,('S A', 0),('Z F', 0),('a1', 0),('sJ', 0),('du', 0),('AR', 0),**('a5', 1)**,('Oa', 0),('IO', 0),('sC', 0),('fi', 0),('ro', 0),**('a:', 1)**,('a<', 0),('fo', 0),('Aa', 0),('C Z', 0),(',d', 0),('sx', 0),('QU', 0),(',a', 0),**('sr', 1)**,('sp', 0),('ko', 0),('Si', 0),('su', 0),('st', 0),('A Z', 0),('si', 0),('sh', 0),('so', 0),**('V Z', 1)**,('sm', 0),('na', 0),('A S', 0),('sa', 0),(',y', 0),**('C D', 1)**,('sf', 0),('se', 0),**('sd', 1)**]

$vMuestra_{m10}$	[('d,', 0),('gr', 0),**('5T', 1)**,('íe', 0),('N V', 0),(');', 0),('ía', 0),('tO', 0),('ge', 0),('ch', 0),('aO', 0),('ga', 0),('N D', 0),**('y2', 1)**,('N A', 0),(':)', 0),('ld', 0),('le', 0),('lb', 0),**('la', 2)**,('Wo', 0),**('tu', 1)**,**('tr', 1)**,**('li', 1)**,('lv', 0),('to', 0),('lu', 0),**('av', 1)**,('RA', 0),('ti', 0),('C N', 0),**('te', 1)**,('F R', 0),('Ja', 0),('ta', 0),('do', 0),('Om', 0),('dm', 0),('GR', 0),**('di', 1)**,('ya', 0),('Of', 0),('Oe', 0),**('de', 1)**,('ye', 0),('yg', 0),('da', 0),**('N S', 1)**,('GA', 0),('ma', 0),('R.', 0),('Ot', 0),('A F', 0),('dO', 0),('yl', 0),('OL', 0),**('qu', 1)**,('Gr', 0),('l!', 0),('OE', 0),('-)', 0),**('2d', 1)**,**('l5', 1)**,('A D', 0),('Bo', 0),('uh', 0),('OS', 0),**('-8', 1)**,**('em', 1)**,**('el', 2)**,**('en', 1)**,('ei', 0),('eh', 0),('F S', 0),('be', 0),('ve', 0),('ed', 0),('fe', 0),**('ea', 1)**,('ow', 0),('ec', 0),**('F Z', 1)**,('F D', 0),('i:', 0),('F C', 0),**('et', 1)**,('iu', 0),('F N', 0),**('86', 1)**,('ep', 0),**('es', 1)**,('er', 0),('rt', 0),('o.', 0),**('N F', 1)**,('rs', 0),('o!', 0),('rd', 0),('wo', 0),('rg', 0),('On', 0),**('ra', 1)**,**('a(', 1)**,('it', 0),('rm', 0),('hí', 0),('ri', 0),('eS', 0),('hO', 0),('wg', 0),('EL', 0),('we', 0),('ba', 0),('EE', 0),('oE', 0),('bo', 0),('Oy', 0),('R N', 0),('ww', 0),**('ud', 1)**,('re', 0),('V A', 0),('EQ', 0),('ja', 0),('ES', 0),('oo', 0),**('on', 1)**,('om', 0),('ol', 0),('rí', 0),**('oc', 1)**,**('D V', 1)**,('oy', 0),('UE', 0),('r.', 0),('ot', 0),**('os', 1)**,('Es', 0),('op', 0),('!!', 0),('xk', 0),**('ci', 1)**,**('lm', 1)**,('5y', 0),**('6)', 1)**,**('ca', 1)**,('R V', 0),('R A', 0),**('F V', 1)**,('D D', 0),('S.', 0),('cr', 0),**('xq', 1)**,**('Z S', 1)**,('cu', 0),('ad', 0),('pr', 0),**('V D', 1)**,('R D', 0),('V F', 0),('cO', 0),**('D A', 1)**,('..', 0),**('1y', 1)**,('V N', 0),('r,', 0),('ee', 0),('pa', 0),(';-', 0),**('eñ', 1)**,**('D N', 2)**,('iÓ', 0),('pi', 0),('pl', 0),('ls', 0),('aj', 0),**('an', 1)**,('V S', 0),('c,', 0),**('(8', 1)**,('UG', 0),('mO', 0),('z,', 0),**(')u', 1)**,('.E', 0),('ht', 0),('.G', 0),**('3-', 1)**,('hh', 0),(',e', 0),('A N', 0),('Sa', 0),**('vo', 1)**,('Se', 0),('he', 0),**('me', 1)**,('Co', 0),('!G', 0),('uz', 0),('mo', 0),**('mi', 1)**,**('us', 1)**,**('ur', 1)**,('mu', 0),('mt', 0),('un', 0),('SO', 0),('mp', 0),**('ue', 1)**,('au', 0),('ug', 0),('uf', 0),('ua', 0),('SE', 0),('aa', 0),('va', 0),**('ac', 1)**,('ab', 0),('ae', 0),('<3', 0),('DI', 0),('F F', 0),('ah', 0),**('is', 1)**,('ir', 0),('am', 0),**('al', 2)**,('iv', 0),('as', 0),('ar', 0),('im', 0),('at', 0),**('io', 2)**,**('in', 1)**,('ia', 0),**('ax', 1)**,('ic', 0),('if', 0),('ni', 0),('Ón', 0),**('nl', 1)**,('N N', 0),('no', 0),('LO', 0),('De', 0),('nd', 0),**('ne', 1)**,('nf', 0),('aS', 0),('LU', 0),('s.', 0),('iO', 0),('.;', 0),**('Ta', 1)**,**('nt', 1)**,('nu', 0),('a!', 0),('fr', 0),('Od', 0),('AD', 0),('S N', 0),**('Z N', 1)**,**('83', 1)**,('Oc', 0),**('ña', 1)**,**('Z C', 1)**,**('S D', 3)**,('S A', 0),**('Z F', 1)**,**('a1', 1)**,('sJ', 0),**('du', 1)**,('AR', 0),('a5', 0),('Oa', 0),('IO', 0),('sC', 0),('fi', 0),**('ro', 1)**,('a:', 0),('a<', 0),('fo', 0),('Aa', 0),**('C Z', 1)**,(',d', 0),('sx', 0),('QU', 0),(',a', 0),('sr', 0),('sp', 0),('ko', 0),('Si', 0),('su', 0),**('st', 1)**,**('A Z', 1)**,('si', 0),('sh', 0),('so', 0),**('V Z', 1)**,('sm', 0),**('na', 1)**,('A S', 0),('sa', 0),(',y', 0),('C D', 0),('sf', 0),**('se', 3)**,('sd', 0)]

