Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

SISTEMA DE EXTRACCIÓN DE CARACTERÍSTICAS DEL ESTILO DISCURSIVO ENTRE HOMBRES Y MUJERES EN UN CORPUS DE OPINIONES TEXTUALES EN ESPAÑOL

Alumno: Javier Valiente Martín

Tutor: Prof. Dª. María Dolores Molina González
Dpto: Informática

Septiembre, 2018

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Doña María Dolores Molina González , tutora del Trabajo Fin de Grado titulado: **“Sistema de extracción de características del estilo discursivo entre hombres y mujeres en un corpus de opiniones textuales en español”**, que presenta Javier Valiente Martín, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2018

El alumno:

Fdo.: Javier Valiente Martín

La tutora:

MOLINA GONZALEZ MARIA DOLORES - 26009191D

Firmado digitalmente por MOLINA GONZALEZ MARIA DOLORES - 26009191D
Nombre de reconocimiento (DN): c=ES, serialNumber=26009191D, sn=MOLINA GONZALEZ, givenName=MARIA DOLORES, cn=MOLINA GONZALEZ MARIA DOLORES - 26009191D
Fecha: 2018.09.06 13:33:38 +02'00'

Fdo.: Prof. Dª. María Dolores Molina González

Índice

Capítulo 1. Introducción

En este capítulo se presentan los principales aspectos que han dado origen a la elaboración del proyecto **"Sistema de extracción de características del estilo discursivo entre hombre y mujeres en un corpus de opiniones textuales en español".**

Este trabajo se centra en una de las áreas del Procesamiento del Lenguaje Natural (PLN), el Author Profiling (AP), del cual se han adquirido una serie de conocimientos durante el desarrollo del mismo. También se han adquirido una serie de conocimientos mediante el desarrollo del trabajo sobre una serie de tecnologías relacionadas con el PLN y el Machine Learning, áreas punteras en el ámbito de la Informática en la actualidad y que si se siguen desarrollando como hasta ahora tendrán unos resultados y utilidades muy importantes para diversas necesidades de la sociedad.

1.1. Introducción al proyecto

El desarrollo de la Web 1.0 hacia la Web 2.0 ha hecho que la cantidad de contenido que podemos encontrar en Internet ya sea creado o compartido, crezca de forma exponencial en los últimos años. Además la sociedad está viviendo una serie de cambios socioculturales, ayudada como no, por una revolución tecnológica que está desembocando en que prácticamente personas de cualquier edad y en diferentes contextos utilicen la red para comunicarse, relacionarse, expresarse, etc.

Millones de personas tienen diariamente conversaciones, escriben opiniones en redes sociales, artículos formales y un largo sin fin de contenido, el cual queda almacenado en la web, esto forma parte del denominado *Big Data.* Como ya sabemos todo este material web, suele tener un autor, es decir, la persona que se ha encargado de, por ejemplo, mostrar en Twitter su opinión acerca de un evento tecnológico al que ha asistido. Pero esta persona que crea contenido, que tiene una identidad digital, no siempre es una persona real o coincide con su verdadera identidad: avatares falsos creados para engañar a la gente, querer ocultar su identidad real, hacerse pasar por otra persona, omitir cierta información de sí mismo, etc. Por ejemplo, una persona que

pretenda mantenerse en el anonimato para realizar un ciberdelito, o un usuario falso que pretenda cometer ciberacoso sexual a un menor son algunos de los casos en que el autor del contenido no es real.

Pero a pesar de todo ello, hay una cosa que siempre está ahí, y esto es el texto que escribe un usuario. El estilo discursivo que conlleva el escribir un texto en internet refleja, aunque puede ser de manera inconsciente por parte del autor, su perfil, las palabras que utiliza y el modo en que las usa.

A raíz de ello, nace el AP, cuyo principal propósito es el de averiguar la mayor información sobre el autor simplemente analizando lo que escribe. Esta es una tarea que se encarga de analizar y estudiar diferentes rasgos psicolingüísticos y sociológicos, como por ejemplo el uso del lenguaje o los rasgos compartidos por grupos similares, con el fin de predecir el mayor número de características del autor del texto, el cual es desconocido, como puede ser su edad, sexo, estado emocional o personalidad entre otras, a partir simplemente del texto plano.

Actualmente AP es una tarea muy puntera en el ámbito de la investigación debido a su enorme utilidad en diversos campos (crimen, marketing, seguridad, etc.). El avance de esta técnica supondría un enorme avance científico e industrial, además de una gran responsabilidad para la persona que tuviera que utilizarla.

Una de las áreas que está destacando en interés dentro del AP es la identificación del género del autor, tema sobre el que trata este trabajo.

1.2. Motivación

Poder darle una utilidad a la ingente cantidad de información que podemos encontrar en la actualidad en Internet, es uno de los objetivos del Author Profiling, el cuál puede ayudar de manera considerable en diferentes campos. En un ámbito policial puede ayudar a detectar ciber-pedófilos o ciber-acosadores, o para detectar la veracidad de una carta de suicidio. En el ámbito de los negocios y el marketing podría servir para saber qué tipo de compradores están interesados en un producto determinado, lo que los consumidores prefieren, etc.

Uno de los principales motivos para la elección de este proyecto es la falta de herramientas especializadas en desarrollar la tarea de AP y específicamente de la identificación del autor en idioma español, segunda lengua más hablada del mundo[1] y tercera lengua más utilizada en internet[2] como se muestra en la Ilustración 1.1.

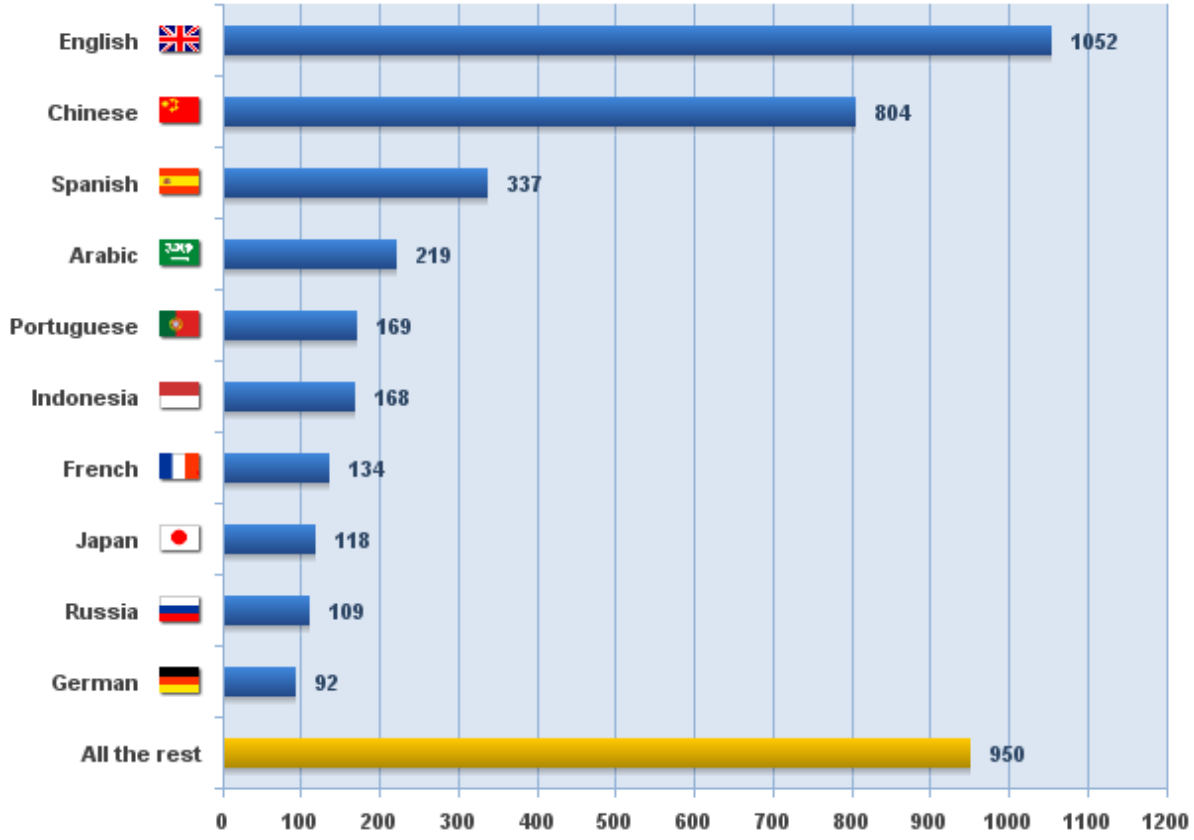


Ilustración 1.1. Millones de usuarios de Internet por idioma (2017).

Esto último hace que la cantidad de información escrita online que se puede utilizar para el análisis sea muy grande. Sin embargo, existe una gran escasez de corpus de documentos escritos en español y etiquetados por el género del autor, ya que la mayoría de ellos son en inglés.

Por ello, se ha decidido buscar un corpus en español etiquetado por género para poder realizar una herramienta encargada de extraer características de cada

[1] https://es.weforum.org/agenda/2018/02/cuales-son-los-idiomas-mas-hablados-en-el-mundo
[2] https://www.internetworldstats.com/stats7.htm

documento y clasificar cada uno de ellos según si ha sido escrito por una persona del género masculino o femenino.

Este proyecto también me aportará el conocimiento de nuevas técnicas que no se han enseñado durante el grado y eran desconocidas para mí. Como primera tarea, realizaré una amplia investigación acerca del AP sobre todo para la identificación del género del autor y por último realizaré una parte más técnica y práctica mediante el diseño de un sistema para extraer características de diferentes textos formales y clasificarlos por sexo del autor. Desarrollaré más ampliamente mis conocimientos sobre Python, dado en el Grado de Ingeniería Informática brevemente.

1.3. Propósito del proyecto

El propósito principal de este proyecto es desarrollar una herramienta capaz de extraer las características del modelo de escritura entre hombres y mujeres de un corpus de opiniones textuales etiquetado por género, y posteriormente clasificar otros textos en función del género del autor.

Para ello, se realizará un estudio de los corpus etiquetados por género existentes en español, con el objetivo de tomarlos como base y se aplicarán diferentes técnicas de PLN.

1.4. Objetivos del proyecto

Los objetivos que se pretenden llevar a cabo durante el desarrollo de este proyecto son:

1. Estudiar la bibliografía relacionada con Author Profiling.
2. Realizar un estudio de los corpus existentes en español etiquetados por género.
3. Estudiar diferentes técnicas para la extracción de características en textos.
4. Implementar una herramienta que incorpore alguno de los métodos de clasificación estudiados para mostrar la utilidad del recurso generado.

5. Redactar una memoria que recoja todo el trabajo desarrollado, así como los manuales de instalación y de usuario.

1.5. Resultados esperados

Una vez finalizado el proyecto, los resultados que deben obtenerse son:

1. Investigación acerca de la clasificación de textos según el género del autor (masculino o femenino).
2. Elección de un corpus de opiniones textuales etiquetadas según el sexo de su autor de un portal web de entre las diferentes posibilidades.
3. Investigación de las características propias del estilo discursivo de hombres y mujeres.
4. Diseño de un sistema que clasifique opiniones por sexo.
5. Redacción de la memoria del proyecto.

1.6. Planificación

El objetivo de la planificación de un proyecto es ajustar un ámbito de trabajo, de manera que facilite estimar de manera razonable el tiempo y los costes que tendrá el mismo. Estas estimaciones son flexibles y pueden ir adaptándose a la situación del proyecto.

1.6.1. Estimación temporal

La planificación temporal se llevará a cabo mediante un diagrama de Gantt, herramienta gráfica que se utiliza con la finalidad de mostrar el tiempo que se le va a dedicar a un conjunto de tareas dentro de un tiempo determinado.

A continuación se muestran las tareas en que se ha dividido el proyecto junto con la duración de cada una de ellas:

- Estudio y comparación de los corpus existentes actualmente, etiquetados por género. (01/02/2018 – 02/04/2018).
- Estudio de diferentes técnicas para la clasificación de textos. (03/04/2018 – 18/05/2018).

- Estudio de las distintas estructuras y arquitecturas a utilizar para almacenar y gestionar la información. (03/04/2018 – 18/05/2018).
- Diseño e implementación de la herramienta de análisis de los textos usando las técnicas de clasificación seleccionadas. (19/05/2018 -- 29/06/2018).
- Realización de pruebas para la experiencia de usuario, así como de estabilidad y de seguridad. (30/06/2018 – 09/07/2018).
- Desarrollo de los manuales de usuario y de instalación (10/07/2018 – 16/07/2018).
- Generación de una memoria con todo el trabajo realizado (02/04/2018 – 03/08/2018)

Estas tareas tienen asignado unas fechas limitadas, aunque se pueden ir adaptando a medida que se desarrolle en el proyecto.

A continuación se muestra el gráfico mencionado anteriormente (Ilustración 1.2.). En él se muestran las tareas listadas y la duración de cada una de ellas y ha sido creado mediante una web online[3].

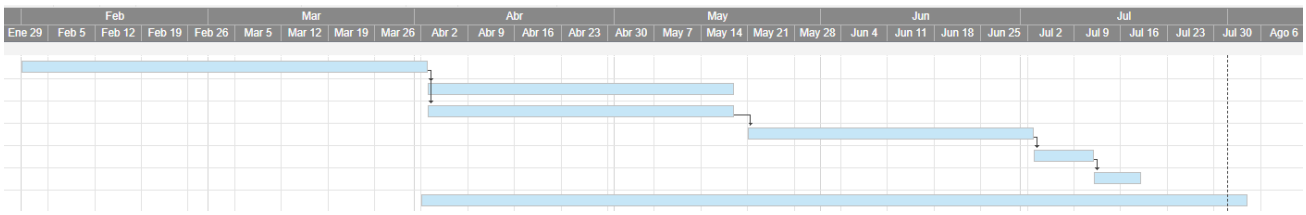


Ilustración 1.2. Diagrama de Gantt

El periodo de tiempo que se estima que ocupe la realización de este proyecto es de 6 meses.

[3] https://es.smartsheet.com/software-diagrama-gantt

1.6.2. Estimación de costes

Cualquier proyecto software implica una serie de costes durante el desarrollo, e igualmente para su posterior mantenimiento. Para el cálculo de estos costes hay que tener en cuenta la duración que tendrá el proyecto para poder tener un cálculo de costes durante el desarrollo.

1.6.2.1. Costes hardware

Estos costes representan los costes asociados al equipamiento físico necesario para el desarrollo del proyecto. A continuación se muestran las especificaciones utilizadas para el desarrollo del proyecto, aunque no tiene por qué ser ese equipamiento específicamente para poder ejecutar el sistema creado.

- Ordenador portátil: Lenovo B50-70 80EU2 ……. 599€
    - Procesador: Intel Core i5-4210U (1.7GHz, 3MB).
    - Memoria RAM / HDD: 8GB / 500GB.
    - Tarjeta gráfica: NVIDIA GeForce 840M.
    - Pantalla: 15.6” LED-Backlight HD (1366x768).

El coste mostrado se debe amortizar durante el desarrollo. Se estima una vida útil de 6 años, una vez pasado ese tiempo, el valor será de 0€. Obtenemos un coste de 99,83€ al año, por lo tanto 8,32€ al mes. Dado que el proyecto tiene una duración estimada de 5 meses el coste total sería de 41,6€.

1.6.2.2. Costes software

Estos costes engloban los necesarios para adquirir las diversas licencias de programas necesarios para la realización del proyecto.

Los programas utilizados y que producen los costes de software son los siguientes:

- Ubuntu 16.04 ……..…..... 0€
- Spyder (Anaconda 3) ……..…… 0€
- Navegadores web …… 0€

- Google Drive ……..…….. 0€
- Microsoft Office 2013 ……..……. 7€/mes

Estos costes al igual que los costes de hardware se tienen que amortizar durante el desarrollo del proyecto. Teniendo en cuenta que el software de Microsoft Office 2013 se utilizará durante toda la duración del proyecto, es decir, 5 meses, su coste sería de 35€, que será el coste total de software.

1.6.2.3. Costes de personal

Se trata del sueldo del personal que se necesita contratar para el desarrollo del proyecto. En este caso, será realizado por una única persona, aunque tendrá diferentes papeles dentro del mismo. Cada uno de ellos tendrá unos costes.

Se necesita el siguiente personal:

- **Analista**: realizará el papel de analizar el sistema, diseñarlo y posteriormente intentar mejorar la eficiencia del mismo.
- **Programador:** encargado de implementar el sistema.
- **Técnico:** encargado de realizar los diferentes manuales necesarios para el uso del sistema, realizar pruebas y evaluar el sistema.

Los salarios anuales de cada miembro del personal se muestran a continuación y han sido obtenidos BOE del 19 de Junio de 2013[4].

Para todos ellos se establece un máximo de 39 horas semanales. Este dato se encuentra en la misma web citada para los salarios.

Teniendo un total de 300 horas para realizar el proyecto, el coste en personal sería el mostrado en la siguiente tabla (Tabla 1.1.)

Personal	Salario anual	Salario/hora	Nº horas	Total
Analista	25.830€	14,16€	75	1.062€
Programador	20.734€	11,37€	200	2.274€
Técnico	15.554 €	8,53€	25	213,25€

[4] https://www.boe.es/diario_boe/txt.php?id=BOE-A-2013-6678

Tabla 1.1. Costes de personal

Por tanto, el coste total de personal será de 3.549,25€.

1.6.2.4. Otros costes

Otros costes que no han sido tenidos en cuenta hasta el momento son los producidos por la conexión a Internet, que tendrá un coste de 12,85€/mes y un total de 50Mb con Vodafone.

El coste total será el de la conexión durante los 5 meses que durará el proyecto, es decir, 64,25€.

1.6.2.5. Coste total

El coste total del proyecto viene determinado por la suma de todos los costes expuestos anteriormente (costes de hardware, costes de software, costes de personal y otros costes). Además a estos costes se considera añadir un 10% de beneficio. En la siguiente tabla (Tabla 1.2.) se muestran un resumen de los costes totales.

Concepto	Coste
Costes hardware	41,6€
Costes software	35€
Costes de personal	3.549,25€
Otros costes	64,25€
TOTAL	3.690,1€

Tabla 1.2. Coste total del proyecto

Añadiendo un el 10% de beneficio, el coste total será de 4.059,11€.

1.7. Estructura del proyecto

A continuación se explica la estructura que tiene la memoria de este proyecto. Esta memoria está formada por seis capítulos:

- En el capítulo 1, titulado “Introducción”, se han explicado los propósitos y objetivos, además de la motivación del trabajo fin de grado (TFG). También se explica la planificación y los costes del mismo.
- En el capítulo 2, titulado “Clasificación de textos en función del género del autor”, se lleva a cabo un resumen del estado del arte de la tarea que se va a llevar a cabo.
- En el capítulo 3, titulado “Estudio y análisis de los recursos en español”, se muestran los diferentes recursos que se han estudiado para la clasificación de textos según género del autor. Se muestran las principales características de cada uno de los recursos.
- En el capítulo 4, titulado “Estudio de técnicas para la clasificación de género”, se realiza una revisión de las diferentes técnicas que se han utilizado o que están disponibles para realizar el proyecto acerca de la clasificación de texto según el género del autor.
- En el capítulo 5, titulado “Desarrollo del proyecto”, como su propio nombre indica se explica el proceso creación del proyecto. Primero se detalla la descripción, los objetivos y requerimientos del proyecto. Posteriormente se elige una metodología de Ingeniería del Software y se enumeran las diferentes historias de usuario que forman el proyecto. Por último se detalla el desarrollo de cada una de las historia de usuario y se comentan las diferentes tecnologías utilizadas para el desarrollo del sistema.
- En el capítulo 6 y último, titulado “Conclusiones y trabajos futuros”, se exponen las conclusiones y valoración personal que se han obtenido del proyecto además de los posibles trabajos a desarrollar en el futuro para mejorar el sistema.
- En el apartado titulado “Bibliografía”, se indican los diferentes materiales que se han consultado y utilizado para la realización del trabajo.
- Por último hay cuatro anexos:
    - Anexo A: muestra como instalar el sistema creado durante el desarrollo de este trabajo de manera que quede listo para su utilización.

- Anexo B: muestra un breve manual para que el usuario pueda utilizar el sistema creado, tanto el extractor de características como el clasificador de textos en función del género del autor.
- Anexo C: muestra un breve manual para el mantenimiento del sistema desarrollado en este trabajo, donde se explica cómo se podrían cambiar los lexicones utilizados por otros al gusto del usuario.
- Anexo D: contiene el índice de las ilustraciones que aparecen en el proyecto.
- Anexo E: contiene el índice de las tablas que aparecen en el proyecto.

Capítulo 2: Clasificación de textos en función del género del autor

La identificación automática de autoría (Automatic Authorship Identification (AAI)) es un área de investigación del Procesamiento del Lenguaje Natural (PLN). En este capítulo se realiza una revisión del estado del arte de una parte concreta de este área, la clasificación de textos en función del género del autor.

AAI está formada por tres campos: atribución de autoría, identificación del autor y perfilado del autor. Los dos primeros campos se centran en obtener el autor de cada documento partiendo de un conjunto de autores donde elegir. Sin embargo, el perfilado del autor (AP) se centra en obtener características específicas del autor que ha escrito cada documento: edad, sexo, ocupación, estado emocional, etc., a partir de diferentes rasgos como por ejemplo el uso del lenguaje, ya sea la estructura o el contenido del mismo.

El AP es una tarea que aún sigue abierta debido a su enorme dificultad, y es por ello que multitud de autores siguen investigando diferentes áreas o espacios que pueden producir una mejora en los resultados. A pesar de ello ya se han realizado un gran número de estudios que han obtenido buenos resultados como el de Penneaker *et al*. (2003), Soler Company & Wanner (2015) o Sap *et al.* (2014) entre otros.

Debido a la finalidad del trabajo que se ha desarrollado, el estado del arte que se muestra a continuación se centra en el AP dirigido a obtener el género del autor (masculino o femenino).

Las investigaciones realizadas por Penneaker *et al.* (2003) son gran parte de la base de los trabajos realizados de AP en cuanto a la identificación del género. Han investigado como se puede obtener información sobre el género del autor de un texto estudiando la variación de diferentes características lingüísticas, estudiando la conexión que existe entre el uso del lenguaje y rasgos de la persona. Estas investigaciones fueron realizadas para el inglés.

Este estudio de Penneaker obtuvo algunas conclusiones, entre ellas destacan:

- Las mujeres utilizan más la primera persona del singular debido a que se preocupan más por ellas mismas. Por otro lado la primera persona del plural si se utiliza en la misma medida en ambos géneros.
- Los hombres hablan de cosas más concretas que las mujeres, lo que les lleva a utilizar más artículos determinados e indeterminados.
- Sin embargo, las mujeres utilizan más palabras cognitivas y sociales debido a que se centran más en las personas (Ej. pensar, entender, etc.).

En cuanto a los trabajos de AP, son pioneros los trabajos de Argamon y Koppel en el año 2003. Ambos se centran en corpus equilibrado de documentos formales extraídos del British National Corpus (BNC)[5]. Sin embargo el primero de ellos, Argamon *et al.* (2003) utilizaron un total de 1.081 características basándose principalmente en palabras de función, que son aquellas que no tienen un significado concreto pero que sirven para comunicarnos ya que relacionándose con otras palabras dan un significado, relacionadas con partes del discurso, que son categorías de palabras que tienen unas propiedades gramaticales similares (Ej. nombres, verbos, adjetivos, pronombres, etc.). Obtuvieron un 80% de accuracy en la identificación del sexo.

Por otro lado Koppel *et al.* (2003) utilizaron un conjunto de palabras de función y n-gramas de 76 etiquetas referentes a partes del discurso y signos de puntuación, obteniendo diferentes valores para la accuracy ya que se hicieron varios experimentos: se obtuvo un 73,7% usando palabras de función, 70,5% usando los n-gramas y un 77,3% mediante la combinación de todas las características.

Estos primeros trabajos, todos ellos en inglés demuestran la aportación de las palabras de función y partes del discurso en la identificación del género del autor.

Más tarde, Estival *et al.* (2007) llevaron a cabo un trabajo utilizando un corpus menos formal que los pioneros, basado en un conjunto de correos electrónicos en inglés. Este trabajo se centra en un grupo de 689 características formadas por

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

frecuencias de signos de puntuación, longitud de las palabras, letras mayúsculas, palabras de función, etiquetas HTML y partes del discurso. Obtuvieron peores resultados que los pioneros con un 56.46% utilizando máquinas de vectores de soporte (Support Vector Machines (SVM)) como algoritmo de clasificación.

Posteriormente, con la llegada de las redes sociales, la tarea del AP se ha centrado en un ámbito más coloquial, menos formal y menos estructurado. Algunos de los estudios más destacados en este ámbito en inglés se explican a continuación.

Schler *et al.* (2006) llevaron a cabo un estudio basándose en un corpus formado por 70.000 blogs y un conjunto de características formado por palabras fuera de diccionario, partes del discurso y palabras de función e hiperenlaces, obteniendo un 80% de accuracy. En el mismo año Yan & Yan (2006) clasificaron un corpus formado por 75.000 posts en inglés utilizando Naive Bayes y utilizando bolsas de palabras junto con elementos extraídos del HTML como el color del fondo, emoticonos ligados a emociones, tipografía, etc., obteniendo un 73,11% de accuracy

Tres años más tarde, Goswami *et al.* (2009) usando el mismo corpus utilizado por Scheler obtuvieron un 89,2% de accuracy, añadiendo al conjunto de características utilizado por Scheler palabras de jerga o la longitud de oraciones.

Mukherjee & Liu (2010) realizaron un estudio en el que se propuso un método que permitía obtener patrones de secuencias de partes del discurso con longitud variables, basándose en un corpus formado por 3.100 blogs etiquetados manualmente por sexo. Su accuracy fue del 88,5%.

En el mismo año, Otterbacher (2010) realizó un estudio utilizando como corpus un conjunto de revisiones de películas en inglés. Además de utilizar las características lingüísticas de las revisiones (uso de pronombres complejidad del vocabulario, uso de personas y tiempos verbales o palabras relacionadas con la familia, violencia o relaciones) también se utilizaron metadatos como la valoración o la fecha de publicación. Obtuvo un 73,71% de accuracy utilizando un clasificador de regresión logística y utilizando la combinación de todas las características disponibles.

También destacan dos trabajos, que utilizaron Facebook para obtener el corpus para realizar la tarea. Estos son los estudios de Schwartz *et al.* (2013) y Sap *et al.* (2014). El primero de ellos utiliza n-gramas y tópicos obtenidos como características para identificar el sexo del autor de usuarios de Facebook. Obtienen un 91,9% de accuracy entrenando su sistema con 700 millones de palabras. En el segundo consiguieron obtener el mismo porcentaje de accuracy (91,9%), basándose en un corpus formato por 75.394 posts de Facebook y utilizando clasificación con máquinas de vectores de soporte.

Todos estos estudios se basan en el idioma inglés, para el que, como puede observarse hay multitud de trabajos. A continuación se muestran algunos trabajos que destacan en cuanto a su desarrollo de la tarea en español.