La matriz de entrenamiento obtenida al generar todos los vectores de muestra es del orden:

$$matrizEntrenamiento = \{vMuestra_m | m \in M\} \times vSistema = 10 \times 308$$

Esta matriz de entrenamiento creada es la que permitirá entrenar al sistema clasificador.

8.2. Matrices de confusión

Las matrices de confusión son una herramienta que permiten visualizar el comportamiento de los resultados obtenidos a partir de un algoritmo de aprendizaje. Cada columna de la matriz hace referencia a las predicciones realizadas de las muestras por el algoritmo mientras que las filas hacen referencia a los valores reales de las muestras (Figura 8.1).

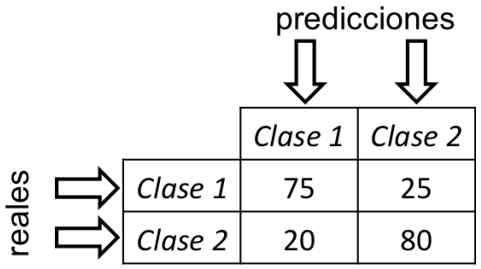

Figura 8.1: Ejemplo de matriz de confusión

Como se aprecia en la Figura 8.1, existe un total de 200 muestras de las cuales 100 pertenecen a la $Clase1$ y 100 a la $Clase2$. En lo que respecta a la $Clase1$, de las 100 muestras existentes, 75 fueron clasificadas correctamente ($Clase1$) y 25 incorrectamente ($Clase2$). Para la $Clase2$, 80 muestras fueron clasificadas correctamente ($Clase2$) y 20 incorrectamente ($Clase1$).