Un año más tarde, Maharjan *et al.* (2014) llevaron a cabo un estudio para la clasificación de documentos en función del género del autor, utilizando un total de 3 millones de características procesadas en una configuración MapReduce, lo que les permitió obtener resultados competitivos, obteniendo un valor de 64,6% de accuracy para el idioma español. También desarrollo el sistema para el inglés para el cuál obtenían un 61,66% de accuracy.

Un año después, se desarrollaron dos estudios que destacan para la tarea de AP en español. Soler-Company & Wanner (2015) realizaron la tarea utilizando un corpus formado por blogs de la sección correspondiente de periódicos, etiquetado a mano con el sexo de los autores. Realizaron la tarea para varios idiomas además del español: inglés, alemán, francés, catalán e italiano. Para la clasificación utilizaron un conjunto de 27 características que incluían primeras personas del singular y del plural, nombres propios, ratio de uso de comas, de interrogaciones, de exclamaciones, frecuencia de interjecciones, media de palabras por oración, media de oraciones por post, palabras de afirmación y negación, palabras vacías o características sintácticas extraídas del árbol de dependencias. La accuracy media que obtuvieron entre todos los idiomas fue del 71,01%, sin embargo, para el español obtuvieron un resultado bastante por encima de la media, con un 88% de accuracy.

Lopez-Monroy *et al.* (2015) propusieron un nuevo método, que consiste en buscar información específica de subperfiles dentro de los perfiles y explotarla. Por ejemplo en lugar de utilizar los perfiles hombres y mujeres, utilizar hombres deportistas, mujeres amas de casa, mujeres adolescentes estudiantes, etc. De esta manera se admite una heterogeneidad entre los perfiles que en un principio parecían perfiles homogéneos y así se consigue información específica de cada subperfil que puede ayudar a clasificar los documentos dentro del conjunto de perfiles. Los autores han testado su sistema con los datasets que se proporcionaron para la competición del PAN en el año 2013 basado en social media en inglés y español, obteniendo un 55,39% y un 66,35% respectivamente para cada idioma. También probaron el sistema con el corpus propuesto para el PAN de 2014, que incluía social media, blogs y Twitter en español, además de revisiones de hotel en inglés. En este caso obtuvieron los resultados que se muestran en la siguiente tabla:

	Social Media	Blogs	Twitter	Revisiones hotel
Inglés	55,39%	78,2%	71,6%	69,27%
Español	66,35%	64,77%	66,85%	-

Tabla 2.1. Resultados de Lopez-Monroy con el corpus del PAN2014.

Por último cabe destacar la labor del PAN[6] que es una organización que se encarga de realizar competiciones en diversas tareas. A partir de 2013 se incluyó el Author Profiling para la clasificación de textos en función del género del autor como una de sus tareas a investigar y comenzaron a realizarse competiciones también sobre esta tarea. Estas competiciones hacen que multitud de grupos repartidos por todo el mundo investiguen acerca de esta tarea consiguiendo cada vez mejores resultados.

[6] https://pan.webis.de/index.html

Capítulo 3: Estudio y análisis de los recursos en español

En este capítulo se va a analizar y explicar los recursos existentes para la creación de un clasificador de textos en función del género del autor. Se realizará una pequeña introducción de los diferentes recursos y se explicarán sus principales características.

3.1. Corpus

Se define como corpus a una colección de textos representativos de una lengua, de un dialecto o un subconjunto de un lenguaje, disponible en formato electrónico (W.Francis & H.Kucera, 1982). En el área del PLN es uno de los principales recursos que se utilizan, debido a la gran información que incluyen. Existen diferentes tipos de corpus en función de sus características. De esta manera, existen corpus textuales u orales, monolingües o multilingües, de propósito general o centrados en un dominio, etiquetados o no etiquetados. En nuestro caso para desarrollar el sistema nos centraremos en un corpus, SpanText que se explicará a continuación, etiquetado en función del género autor del texto, monolingüe (en idioma español), de propósito general y textual.

Encontrar un corpus de documentos etiquetado conforme se ha explicado anteriormente, equilibrado en cuanto al número de textos para cada sexo y el principal inconveniente, que estuviera en español, ha sido una de las grandes dificultades del trabajo, ya que hay una gran falta de colecciones de textos fiables para realizar esta tarea de Author Profiling.

Los corpus que se han conseguido y se ha valorado su utilización se explican a continuación.

3.1.1. Author Profiling en PAN

Existen diferentes corpus proporcionados para las diferentes competiciones de AP que organiza el PAN y que se pueden descargar para usar de manera gratuita.

3.1.1.1. PAN 2013

Corpus construido buscando en repositorios abiertos y públicos, donde sus usuarios se etiquetan con información como la edad y el sexo. La versión en español está formada por tres partes: entrenamiento (75.900 documentos), preevaluación (6.800 documentos), evaluación (8.160 documentos). Se encuentra etiquetado por sexo y edad.

3.1.1.2. PAN 2014

Corpus formado a partir de: Twitter, social media, blogs y revisiones de hotel. Todo el corpus se encuentra en español e inglés, menos las revisiones de hotel que solo están en inglés. Al igual que el anterior, la versión en español también se encuentra etiquetado por sexo y edad y se encuentran dividido en las mismas 3 partes principales: 1.538 documentos para el entrenamiento, 162 documentos para la preevaluación y 712 documentos para la evaluación.

3.1.1.3. PAN 2015

En 2015, el corpus proporcionado por el PAN para la competición fue extraído de Twitter y se encontraba en cuatro idiomas: inglés, español, italiano y holandés. Se encuentra equilibrado por sexo y etiquetado por edad y sexo al igual que los anteriores en el caso del español. Además también está formado por 3 grupos: entrenamiento, preevaluación y evaluación con 110, 30 y 88 documentos respectivamente.

3.1.1.4. PAN 2016

El año 2016 el corpus que se ofreció para realizar la tarea estaba formado por un conjunto de tweets para el entrenamiento (debido a la privacidad de Twitter solamente se proporciona el enlace del tweet), textos de redes sociales para la preevaluación y de blogs para la evaluación. Para cada una de estas fases se proporcionan 250, 64, 56 documentos en español. Este corpus se proporciona en inglés, español y alemán, estado etiquetado por género y edad para los dos primeros y solamente con el género para el último.

3.1.1.5. PAN 2017

En 2017, el corpus estaba centrado en un conjunto de tweets, con la novedad de que se puede encontrar en español, inglés, portugués y árabe. Como los anteriores se encuentra etiquetado con género y también se proporcionan variaciones específicas de la lengua nativa del autor. El corpus contiene 500 autores, divididos en 300 para el entrenamiento y 200 para el test. Cada autor contiene 100 tweets.

3.1.2. SpanText

SpanText se trata de un corpus en español creado por el Laboratorio de Investigación y Desarrollo en Inteligencia Computacional de la Facultad de Ciencias Físico, Matemáticas y Naturales, de la Universidad Nacional de San Luis (Argentina). Sus autores son María Paula Villegas, María José Garciarena Ucelay, Marcelo Luis Erreclade y Leticia Cecilia Cagnina.

Se trata de una colección de 1000 textos formales de diversos temas en español, etiquetados con la edad y el sexo del autor. Los documentos fueron recuperados de la web y escritos por diferentes autores. Además fueron etiquetados manualmente.

Lo que motivó a este grupo a crear un nuevo corpus fue el proporcionado para la competición del PAN de 2013, el cual tenía una enorme cantidad de ruido debido a que se formó mediante una recuperación de documentos de la web 2.0 (semántica incorrecta o incomprensible, errores gramaticales, sintácticos, etc.). Esto no permitía obtener los mejores resultados que si se podrán obtener con un nuevo corpus que no tuviera tanto ruido y estuviera más limpio. Debido a ello y por la falta de algún otro corpus en español crearon este nuevo corpus, llamado *SpanText* por sus creadores.

Este recurso proporciona dos versiones del corpus: una versión balanceada con el mismo número de documentos masculinos y femeninos, e igual para las edades, y otra versión no balanceada, en la que no es el mismo número de documentos para cada categoría.

3.2. Lexicón

Se denomina lexicón a un diccionario o listado de términos representativos de una lengua, región, materia, etc. En el ámbito del PLN, sería un conjunto de términos que representan, por ejemplo sentimientos, emociones, etc. Los lexicones surgieron debido a que con la existencia de corpus no etiquetados, se tenía una enorme dificultad a la hora de tomar decisiones acerca del tema, género del autor, sentimiento, emoción, etc., tratado en dichos documentos. Los lexicones se pueden obtener de tres formas diferentes:

- Método manual: este método se ha utilizado durante el desarrollo de este trabajo para la creación de lexicones de palabras identificativas de hombres y mujeres.
- Método basado en diccionario: se basa en coger un conjunto de palabras con orientación conocida (semillas) e ir incrementando el número de palabras a partir de algún diccionario o base de conocimiento léxico, teniendo en cuenta por ejemplo los antónimos (con significado contrario a la semilla) y sinónimos (con el mismo significado al de la semilla).
- Método basado en corpus: a partir de una lista de palabras se intenta buscar otras relacionadas con ellas, en un corpus de dominio específico.

Los diferentes lexicones que se han generado o explorado para la realización del presente trabajo son los explicados a continuación.

3.2.1. Diccionario de palabras masculinas y femeninas

Para la realización de este proyecto se ha decidido crear nuestro propio diccionario de palabras propias de hombres y mujeres. Se han creado dos listas de palabras, que se muestran a continuación, y que cada una se ha guardado en un archivo .txt:

- Lexicón de palabras más identificativas del género masculino, formado por un total de 143 palabras.
- Lexicón de palabras más identificativas del género femenino, formado por un total de 147 palabras.

Algunos de los factores que han ayudado a decidir que palabras pueden ser factibles de pertenecer a cada una de las listas han sido:

- Los hombres hablan más de cosas concretas con respecto a las mujeres (ej. el motor del coche esta averiado) (Pennebaker *et al.,* 2003*)*.
- Las mujeres, sin embargo, utilizan más palabras cognitivas y sociales que los hombres (ej. pensar, saber, etc.) (Pennebaker *et al.,* 2003).
- Los hombres suelen hablar más de aspectos relacionados con la política o la tecnología (videojuegos, ordenadores. etc.), mientras que las mujeres lo suelen hacer más sobre aspectos de moda o familia (bikini, hijos, etc.). Esto se debe a los gustos que más predominan de cada sexo y sus prioridades.
- También, los hombres son más propensos a decir palabras malsonantes con respecto a las mujeres.
- Las mujeres usan un lenguaje más cálido y agradable que los hombres. También que algunas de las palabras más utilizadas por mujeres están relacionadas con ‘maravilloso’, ‘feliz’, ‘cumpleaños’, ‘hija’, ‘bebe’, ‘agradecida’, ‘ilusionado’ y para los hombres ‘libertad’, ‘ganar’, ‘perder’, ‘enemigo’, ‘batalla’. (Park, Gregory, *et al.,* 2016*)*

Además de la investigación que se llevó a cabo para obtener los factores anteriores, también se realizó una búsqueda en el corpus utilizado (SpanText), escogiendo las primeras 100 palabras que más se repetían (únicamente nombres y adjetivos) para el total de documentos escritos mujeres y de la misma forma para los documentos escritos por hombres. Se han elegido ese número de palabras (100) por que se ha considerado que los 100 primeros nombres o adjetivos son los más significativos del total de todas las palabras que aparecen en el corpus. Posteriormente se hizo una revisión manual de todas ellas, llevando a cabo el siguiente proceso: si una palabra está muy repetida en los textos masculinos y no está o está pocas veces en la lista de palabras más repetidas de los textos escritos por mujeres, se toma la palabra para incluirla en el lexicón de palabras masculinas. De la misma forma pero a la inversa se realiza para obtener palabras que pueden ser significativas para el género femenino. Para el lexicón de hombres algunas de las

palabras añadidas han sido, Android, dólar, fuerza, enemigo, etc. Por otro lado para el lexicón de mujeres han sido, alegría, cumpleaños, familia, hermana, hijo, etc.

3.2.2. iSOL

iSOL (Improved Spanish Opinion Lexicon) es una lista de palabras que indican una opinión positiva o negativa (polaridad) en español e independientemente del dominio que se trate.

Este lexicón es una mejora de uno anterior llamado SOL (Spanish Opinion Lexicon) el cuál se realizó a partir de la lista de palabras que mantiene el profesor Bing Liu (Bing Liu's Opinion Lexicon (BLEL)). La lista de palabras ha sido traducida de manera automática mediante el traductor Reverso y por último fueron corregidas de manera manual.

Sus principales características son:

- Contiene un total de 2.509 palabras positivas y 5.626 negativas.
- Cada palabra está escrita en minúscula, sin acentos y sin caracteres especiales.
- Contiene palabras mal escritas en español, pero que son frecuentemente utilizadas cuando se trata de mostrar una opinión. Esta técnica también era utilizada en BLEL, en la cual que está basado este lexicón.
- Otra técnica que se utilizó fue la de reemplazar manualmente los n-gramas por unigramas. Ej. *brainless* en español se traduce como *sin cerebro,* en este caso usaríamos *descerebrado.*
- No existe ninguna palabra que esté presente en ambas listas (palabras positivas y negativas).
- Cada palabra está en el lexicón en su forma masculina, femenina, singular y plural, en caso de que exista alguna de ellas.

3.2.3. SEL

SEL es un diccionario en español formado por 2.036 palabras, en la que cada una de ellas está asociada con la medida del Factor de probabilidad del uso afectivo

(FPA) a una de las seis emociones básicas, que son: alegría, enojo, miedo, sorpresa, repulsión y tristeza.

Para la creación de este lexicón cada palabra fue marcada manualmente por 19 expertos dentro de la siguiente escala: nulo, bajo, medio y alto y se implementaron varios umbrales de acuerdo. Una vez evaluadas cada una de las palabras se creaba su FPA correspondiente.

Las palabras que forman este diccionario se obtuvieron en primer lugar de la traducción de WordNet-Affect, utilizando Google Traductor, Babylon y English-Spanish Interpreter Pro. Posteriormente se realizó una revisión manual de cada una de las palabras para garantizar la calidad de la traducción.

3.2.4. LIWC

Linguistic Inquiry and Word Count [7] (LIWC) es una potente herramienta de investigación basada en una ciencia sólida. Su funcionamiento se basa en leer los textos que se le proporcionen y contar el porcentaje de palabras que expresen emoción, preocupaciones sociales e incluso partes de la oración.

Fue creado por investigadores interesados en psicología social, de salud, etc., por lo que las categorías del lenguaje fueron creadas para capturar estados psicológicos y sociales de las personas.

LIWC está basado en un conjunto de diccionarios, entre ellos se encuentra un diccionario de palabras sociales (familia, amigos, etc.), signos de puntuación, palabras afectivas (positivas y negativas), etc. Después del proceso de identificar que palabras se encuentran en el texto que se le pasa y también en sus diccionarios, calcula los porcentajes con respecto al total de palabras, para relacionar el texto con una categoría.

[7] http://liwc.wpengine.com/how-it-works/

Capítulo 4: Estudio de técnicas para la clasificación de textos según género del autor

En este capítulo se pretende explicar las diferentes técnicas que existen y se han utilizado anteriormente para realizar la tarea de clasificar los textos según el género del autor.

Todos los trabajos cuyo objetivo es el Author Profiling, y en especial identificación del género necesitan de un corpus de documentos para clasificar. Estos corpus pueden obtenerse de diversos dominios:

- Los hay que utilizan un corpus de documentos formales, como extraídos del British National Corpus (BNC), blog de opinión del NY Times, etc.
- También se utilizan corpus de un dominio menos formal, como por ejemplo aquellos formados por correos electrónicos.
- Por último, no pueden faltar los corpus extraídos de la enorme cantidad de información que proporcionan el emerger de las redes sociales, en un ámbito claramente informal en la mayoría de los casos, más coloquiales y menos estructurados. Por ejemplo blogs, posts, Twitter, Facebook, etc.

La tarea de AP se puede realizar mediante tres enfoques:

1. Machine learning supervisada o no supervisada: la diferencia principal entre ambas es que la forma supervisada contiene una parte de entrenamiento en la que el clasificador intenta aprender a partir de un conjunto de documentos que ya están etiquetados, para posteriormente realizar el test, y otra parte de test en la que se ayuda del entrenamiento realizado para etiquetar los textos, mientras que la forma no supervisada no tiene esa parte de entrenamiento ya que no posee un corpus etiquetado. En cuanto al método supervisado que es el elegido para utilizar en el desarrollo del proyecto destacar que, como ya se ha dicho anteriormente precisa tener un conjunto de documentos etiquetados para de esta forma poder crear un modelo capaz de clasificar nuevos

documentos. Este conjunto de documentos etiquetado debe ser representados mediante un conjunto de características, con el cuál se genera un modelo estadístico que sirve para modelar la distribución de los datos, de forma que al tener un documento sin etiquetar pero representado de la misma forma que el conjunto de entrenamiento, el algoritmo será capaz de clasificarlo obteniendo la distribución de datos a la que más se ajusta.

2. Basada en lexicón: consiste en que a partir de un conjunto de documentos no etiquetados y uno o varios lexicones (lista de palabras representativas de algún grupo social, tema, etc.), se clasifican los documentos. El proceso para decidir a qué grupo, o en el caso de nuestro trabajo a que género pertenece el documento se basa en tomar el texto como una bolsa de palabras o vectores de palabras y buscar coincidencias entre las palabras que contienen esos textos y las que contienen los lexicones utilizados. Por ejemplo, si se tienen dos lexicones uno de palabras masculinas y otro de palabras femeninas, para cada documento, buscamos las palabras que contiene y que se encuentren en los lexicones y en función del número de palabras que coincidan de cada uno de ellos se elegirá una categoría u otra.
3. Híbrida: es un enfoque en el que se combinan los dos anteriores.

Gran parte de los trabajos de AP se definen como un trabajo de machine learning supervisado. Debido a esto es por lo que se ha elegido SVM (algoritmo supervisado) para la realización del TFG. A continuación se explican algunas formas en que se ha planteado esta tarea en estudios anteriores.

Una de las más exploradas en trabajos ya realizados anteriormente es basándose en la extracción de características, que se obtienen de los diferentes textos que se quieren clasificar y a partir de ellos poder clasificar los documentos. Algunas de las características que más se extraen son las siguientes:

- Palabras más frecuentes o que más aparecen en un conjunto de documentos.
- N-gramas: se trata de una subsecuencia de n palabras que podemos encontrar dentro de una secuencia. Existen los unigramas, bigramas, trigramas, etc.

- Palabras de función: son aquellas palabras que no tienen un significado propio pero que ayudan a construir el discurso mediante la relación de otras palabras. Son palabras de función las preposiciones, pronombres y verbos auxiliares.
- Part-of-Speech: es una categoría de palabras que tienen propiedades gramaticales similares.
- Hiperenlaces, o también llamados link, es un elemento propio de la web que hace referencia a otro elemento o recurso.
- Bolsas de palabras: consiste en tomar los documentos como un conjunto de palabras sin orden ni relaciones entre ellos.
- Emoticonos: conjunto de signos de puntuación, o más actualmente pequeños iconos que representan alguna emoción. (Ej. ☺, ☹,etc.)
- Basadas en caracteres: frecuencia de letras mayúsculas, de signos de puntuación, número de caracteres, etc.
- Basadas en palabras: número de palabras por texto, número de stopwords, etc.
- Basadas en oraciones: número de oraciones, número de palabras por oración, etc.
- Basadas en diccionarios: se utilizan diferentes lexicones para obtener características, por ejemplo mediante la frecuencia de las palabras de esos lexicones en los textos.
- Basadas en dependencias sintácticas: capturan dependencias sintácticas del tipo complemento directo, sujeto, modificadores, etc. y determinan la distancia media entre esas dependencias.
- Basada en TF-IDF: frecuencia de término-frecuencia inversa de documento. Está formada por TF (Término-Frecuencia), que determina la frecuencia de aparición de cada término en un documento e IDF (Frecuencia Inversa por Documento), que asigna una mayor importancia a los términos que aparezcan en pocos documentos y una menor importancia a aquellos que aparezcan en gran cantidad de ellos. IDF proporciona la importancia de cada término con respecto a todo el corpus. TF-IDF es una técnica que nos permite representar el contenido del texto como características. Nos permite saber qué importancia tiene cada palabra dentro de una colección de documentos. La fórmula para calcular el TF-IDF es la siguiente:

$$TF - IDF = tf_d * log \frac{N}{n_d}$$

donde tf_d es la frecuencia del término en el documento, N el número total de documentos y n_d el número de documentos donde aparece el término.

Normalmente, en los trabajos de AP basados en la extracción de características se usan un número muy elevado de ellas, fácilmente superior a 1000 y muy rara vez por debajo de 500. Sin embargo, existe un trabajo (Company, J. S., & Wanner, L., 2007*)* en el que su objetivo es demostrar que también se pueden obtener buenos resultados utilizando un número de características muy reducido, en concreto 83 (obtienen un 82.72% de accuracy usando todas las características disponibles). Para ello utilizan características orientadas a la estructura en lugar de al contenido.

Una vez se han extraído las características, se pueden aplicar diferentes algoritmos para la clasificación de los textos basándose en esas características, algunos de ellos son:

- Árboles de decisión.
- Support Vector Machines (SVM), que ha sido el algoritmo elegido para la realización del trabajo ya que es uno de los algoritmos más utilizados en tareas de clasificación. SVM es efectivo en espacios de alta dimensión y utiliza un subconjunto de puntos de entrenamiento en la función de decisión, por lo que desde el punto de vista de memoria es bastante eficiente. Además se puede utilizar con diferentes núcleos como por ejemplo Lineal, RBF (Radial Basis Function), polinomial, etc. Los núcleos utilizados en el desarrollo del proyecto han sido el Linear y RBF (Gaussian). El kernel Linear es el más básico para utilizar con SVM, y su funcionamiento consiste en que el límite de decisión es una línea recta. Los parámetros que más se suelen modificar a la hora de usar este nucleo son C (explicado a continuación) y random_state que es la semilla del generador de números pseudoaleatorios para usar al mezclar los datos. Por otro lado el núcleo RBF traza un límite más complejo basado en la siguiente fórmula:

$$K(x, x') = \exp(-\frac{||x - x'||}{2\sigma^2})^2$$

donde $||x-x'||^2$ es la distancia euclidiana al cuadrado entre dos punto de datos x y x'. El kernel RBF tiene dos parámetros que se pueden modificar. Estos son Gama y C. Gama se considera la región de decisión, si es bajo la curva límite es muy baja y por tanto la región de decisión muy amplia, sin embargo, si Gama es alto, la curva del límite es alta por lo que se crean islas de límites de decisión alrededor de los puntos de datos. Por otro lado el parámetro C es el parámetro de penalización, conforme mayor sea este parámetro más intolerante a los puntos clasificados erróneamente será, lo que hará que el límite de decisión sea más severo.

- SMO, que es una variante de SVM.
- Naïve Bayes.
- Algoritmos de regresión.

Existen también ciertos trabajos que utilizan un modelo supervisado junto con uno no supervisado, a estos sistemas se les llama metaclasificadores. Un trabajo realizado usando esta técnica se basa en realizar un proceso con el objetivo de encontrar perfiles y subperfiles. Este método tiene varias etapas:

- La primera etapa consiste en representar los términos en el espacio de perfiles: se busca las relaciones entre cada término (n-grama, signo de puntuación…) con cada uno de los posibles perfiles.
- La segunda se trata de representar los documentos en el espacio de perfiles: al igual que la anterior etapa, se buscan las relaciones entre cada documento y los diferentes perfiles.
- En la tercera fase se obtienen subperfiles mediante un proceso no supervisado de agrupamiento: este proceso se lleva a cabo debido a que los documentos que pertenecen a un perfil pueden ser demasiado diferentes entre ellos, por ello se buscan diferentes subperfiles heterogéneos dentro de un mismo perfil. Se aplica un agrupamiento a un conjunto de documentos de entrenamiento

etiquetados con el perfil al que pertenecen y cada uno de los grupos resultantes se toma como un subperfil.

- Por último, términos y documentos se representan en el nuevo espacio de subperfiles, realizando el mismo proceso que en la primera y segunda etapa.

Capítulo 5: Desarrollo del proyecto

En este capítulo se va a explicar el desarrollo del software creado para el proyecto. En él se describe el problema a resolver, los objetivos del sistema, la metodología que se utiliza, las historias de usuario del proyecto, la solución propuesta, junto con su justificación y descripción y por último las herramientas utilizadas.

5.1. Descripción del problema

Anteriormente se ha presentado el propósito y los objetivos de este proyecto. A continuación se procede a explicar la descripción del problema que se pretende solucionar con este trabajo.

Actualmente la cantidad de información que se genera a través de Internet y que esta visible para todo el mundo ha crecido de manera exponencial en los últimos años. Prácticamente todo el mundo genera información en la red ya sea mediante un blog, redes sociales, foros, microblogs, etc.

Analizar esta ingente cantidad de información es de gran utilidad, ya que proporciona múltiples beneficios para diferentes sectores. Uno de esos beneficios sería el que pretende darle el Author Profiling, que no es otro que obtener características de los autores de la información publicada en la red.

En la actualidad, AP es una tarea puntera en el ámbito del PLN debido a la gran cantidad de áreas a las que puede ayudar: inteligencia de comercio, crímenes, delitos informáticos, etc.

Sin embargo, la realización de esta tarea en español no esta tan investigada como en inglés, por ello, se va a desarrollar este proyecto en el que se generará una herramienta que podrá ser utilizada para clasificar textos en español en función del género del autor (masculino o femenino).

5.2. Objetivos del sistema

El objetivo principal del sistema es crear una herramienta que permita extraer características de un conjunto de textos etiquetados con el género del autor, para posteriormente realizar una clasificación de parte de ellos (conjunto de test), además de mostrar los datos de acierto resultantes.

Este objetivo se puede descomponer en:

- **Objetivo 1**: partición del conjunto de documentos. El total de documentos se dividirá en dos grupos, 60% serán utilizados para el entrenamiento del sistema y el 40% restante será utilizado para el test del sistema.
- **Objetivo 2:** extracción de características. Para cada documento, extraer el texto y buscar ciertas características dentro de él.
- **Objetivo 3**: extracción de características TF-IDF de los documentos y clasificación de los textos. Cada uno de los documentos de los que se extraen características será etiquetado como masculino o femenino. Para la clasificación se pueden utilizar todas las características, o únicamente elegir parte de ellas.

5.3. Metodología para el desarrollo de software y método a seguir

5.3.1. Metodología

La metodología para el desarrollo de software en el marco de la ingeniería del software es un proceso que se sigue a la hora de diseñar una solución con el fin de estructurar, planificar y controlar el desarrollo de la misma.

Existen diversos tipos de metodologías, pero las dos más utilizadas son la tradicional y la ágil.

La metodología tradicional se caracteriza por:

- Se basa en un ciclo de vida de desarrollo de software en cascada, en la que las etapas se realizan de manera secuencial.
- No se puede hacer una tarea si no se ha completado al anterior.

- Una vez finalizada una etapa no puede modificarse.
- Cada etapa solo se ejecuta una vez.

Estas características hacen que esta metodología tenga algunos inconvenientes:

- Si una vez finalizada una tarea nos damos cuenta de que hay algo mal hecho, habría problemas para volver atrás a modificarlo.
- El usuario no ve el producto hasta la finalización del proyecto, lo que provoca que pueda haber cosas que no quiera o que estén mal con respecto a sus requisitos.

En la Ilustración 5.1, que se muestra a continuación, se puede ver el ciclo de vida de una metodología tradicional:

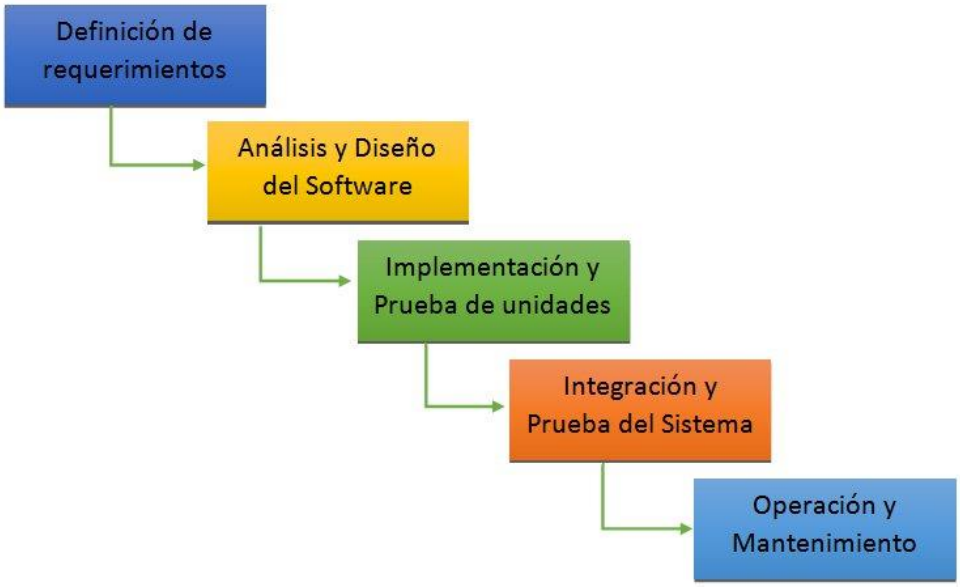


Ilustración 5.1. Ciclo de vida de la metodología tradicional.

Por otro lado se encuentra la metodología ágil, caracterizada por:

- Se basa en un ciclo de vida de desarrollo del software iterativo e incremental, en la que el desarrollo del proyecto se divide en diversos bloques temporales que se denominan iteraciones.

- Las etapas de cada ciclo se repiten y se van añadiendo funcionalidad al sistema. Al finalizar cada etapa se muestra al cliente para que vaya observando el desarrollo del producto.
- Cada iteración es corta, de 2 a 4 semanas.
- Se hacen entregas parciales al finalizar cada iteración para que el usuario vaya validando que se cumplen los requisitos. Estas entregas son resultados usables.
- Se pueden solapar varias etapas.
- Los requisitos pueden ir cambiando durante el desarrollo del sistema.

En la siguiente imagen, Ilustración 5.2, se muestra el ciclo de vida de una metodología ágil:

Ilustración 5.2. Ciclo de vida de la metodología ágil.

Una vez se han examinado ambas metodologías se ha decidido utilizar una metodología ágil basada en un ciclo de vida iterativo e incremental, principalmente por los siguientes motivos:

- Nuestro proyecto está formado por diversas iteraciones y tendrá diferentes entregas parciales.
- Al realizar la entrega de cada una de las entregas pueden surgir cambios debidos al mal planteamiento de requisitos en un principio, y mediante esta metodología no habrá problemas en realizarlos.
- Al finalizar cada iteración la tutora del proyecto, que tendrá el rol de cliente, podrá ver una parte usable del proyecto para ir verificando el cumplimiento de los requisitos.

5.3.2. Método ágil

De entre los diferentes métodos ágiles existentes, algunos de ellos se muestran en la Ilustración 5.3, se ha elegido como método ágil a utilizar **SCRUM.**

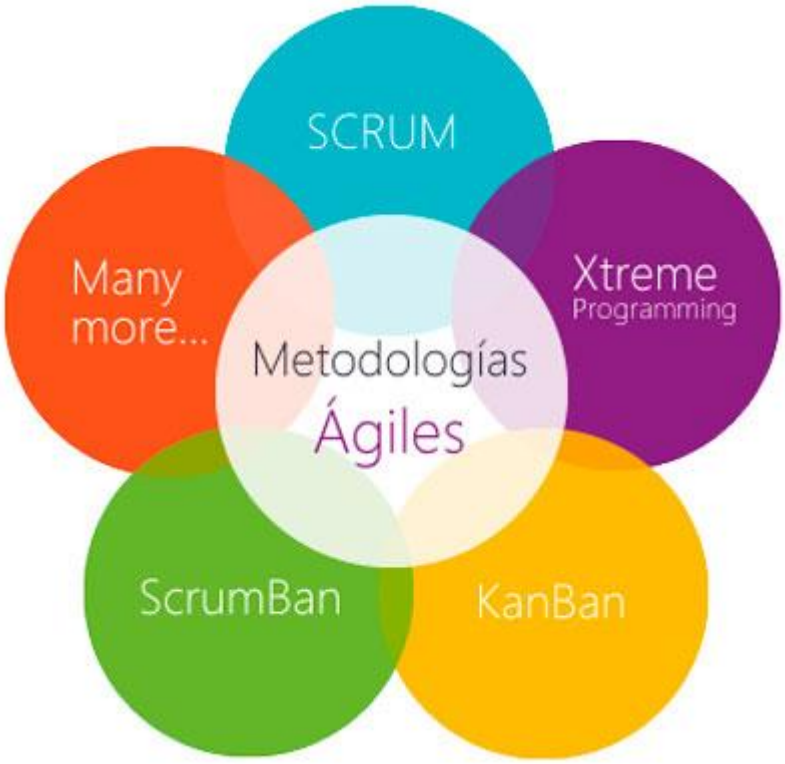


Ilustración 5.3. Métodos ágiles.

SCRUM hace que un proyecto se ejecute en bloques con un determinado tiempo (suelen ser de 2 o 4 semanas aunque puede variar), que se denominan iteraciones o sprints.

Cada una de los sprints, cuyo ciclo se representa en la Ilustración 5.4, tiene la obligación de presentar un resultado completo, y además deberá de tener un incremento en el producto respecto al entregado en el sprint anterior.

Cada sprint está delimitado por una reunión de planificación del sprint y una reunión retrospectiva. Se realizarán diversas reuniones de seguimiento breve, al tratarse de un TFG, las reuniones no podrán ser diarias, por lo que serán algo más flexibles. Al final del sprint se entrega el producto al cliente.

Ilustración 5.4. Ciclo de un sprint.

Las actividades de la metodología Scrum son las siguientes:

- Planificación del proyecto.
- Selección de requisitos.
- Planificación de la iteración. Se divide en dos partes:
  - En la primera el cliente muestra al equipo su lista de requisitos ordenada por prioridad y pone una meta para la iteración. El equipo la examina y pregunta en caso de que hubiera alguna duda.
  - En la segundase planifica la iteración: se definen las tareas, estimación de tiempos y asignación de tareas.
- Ejecución de la iteración (sprint).
- Demostración de requisitos completados: reunión con el cliente para presentarle los requisitos que se han completado en la iteración y el desarrollo producido respecto a iteraciones anteriores.
- Retrospectiva: por último se analiza cómo han trabajado durante la iteración, cosas bien hechas, cosas que cambiar…

5.4. Historias de usuario

Una de las principales partes de la metodología ágil son las historias de usuario. A partir de ellas se obtienen los requisitos que forman un proyecto software.

Una historia de usuario es una especificación de una parte del software, describe la funcionalidad que podrá ser útil para el usuario del proyecto. A partir de ellas se puede empezar a trabajar, y pueden ser modificadas durante el desarrollo y adaptarse a cambios en los requisitos.

Para la creación de una historia de usuario se necesita la comunicación y acuerdo entre equipo y cliente. Se centrar en los aspectos más relevantes y son de pequeño tamaño.

Una historia de usuario puede estar formada por:

- ID: identificador de la historia de usuario.
- Título: título de la historia de usuario. Debe ser lo más claro posible.
- Prioridad de la historia de usuario respecto al resto de historias de usuario. Puede tomar los siguientes valores: baja, media, alta, crítica, aunque nosotros no utilizaremos en nuestro proyecto esta última.
- Descripción: descripción breve de la finalidad de la historia de usuario.
- Dependencias: en el caso en que una historia de usuario dependa de otra.
- Criterios de aceptación: son las pruebas que se le realizan a una historia de usuario para comprobar que se ha desarrollado correctamente.

A continuación (Tabla 5.1.), se muestra la estructura que van a tener las historias de usuario.

ID	TÍTULO
DESCRIPCIÓN	
PRIORIDAD	DEPENDENCIAS

Tabla 5.1. Plantilla historias de usuario

A continuación se muestran las distintas historias de usuario del proyecto a realizar:

1	ELECCIÓN DE CORPUS
Como usuario quiero disponer de un corpus de documentos en español etiquetados por el género del autor	
Alta	-

Tabla 5.2. Historias de usuario: "Elección de corpus"

2	DIVISIÓN DEL CORPUS
Como usuario quiero el conjunto de documentos dividido en dos partes: 60% de documentos para el entrenamiento y 40% de documentos para el test con el objetivo de realizar la clasificación y evaluación	
Media	1

Tabla 5.3. Historias de usuario: "División del corpus"

3	EXTRAER CARACTERÍSTICAS
Como usuario quiero poder extraer diversas características de cada documento	
Alta	**1**

Tabla 5.4. Historias de usuario: "Extraer características"

4	CLASIFICACIÓN DOCUMENTOS SEGÚN GÉNERO
Como usuario quiero clasificar los distintos documentos en función del género de su autor (masculino y femenino)	
Alta	**1,3**

Tabla 5.5. Historias de usuario: "Clasificación documentos según género del autor"

5	ELEGIR CARACTERÍSTICAS
Como usuario quiero poder elegir qué características utilizar a la hora de clasificar los documentos.	
Baja	**-**

Tabla 5.6. Historias de usuario: "Elegir características"

6	EVALUACIÓN DEL SISTEMA
Como usuario quiero poder evaluar el sistema de clasificación de género en función de las características elegidas	
Media	**-**

Tabla 5.7. Historias de usuario: "Evaluación del sistema"

5.5. Propuesta de solución

A continuación se propone la solución elegida para este TFG. Consiste en desarrollar una herramienta para la clasificación de documentos en función del sexo del autor. Para ello se necesita desarrollar los siguientes módulos:

- Sistema para la división del corpus, formando dos corpus: corpus de entrenamiento del sistema (60% del total de documentos) y corpus de test (40% del total de documentos).
- Sistema para la extracción de características de los distintos documentos de los corpus.
- Sistema para la clasificación de textos, que etiquete cada documento en función del género del autor (masculino o femenino).

5.6. Descripción de la solución

En este apartado se explica el desarrollo de la solución elegida mediante iteraciones. Cada iteración, representará una historia de usuario de las mencionadas anteriormente, por tanto habrá un total de 5 iteraciones.

5.6.1. Iteración 1

Iteración 1: "Como usuario quiero disponer de un corpus de documentos en español etiquetados por el género del autor"

La iteración 1 se corresponde con la historia de usuario "Elección de corpus", que se muestra en la Tabla 5.2.

Para la elección del corpus para proporcionarlo al usuario, se han estudiado diferentes corpus de documentos, todos ellos en español (ya que el TFG está enfocado en este idiomas por lo ya explicado en el capítulo 1 punto 1.2.) y etiquetados como mínimo con el género del autor de cada documento. Todos ellos han sido explicados en el apartado 2.

A continuación se explica más a fondo el corpus elegido finalmente, que ha sido el SpanText:

Como ya se dijo, este corpus fue creado por María Paula Villegas, María José Garciarena Ucelay, Marcelo Luis Erreclade y Leticia Cecilia Cagnina, pertenecientes al Laboratorio de Investigación y Desarrollo en Inteligencia Computacional de la Facultad de Ciencias Físico, Matemáticas y Naturales de la Universdidad Nacional de San Luis, Argentina.

Se accedió a él mediante contacto por email con uno de sus creadores.

Este corpus se creó a partir del proporcionado para la competición del PAN de 2013[8], el cual contenía mucho ruido por lo que los resultados no fueron lo suficientemente satisfactorios.

SpanText es una colección de textos formales en español, etiquetados con la edad y el género del autor (se utilizará únicamente el género). Los documentos fueron recuperados de la web y escritos por diferentes autores. Además fueron etiquetados manualmente.

Se define como formal debido a que contiene un porcentaje muy pequeño de palabras que no existen como tal, abreviaciones, contracciones, emoticonos, etc. Se basan principalmente en periódicos, blogs de confianza, reportajes de estudiantes, libros, etc. Todos los documentos tienen la codificación UTF-8.

Las principales características del corpus además de las brevemente mencionadas son:

[8] https://pan.webis.de/clef13/pan13-web/index.html

- Está formado por 1000 documentos en español.
- Los documentos hablan sobre diversos temas.
- Cada texto tiene como mínimo 150 palabras.
- Los textos que forman el corpus han sido escritos por personas de habla española procedentes de España y América Latina.

Este recurso proporciona dos versiones del corpus:

- **Balanceada**: tiene el mismo número de documentos masculinos y femeninos.
- **No balanceada**: no tiene el mismo número de documentos masculinos que femeninos.

En la siguiente tabla (Tabla 5.8), se muestra información acerca de las versiones que se proporcionan para el corpus SpanText:

Versión	Masculinos	Femeninos	Palabras	Oraciones
Balanceada	500	500	792.567	36.116
No balanceada	700	300	814.031	36.781

Tabla 5.8. Distribución de los documentos y características de ellos para las dos versiones del corpus

Cada una de estas versiones está dividida en 6 clases:

-10sMasc - 10sFem

-20sMasc - 20sFem

-30sMasc - 30sFem

Como se puede observar para cada género existen 3 clases, cada una de ellas engloba personas de un rango de edad determinado, de manera que los documentos de las clases *10s* son escritos por autores cuya edad se encuentra entre 13 y 17 años, los de las clases *20s* por autores con edad entre 23 y 27 y los de las clases *30s* con edad entre 33 y 47 aproximadamente.

Teniendo en cuenta todas estas características citadas, las principales por las que se decidió utilizar este corpus son:

- Los textos están etiquetados con el género del autor. Extraer características y clasificarlos es el objetivo de este trabajo.
- Contiene los textos divididos por edades y sexo, aunque se unirán en dos clases únicamente: masculino y femenino.
- Los documentos tienen un porcentaje muy pequeño de ruido, y son formales, lo que ayuda a realizar el trabajo.

Además se ha utilizado la versión balanceada del corpus, que contiene el mismo número de textos masculinos y femeninos, ya que ayuda a obtener mejores resultados ya que la clasificación no se verá afectada por la clase mayoritaria, como si pasaría en caso de utilizar la versión no balanceada.

Se creará una partición del corpus (explicada en la siguiente iteración (Iteración 2)), de manera que se utilizará:

- **60%** de los documentos para el entrenamiento del sistema (600 documentos), de los que 300 son escritos por hombres y 300 por mujeres.
- **40%** de los documentos para el test del sistema (400 documentos), de los que 200 son escritos por hombres y 200 por mujeres.

5.6.2. Iteración 2

Iteración 2: “Como usuario quiero el conjunto de documentos dividido en dos partes: 60% de documentos para el entrenamiento y 40% de documentos para el test con el objetivo de realizar la clasificación y evaluación”

La iteración 2 se corresponde con la historia de usuario “División del corpus”, que se muestra en la Tabla 5.3.

El corpus elegido ha sido el SpanText, cuyas características han sido explicadas en la iteración anterior (Iteración 1).

Para dividir el conjunto de documentos que contiene el corpus en dos partes se ha desarrollado un script en el que se han recorrido todos los directorios que forman el corpus en su versión balanceada (Ilustración 5.5).

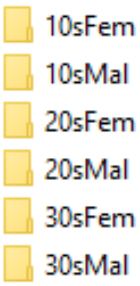


Ilustración 5.5. Directorios que forman SpanText

Cada una de estas carpetas tiene un total de 166 o 167 documentos, con un número total de documentos de 1000, de los cuales 500 están etiquetados como femeninos y los otros 500 como masculinos. Estos documentos se quieren dividir en dos partes:

- 60% de los documentos formarán el conjunto de entrenamiento.
- 40% de los documentos formarán el conjunto de evaluación o test.

Todos los directorios se dividen en dos partes (entrenamiento y test) en función de los porcentajes que se quieren obtener para cada conjunto (60-40), de manera que finalmente cada uno de los dos conjuntos creados tendrá el mismo número de documentos etiquetados como femeninos y masculinos, tal y como se muestra en la siguiente tabla (Tabla 5.9.).

Género/Conjunto	Entrenamiento	Test	Total
Masculino	300	200	500
Femenino	300	200	500
Total	600	400	1000

Tabla 5.9. Tamaño de los conjuntos del corpus

Ambos conjuntos (entrenamiento y test) se guardan en ficheros con formato CSV, en los que cada línea representará un documento y tendrá la siguiente estructura:

"ID";"Género";"Edad";"Texto"

En la siguiente ilustración Ilustración 5.6 se puede observar un ejemplo de parte del corpus con el formato especificado:

```
"3";"Femenino";"10";"En una gran fantasía, en un palacio de hielo, en él una bella muchacha,..."
"4";"Femenino";"10";"El primer día de la primavera llena de flores, aromas y colores, un ..."
"5";"Femenino";"10";"La tan esperada noche de brujas por fin había llegado, los niños se ..."
```

Ilustración 5.6. Ejemplo formato corpus.

- **ID**: identificador de cada documento.
- **Género**: Masculino o Femenino en función del autor del documento.
- **Edad**: Edad del autor del documento, existen tres grupos: 10, 20, 30.
- **Texto**: el texto que contiene el documento y que será analizado.

Todos los campos van entrecomillados y separados por ';'.

5.6.3. Iteración 3

Iteración 3: "Como usuario quiero poder extraer diversas características de cada documento"

La iteración 3 se corresponde con la historia de usuario "Extraer características", que se muestra en la Tabla 5.4.

En el desarrollo de esta iteración, se realiza la extracción de características del corpus una vez que está divido en las dos partes que se necesitan.

Se encuentran organizadas en 4 categorías como se muestra en la siguiente tabla (Tabla 5.10.):

Feature Category	#Features
Character-based (C)	3
Word-based (W)	6
Sentence-based (S)	2
Dictionary-based (D)	3

Tabla 5.10. Categorías de las características

- Character- based: son aquellas características que se basan en caracteres individuales (espacios, signos de puntuación, letras y números).
- Word-based: son aquellas basadas en palabras completas.
- Sentence-based: son las características que afectan a oraciones completas y su contenido.
- Dictionary-based: son las características que están formadas gracias al manejo de diccionarios o lexicones (lexicones de palabras con polaridad, lexicones de palabras de hombres o mujeres, etc).

En total se han extraído 14 características, que se enumeran a continuación. Las 9 primeras características han sido utilizadas en el artículo desarrollado por Soler, J., & Wanner, L. (2014)

1. Número de caracteres por texto (Character-Based (C)).
2. Número de palabras por texto (Word-Based (W)).
3. Frecuencia de signos de puntuación en función del número de caracteres por texto (C).
4. Frecuencia de letras mayúsculas en función del número de palabras por texto (C).
5. Número de palabras diferentes por texto (W).
6. Frecuencia de “stopwords” o palabras vacías en función del número de palabras por texto (W).

7. Número de oraciones por texto (Sentence-Based(S)).
8. Número medio de palabras por oración (S).
9. Frecuencia de palabras con polaridad en función del número de palabras por texto (Dictionary-Based (D)): las mujeres son más afectivas que los hombres y por ello, utilizan más palabras que tengan polaridad, por lo tanto está es una característica que diferencia ambos géneros.

Peenebaker et al. (2003) han investigado acerca de las diferencias entre hombres y mujeres a la hora de expresarse. Algunas de las principales diferencias que han extraído, mujeres utilizan más los pronombres y hombres utilizan más los determinantes y las preposiciones, se han utilizado como característica en este TFG.

10. Frecuencia de preposiciones en función del número de palabras por texto (W).
11. Frecuencia de pronombres en función del número de palabras por texto (W)
12. Frecuencia de determinantes en función del número de palabras por texto (W).

Por último se ha pensado utilizar dos lexicones con palabras propias en el vocabulario de mujeres que no lo sean en el de los hombres y viceversa:

13. Frecuencia de palabras propias de los hombres en función del número de palabras por texto basándonos en el lexicón de palabras identificativas del género masculino creado (D).
14. Frecuencia de palabras propias de las mujeres en función del número de palabras por texto basándonos en el lexicón de palabras identificativas del género femenino creado (D).

Al texto de cada uno de los documentos se le aplica un preprocesamiento que ayuda a obtener las características de una manera más efectiva. Este proceso está formado por los siguientes pasos:

1. Realizar una pequeña limpieza, ya que el texto de algunos documentos comienza con un signo de interrogación ‘?’, el cual se elimina ya que no es significativo.

2. Realizar una tokenización, que consiste en separar el texto de un documento, ya sea en oraciones o en palabras a partir de tokens. En el caso de este proyecto se aplican ambos. En primer lugar se realiza una tokenización de oraciones, se divide el texto en oraciones utilizando el tokenizador de oraciones de NLTK[9].
3. Posteriormente se aplica la tokenización a nivel de palabra. En este caso se aplica un tokenizador basado en una expresión regular, '\w+', a través de la cual se obtienen simplemente las palabras (conjunto de 1 o más caracteres alfanuméricos).
4. El último proceso de preprocesamiento que se aplica, consiste en sustituir todas las letras mayúsculas por letras minúsculas. Este proceso se realiza ya que los lexicones que se van a utilizar se encuentran escritos en minúscula, de forma que por ejemplo en el texto estuviera la palabra "Trabajo" y en el lexicón estuviera "trabajo", no se reconocería. La segunda razón por la que se realiza este proceso es para normalizar todo el texto ya que, utilizando el mismo ejemplo que antes, "Trabajo" es la misma palabra que "trabajo", pero al no estar escritas igual no se reconocerían como la misma palabra.

Una vez se tiene el texto dividido en oraciones, se obtienen las primeras características: se cuenta el número de caracteres del texto (letras, números, signos de puntuación y espacios), se calcula la frecuencia de signos de puntuación respecto al total de caracteres por texto, para lo cual se han tenido en cuenta los signos de puntuación que se muestran en la Ilustración 5.7, y la frecuencia de letras mayúsculas en función del número total de palabras por texto.

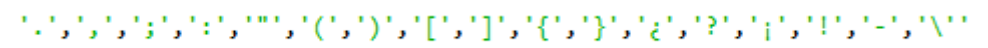


Ilustración 5.7. Signos de puntuación considerados

Cuando el preprocesamiento se ha aplicado en su totalidad, se extraen el resto de características: número de palabras por texto, número de palabras diferentes por texto, frecuencia de stopwords, de pronombres (más utilizados por mujeres), de

[9] https://www.nltk.org/api/nltk.tokenize.html

preposiciones (más utilizados por hombres), de determinantes (más utilizados por hombres), número de oraciones por texto, número medio de palabras por oración y las basadas en lexicones.

Para extraer las características léxicas se han utilizado dos lexicones creados manualmente, uno de palabras masculinas y otro de femeninas y también se ha utilizado iSOL, todos ellos explicados en el capítulo 3 en profundidad. El proceso a seguir consiste en contar el número de palabras que existen en el texto y también se encuentran presentes en el lexicón, así se obtienen 3 características, una para representar la frecuencia de palabras típicas de hombres, otra para la frecuencia de palabras típicas de mujeres y la última para representar la frecuencia de palabras con polaridad (positiva, negativa).

Todas las características que se han extraído se han guardado en un fichero con formato CSV, en el que cada línea representa las características extraídas para cada documento. Las características de cada línea se encuentran separadas entre ellas por ';' y todas ellas van entrecomilladas, tal y como se muestra en la ilustración 5.8.

```
"7598","1348","0.03224532771782048","0.12388724035608309","562","0.5126112759643917",..."11"
"1962","299","0.04485219164118247","0.2508361204013378","183","0.41471571906354515",..."18"
"5179","900","0.01139216064877389 4","0.08222222222222222","361","0.4944444444444446",..."36"
```

Ilustración 5.8. Ejemplo formato fichero características.

5.6.4. Iteración 4

Iteración 4: "Como usuario quiero clasificar los distintos documentos en función del género de su autor (masculino y femenino)"

La iteración 4 se corresponde con la historia de usuario "Clasificación documentos según género", que se muestra en la Tabla 5.5.

En esta iteración, se lleva a cabo el desarrollo del sistema de clasificación de documentos en función del género del autor. Para el desarrollo del clasificador se ha utilizado un método supervisado de clasificación basado en Máquinas de vectores de soporte (Support Vector Machine (SVM)). Para ello, se ha optado por la herramienta 'scikit-learn' que proporciona Python.

El sistema desarrollado proporciona la opción de poder elegir que núcleo utilizar de los dos que se proponen: RFB, es el núcleo por defecto o Linear. Estos núcleos se ejecutan con los valores por defecto para los parámetros explicados en el punto 4.

Para la clasificación se han utilizado las características extraídas de los documentos, explicadas en la iteración anterior (Iteración 3) y la representación TF-IDF. Además en esta iteración se hace útil la división del corpus realizada en la Iteración 2 (conjunto de entrenamiento y conjunto de test). Para aplicar el clasificador se han seguido varios pasos:

- Aplicar TF-IDF al conjunto de documentos ya divididos en dos grupos: mediante este paso se obtiene una representación del texto contenido en el documento que puede ser utilizada como características.
- Unir las características obtenidas al aplicar TF-IDF junto con las extraídas en la iteración anterior para cada documento.
- Aplicar el clasificador (con núcleo RBF o Linear): primero se entrena con el modelo SVM el conjunto de características de cada documento del conjunto de entrenamiento junto con las etiquetas de cada uno de ellos. Tras el entrenamiento se aplica la predicción de la clasificación al conjunto de documento de test y se obtienen las etiquetas para cada uno de los documentos.

Por último se muestran los resultados obtenidos tras la clasificación

5.6.5. Iteración 5

Iteración 5: “Como usuario quiero poder elegir qué características utilizar a la hora de clasificar los documentos.”

La iteración 5 se corresponde con la historia de usuario “Elegir características”, que se muestra en la Tabla 5.6.

Para que el usuario pueda elegir las características con las que quiere ejecutar el clasificador se ha utilizado una librería de Python llamada *argparse*[10], a través de

[10] https://docs.python.org/3/library/argparse.html

la cual se puede incluir argumentos a la hora de ejecutar nuestro script por consola. Permite definir qué argumentos son obligatorios o qué valores pueden tomar. Esta librería genera automáticamente los mensajes de ayuda, uso y error cuando se inserta algún argumento inválido.

Para cumplir esta iteración, se añadió un argumento a la hora de ejecutar el script en el que se pueden elegir qué características utilizar, ya que a cada una de ellas se le asignó un número. El listado de posibles características que se pueden elegir y el comando que se tiene que utilizar se muestra en la ayuda del programa, que se puede ver ejecutando 'python <nombreScriptClasificador> --help' o 'python <nombreScriptClasificador –h' tal y como se puede observar en la Ilustración 5.9.

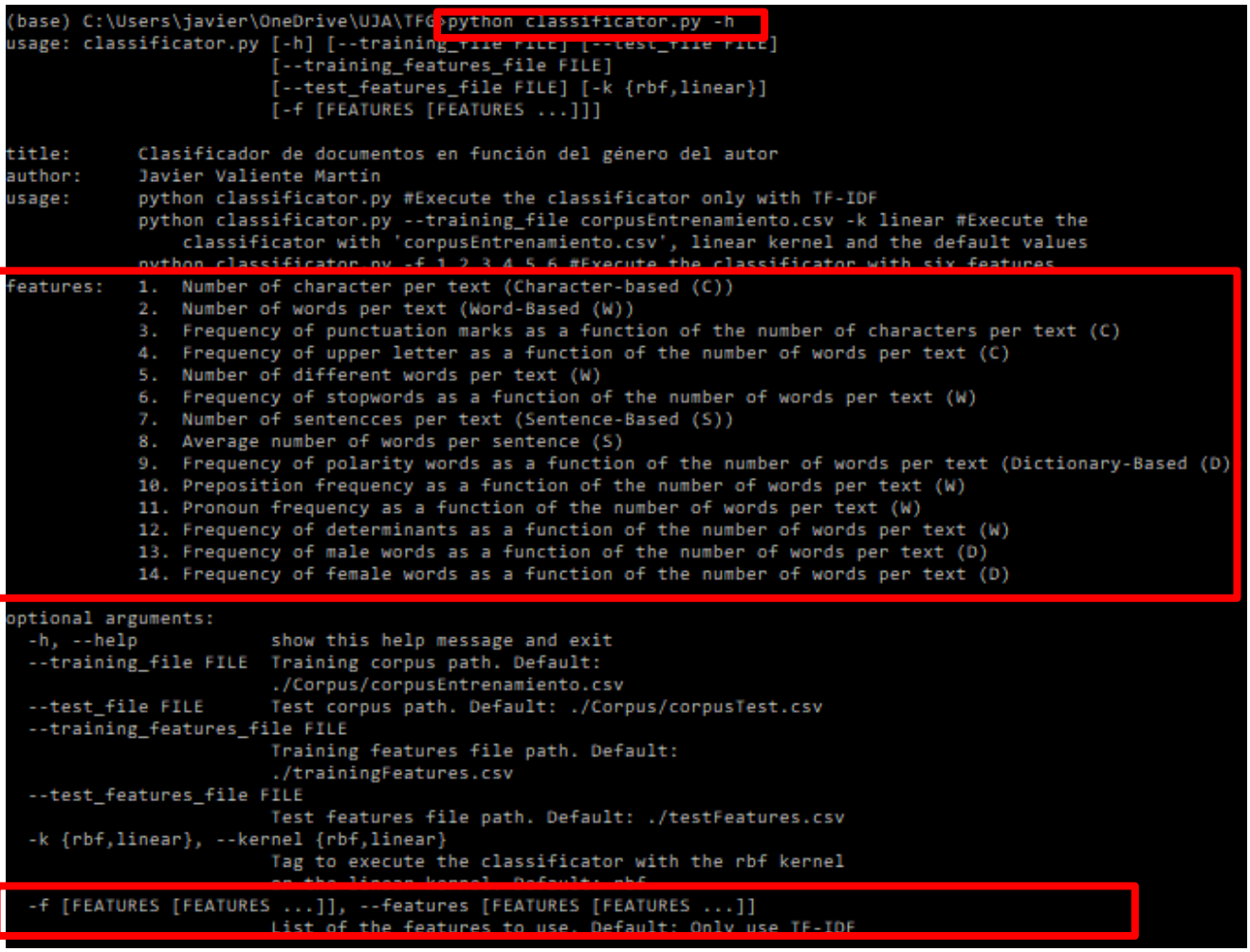


Ilustración 5.9. Ayuda del script del clasificador

Por ejemplo, si se quiere ejecutar el clasificador con las características 1, 2, 3, 4 y 5, el comando utilizado sería: 'python classificator.py –f 1 2 3 4 5'. Por defecto solamente se utilizaría TF-IDF ('python classificator.py –f').

5.6.6. Iteración 6

Iteración 6: "Como usuario quiero poder evaluar el sistema de clasificación de género en función de las características elegidas"

La iteración 6 se corresponde con la historia de usuario "Evaluación del sistema", que se muestra en la Tabla 5.7.

Para la evaluación del sistema se han llevado a cabo una serie de pruebas con diferentes combinaciones de las características que se han extraído y con dos núcleos distintos (RBF y Linear) para el clasificador. Se utilizan las 4 medidas más utilizadas en la clasificación de textos:

- Precisión (P): es el total de textos que han sido correctamente clasificados entre el total de textos que han sido clasificados, para una clase dada.

$$\text{Pmasc} = \frac{VM}{VM + FM} \qquad \text{Pfem} = \frac{VF}{VF + FF} \qquad \textbf{Macro P} = \frac{\boldsymbol{Pmasc + Pfem}}{\boldsymbol{2}}$$

- Recall (R): también llamado exhaustividad, es el número de textos que son correctamente clasificados sobre el total de textos de esa clase.

$$\text{Rmasc} = \frac{VM}{VM + FF} \qquad \text{Rfem} = \frac{VF}{VF + FM} \qquad \textbf{Macro R} = \frac{\boldsymbol{Rmasc + Rfem}}{\boldsymbol{2}}$$

- F1-score (F1): consiste en la medida de exactitud de la validación. Se considera la precisión y la exhaustividad para calcularlo.

$$\textbf{F1} = \frac{\boldsymbol{2PR}}{\boldsymbol{P + R}}$$

- Accuracy o exactitud: es el resultado más relevante, muestra la proporción de documentos que se han clasificado correctamente.

$$\textbf{Acc} = \frac{\boldsymbol{VM + VF}}{\boldsymbol{VM + VF + FM + FF}}$$

Estas medidas se calculan a partir de la matriz de confusión (Tabla 5.11).

		Estimado por el sistema	
		Masculino	**Femenino**
Realidad	**Masculino**	Verdadero Masculino (VM)	Falso Femenino (FF)
	Femenino	Falso Masculino (FM)	Verdadero Femenino (VF)

Tabla 5.11. Matriz de confusión

- Verdadero Masculino (VM): textos que han sido clasificados como masculinos y que realmente lo son.
- Verdadero Femenino (VF): textos que han sido clasificados como femeninos y que realmente lo son.
- Falso Masculino (FM): textos que han sido clasificados como masculinos y que realmente no lo son.
- Falso Femenino (FF): textos que han sido clasificados como femeninos y que realmente no lo son.

Los resultados obtenidos se muestran a continuación:

- La tabla 5.12. muestra las pruebas utilizando el kernel por defecto (RBF (Gaussian)) en orden ascendente de accuracy:

Feature combination	#Features	Precision	Recall	F1	Accuracy (%)
Sentence-based (S)	2	64	60	58	60.5
S + D	5	64	60	58	60.5
C + D	6	63	62	62	62.5
Character-based (C)	3	63	62	62	62.5
C + W + S + D	14	63	63	63	62.75
C + W	9	63	63	63	63.25
C + W + D	12	63	63	63	63.25
10 + 11 + 12 + 13 +14	5	66	64	62	63.75
C + S	5	64	64	64	63.75
C + S + D	8	64	64	64	63.75
TFIDF only	1	67	64	63	64.25
Dictionary-based (D)	3	68	65	63	64.5
13 + 14	2	68	65	63	64.5
W + 13 + 14	8	72	72	72	72
W + D	9	72	72	72	72

Word-based (W)	6	72	72	72	72
W + S + D	11	73	73	73	73.25
W + S	8	73	73	73	73.25

Tabla 5.12. Resultados con corpus SpanText y núcleo RBF

- La tabla 5.13. muestra las pruebas utilizando el kernel Linear, también en orden ascendente de accuracy:

Feature combination	#Features	Precision	Recall	F1	Accuracy (%)
Character-based (C)	3	57	57	57	57
C + D	6	59	58	58	58.5
C + S	5	59	59	59	59
C + S + D	8	59	59	59	59
W + D	9	65	62	60	61.75
W + S + D	11	66	62	60	62.5
W + S	8	66	63	61	63
W + 13 + 14	8	67	64	62	63.75
C + W	9	67	65	63	64.75
C + W + D	12	67	65	63	64.75
Word-based (W)	6	68	66	65	66
C + W + S + D	14	71	67	66	67.25
TFIDF only	1	68	68	68	68.25
Dictionary-based (D)	3	68	68	68	68.25
13 + 14	2	68	68	68	68.25
10 + 11 + 12 + 13 + 14	5	69	69	69	68.75
Sentence-based (S)	2	70	70	70	69.75
S + D	5	70	70	70	69.75

Tabla 5.13. Resultados con corpus SpanText y núcleo Linear

Como se puede observar en las tablas anteriores los resultados rondan el 65% de accuracy, siendo el mayor porcentaje obtenido utilizando el núcleo RBF y las características basadas en palabras y basadas en oraciones, obteniendo un 73.25%.

Para comprobar la consistencia del sistema desarrollado con otros corpus, se ha evaluado utilizando otro corpus, extraído de la propuesta para el PAN2014. Este pequeño corpus de prueba formado está compuesto por 46 documentos escritos por mujeres y 46 escritos por hombres para el conjunto de entrenamiento y 32 documentos escritos por mujeres y 32 escritos por hombres para el conjunto de test. Utilizando este corpus, junto con las características extraídas del mismo, los resultados son los siguientes:

- La tabla 5.14. muestra las pruebas utilizando el kernel por defecto (RBF (Gaussian)) en orden ascendente de accuracy:

Feature combination	#Features	Precision	Recall	F1	Accuracy (%)
C + W + S + D	14	59	58	56	57.81
C + W	9	60	58	55	57.81
C + W + D	12	60	58	55	57.81
C + D	6	62	59	57	59.375
Character-based (C)	3	62	59	57	59.375
C + S	5	61	59	58	59.375
C + S + D	8	61	59	58	59.375
W + 13 + 14	8	64	64	64	64.06
W + D	9	64	64	64	64.06
Word-based (W)	6	64	64	64	64.06
Sentence-based (S)	2	66	66	65	65.625
S + D	5	66	66	65	65.625
W + S + D	11	77	75	75	75
W + S	8	77	75	75	75
Dictionary-based (D)	3	83	83	83	82.81
10 + 11 + 12 + 13 + 14	5	84	84	84	84.375
TFIDF only	1	84	84	84	84.375
13 + 14	2	84	84	84	84.375

Tabla 5.14. Resultados con corpus 2 y núcleo RBF

- La tabla 5.15. muestra las pruebas utilizando el kernel Linear, también en orden ascendente de accuracy:

Feature combination	#Features	Precision	Recall	F1	Accuracy (%)
C + S + D	8	81	81	81	81.25
W + S + D	11	81	81	81	81.25
W + S	8	81	81	81	81.25
C + W	9	83	83	83	82.81
C + W + D	12	83	83	83	82.81
C + W + S + D	14	83	83	83	82.81
Dictionary-based (D)	3	83	83	83	82.81
Sentence-based (S)	2	83	83	83	82.81
S + D	5	83	83	83	82.81
C + D	6	86	84	84	84.375
C + S	5	84	84	84	84.375
W + D	9	86	84	84	84.375
TFIDF only	1	85	84	84	84.375
13 + 14	2	85	84	84	84.375
10 + 11 + 12 + 13 + 14	5	85	84	84	84.375
Character-based (C)	3	87	86	86	85.94
W + 13 + 14	8	87	86	86	85.94
Word-based (W)	6	87	86	86	85.94

Tabla 5.15. Resultados con corpus 2 y núcleo Linear

Como se puede observar en las tablas anteriores, los resultados con este segundo corpus incluso superan el primer corpus con el que se trabajó el sistema, obteniendo un porcentaje máximo de accuracy del 85.94% utilizando el núcleo Linear y las características basadas en palabras junto con los diccionarios de palabras masculinas y femeninas o las características basadas en caracteres, ya que dan el mismo resultado. Por tanto quedó demostrada la consistencia del sistema con otros corpus.

5.7. Herramientas para la implementación del sistema

5.7.1. Python

El sistema se ha desarrollado con el lenguaje de programación Python (Ilustración 5.10.), en su versión 3.6. Este lenguaje fue creado por un investigador holandés llamado Guido Van que trabajaba en el centro de investigación CWI de Ámsterdam.

Python es un lenguaje de programación interpretado, de alto nivel pero fácil de aprender y de utilizar, además es de código abierto por lo que su uso es gratuito.

Ilustración 5.10. Lenguaje de programación Python

Permite la programación orientada a objetos y programación estructurada. Puede ser utilizado para desarrollo web, acceso a bases de datos, desarrollo software, etc. Debido a su sintaxis simple y su naturaleza interpretada, es ideal para realizar scripting sobre la mayor parte de las plataformas.

Además dispone de una línea de comandos a través de la cual se pueden introducir y ejecutar sentencias, produciendo resultados.

Python puede ser utilizado con una serie de librerías que proporcionan una serie de utilidades, para la creación del sistema se han utilizado las siguientes:

- csv: utilizada para trabajar con ficheros '.csv'.
- re: se usa para trabajar con expresiones regulares.
- xml.etree.ElementTree: librería que permite trabajar con la estructura de ficheros 'XML'.
- NLTK: conjunto de bibliotecas para trabajar con el PLN.
- Sklearn: librería para utilizar *Machine Learning* en Python.
- Numpy: proporciona funciones matemáticas de alto nivel para operar con matrices y vectores.
- Argparse: librería para agregar argumentos a la ejecución por línea de comandos.

Capítulo 6: Conclusiones y trabajos futuros

En este capítulo se describen las conclusiones que se han obtenido una vez finalizado el proyecto y las mejoras que se podrían aplicar a este trabajo en el futuro.

6.1. Conclusiones

En este punto, doy por finalizado el proyecto ya que se han conseguido realizar los objetivos que la tutora del proyecto y yo nos habíamos marcado.

La elección de este TFG se ha llevado a cabo debido a tener un gran interés por el campo del Procesamiento del Lenguaje Natural. Durante la realización del proyecto he cursado la asignatura de PLN que se imparte en el 4º curso del Grado de Ingeniería Informática, lo que me ayudo a introducirme en este ámbito de la informática ya que era el primer contacto que tenía con él y a motivarme aún más a la realización del proyecto ya que encontré una gran utilidad debido a la multitud de aplicaciones que se pueden crear y problemas que se pueden solucionar gracias al PLN.

Al obtener más información sobre este campo pude observar que aún queda mucho por investigar y que puede ser muy interesante hacerlo en un futuro inmediato, ya que permitiría analizar y obtener información de la gran cantidad de datos que existen hoy en día debido a la Web 2.0.

Una de las aplicaciones del PLN que, en nuestro idioma el español, no está muy desarrollada es la clasificación de documentos en función del género del autor, lo que nos hizo pensar a mi tutora y a mí en desarrollar un sistema que fuese capaz de realizar esta tarea citada.

La realización de este proyecto pienso que ha sido muy buena, ya que gracias a él he podido profundizar más en el mundo del PLN, en el que me gustaría trabajar en el futuro. Más concretamente he aprendido acerca de clasificación de textos, además he progresado en otros campos como son los siguientes:

- Aprender un lenguaje de programación que no conocía: Python.
- Descubrir diferentes librerías de Python que son utilizadas en el PLN: nltk, scikit-learn, etc.

- Estudiar los distintos enfoques que se utilizan en la clasificación de documentos.
- Estudiar un tema, en este caso la clasificación de textos en función del género del autor, mediante la lectura de diversos artículos, cosa que antes no era capaz de hacer.

6.2. Trabajos futuros

Uno de los principales problemas que he encontrado durante el desarrollo del proyecto ha sido la escasez de corpus limpios y etiquetados por el género del autor en español, por lo que un trabajo futuro podría ser la unión de los pocos corpus que hay, o la limpieza de ellos, haciendo mucho más fácil y efectiva la clasificación de los mismos. También se podría llevar a cabo una etiquetación manual de los documentos por expertos para tener un corpus aún mayor con el que trabajar el sistema creado.

Otro trabajo que podría producir una mejora en el sistema es la realización de un análisis sintáctico para poder obtener más características con las que entrenar nuestro sistema y posiblemente obtener mejores resultados.

También, se podría crear una interfaz gráfica que ayudará a mostrar los resultados y elegir los diferentes parámetros que pueden modificarse a la hora de ejecutar el sistema.

Por último se podría ampliar el sistema hacia otros idiomas, que ya han sido investigados en esta tarea como por ejemplo inglés, alemán o portugués.

Bibliografía

Villegas, M. P., Garciarena Ucelay, M. J., Errecalde, M. L., & Cagnina, L. (2014). A Spanish text corpus for the author profiling task. In XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014).

Molina-González, M. D., Martínez-Cámara, E., Martín-Valdivia, M. T., & Perea-Ortega, J. M. (2013). Semantic orientation for polarity classification in Spanish reviews. Expert Systems with Applications, 40(18), 7250-7257.

Park, G., Yaden, D. B., Schwartz, H. A., Kern, M. L., Eichstaedt, J. C., Kosinski, M., ... & Seligman, M. E. (2016). Women are warmer but no less assertive than men: Gender and language on Facebook. PloS one, 11(5), e0155885.

Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., ... & Gordon, J. (2012, October). Empirical study of machine learning based approach for opinion mining in tweets. In Mexican international conference on Artificial intelligence (pp. 1-14). Springer, Berlin, Heidelberg.

Rangel F., Rosso P., Koppel M., Stamatatos E., Inches G. Overview of the Author Profiling Task at PAN 2013.
In: Forner P., Navigli R., Tufis D. (Eds.), CLEF 2013 Labs and Workshops, Notebook Papers. CEUR-WS.org, vol. 1179, Valencia, Spain, September 23-26

Rangel F., Rosso P., Chugur I., Potthast M., Trenkmann M., Stein B., Verhoeven B., Daelemans W. Overview of the 2nd Author Profiling Task at PAN 2014.
In: Cappellato L., Ferro N., Halvey M., Kraaij W. (Eds.) CLEF 2014 Labs and Workshops, Notebook Papers. CEUR-WS.org, vol. 1180, pp. 898-827

Francisco Rangel, Fabio Celli, Paolo Rosso, Martin Potthast, Benno Stein, Walter Daelemans. Overview of the 3rd author profiling task at pan 2015.
In Cappellato L., Ferro N., Jones G., San Juan E. (Eds.) CLEF 2015 labs and workshops, notebook papers. CEUR Workshop Proceedings. CEUR-WS.org, vol. 1391, 2015.

Francisco Rangel, Paolo Rosso, Ben Verhoeven, Walter Daelemans, Martin Potthast, Benno Stein. Overview of the 4th author profiling task at PAN 2016: cross-genre evaluations.
In Working Notes Papers of the CLEF 2016 Evaluation Labs, CEUR Workshop Proceedings. CLEF and CEUR-WS.org, vol.1609, 2016.

Francisco Rangel, Paolo Rosso, Martin Potthast, Benno Stein. Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter.
In Working Notes Papers of the CLEF 2017 Evaluation Labs, CEUR Workshop Proceedings. CLEF and CEUR-WS.org, vol. 1866, 2017

Pennebaker, J. W., Mehl, M. R., & Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. Annual review of psychology, 54(1), 547-577.

Company, J. S., & Wanner, L. (2007, May). How to use less features and reach better performance in author gender identification. In The 9th edition of the Language Resources and Evaluation Conference (LREC) (pp. 26-31).

López-Monroy, A. P., Montes-y-Gómez, M., Escalante, H. J., Villaseñor-Pineda, L., & Stamatatos, E. (2015). Discriminative subprofile-specific representations for author profiling in social media. Knowledge-Based Systems, 89, 134-147.

De-Arteaga, M., Jimenez, S., Duenas, G., Mancera, S., & Baquero, J. (2013). Author profiling using corpus statistics, lexicons and stylistic features. Online Working Notes of the 10th PAN evaluation lab on uncovering plagiarism, authorship. and social misuse, CLEF.

Pérez Pérez, M. J. (2012). Guía comparativa de Metodologías ágiles.

Pardo, F. M. R. (2016). Author Profiling en Social Media: Identificación de Edad, Sexo y Variedad del Lenguaje (Doctoral dissertation).

Koppel, M., Argamon, S., & Shimoni, A. R. (2002). Automatically categorizing written texts by author gender. Literary and Linguistic Computing, 17(4), 401-412.

Argamon, S., Koppel, M., Fine, J., & Shimoni, A. R. (2003). Gender, genre, and writing style in formal written texts. TEXT-THE HAGUE THEN AMSTERDAM THEN BERLIN-, 23(3), 321-346.

Estival, D., Gaustad, T., Pham, S. B., Radford, W., & Hutchinson, B. (2007). Author profiling for English emails. In Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (pp. 263-272).

Schler, J., Koppel, M., Argamon, S., & Pennebaker, J. W. (2006, March). Effects of age and gender on blogging. In AAAI spring symposium: Computational approaches to analyzing weblogs (Vol. 6, pp. 199-205).

Yan, X., & Yan, L. (2006, March). Gender Classification of Weblog Authors. In AAAI spring symposium: computational approaches to analyzing weblogs (pp. 228-230).

Goswami, S., Sarkar, S., & Rustagi, M. (2009, March). Stylometric analysis of bloggers' age and gender. In Third International AAAI Conference on Weblogs and Social Media.

Mukherjee, A., & Liu, B. (2010, October). Improving gender classification of blog authors. In Proceedings of the 2010 conference on Empirical Methods in natural Language Processing (pp. 207-217). Association for Computational Linguistics.

Juan Soler Company, & Wanner, L. (2015). Multiple Language Gender Identification for Blog Posts. In CogSci.

Otterbacher, J. (2010, October). Inferring gender of movie reviewers: exploiting writing style, content and metadata. In Proceedings of the 19th ACM international conference on Information and knowledge management (pp. 369-378). ACM.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., ... & Ungar, L. H. (2013). Personality, gender, and age in the language of social media: The open-vocabulary approach. PloS one, 8(9), e73791.

Sap, M., Park, G., Eichstaedt, J., Kern, M., Stillwell, D., Kosinski, M., ... & Schwartz, H. A. (2014). Developing age and gender predictive lexica over social media. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1146-1151).

Maharjan, S., Shrestha, P., Solorio, T., & Hasan, R. (2014, November). A straightforward author profiling approach in mapreduce. In Ibero-American Conference on Artificial Intelligence (pp. 95-107). Springer, Cham.

Francis, W. N., & Kucera, H. (1982). Frequency analysis of English usage.

Anexo A: Manual de instalación

En este primer anexo se presenta el manual de instalación que hay que realizar en caso de querer ejecutar el sistema.

El único proceso que habrá que llevar a cabo para poner en marcha el sistema será la instalación de Python y sus componentes.

Para la realización de este proyecto se ha utilizado Python en su versión 3.6. En primer lugar descargamos la versión correspondiente desde la web de Python[11].

En caso de querer utilizar el sistema en Ubuntu que ha sido el sistema operativo utilizado para crearlo, se puede facilitar el proceso mediante el uso de la herramienta **apt-get**, utilizando el siguiente comando:

```
sudo apt-get install python3.6
```

Posteriormente se debe instalar la herramienta **pip 3**, ya que facilitará el proceso de instalación de los módulos de Python necesarios. Para instalar esta herramienta habrá que utilizar los siguientes comandos:

```
sudo apt-get install python3-setuptools

sudo easy_install3
```

Una vez instalado **pip 3**, se instalarán los módulos necesarios, con los siguientes comandos:

```
sudo pip3 install nltk

sudo pip3 install sklearn

sudo pip3 install numpy

sudo pip3 install argparse
```

[11] http://www.python.org/downloads/

Una vez instalados los paquetes necesarios, el sistema podrá ser ejecutado.

Anexo B: Manual de usuario

En este segundo anexo se presenta un manual del sistema creado con la intención de resolver cualquier duda que pueda tener el usuario a la hora de utilizarlo.

Para poder utilizar el sistema es necesario que el usuario tenga un ordenador, aunque no es necesario que tenga acceso a Internet.

El primer paso una vez se tiene un corpus divido en dos subconjuntos (entrenamiento y test), consiste en extraer las características de ese conjunto de documentos. Para ello simplemente es necesario estar situados en la carpeta que contiene el script encargado de extraer las características y ejecutar el siguiente comando:

python featuresExtraction.py

Este comando ejecutará el extractor de características con los corpus que se proporcionan por defecto, procedentes del corpus SpanText, explicado en el capítulo 3 punto 3.1.2. También se puede ejecutar con otro corpus, para ello bastará con seguir las indicaciones que se muestran al ejecutar el comando que se muestra en la siguiente ilustración (Ilustración B.1.):

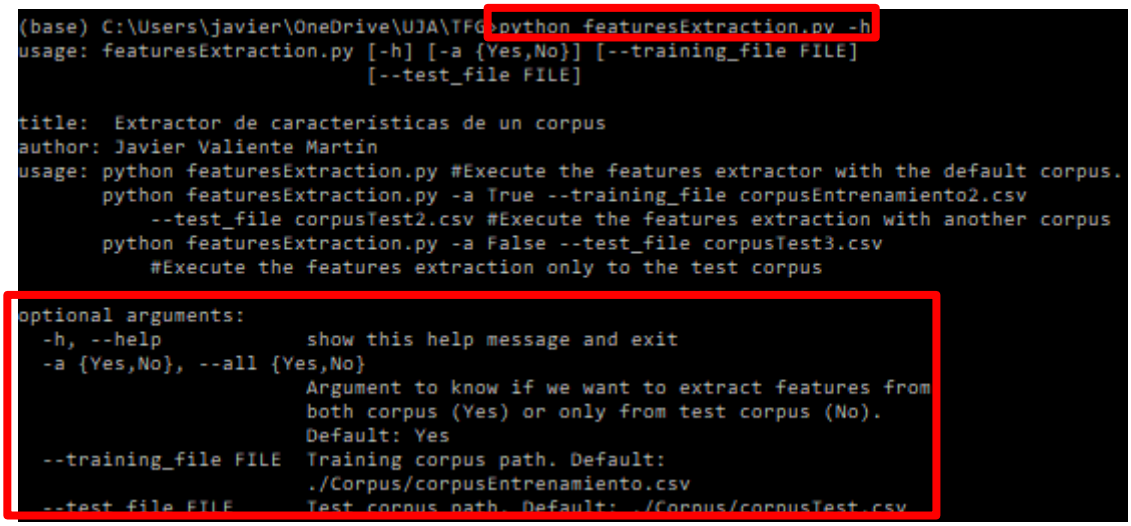


Ilustración B.1. Ayuda del extractor de características.

También es posible, mediante el argumento '-a {Yes, No} --all {Yes, No}', elegir si queremos obtener las características de ambos corpus (entrenamiento y test) o únicamente queremos obtenerlas del corpus de test.

A continuación se muestran algunos ejemplos de ejecución del extractor de características:

- Ejecuta el extractor de características sobre ambas partes de un corpus (entrenamiento y test) diferente al propuesto por defecto:

python featuresExtraction.py -a *True* --training_file *corpusEntrenamiento2.csv* --test_file *corpusTest2.csv*

- Ejecuta el extractor de características solo a un corpus de test:

python featuresExtraction.py --all *False* --test_file *corpusTest3.csv*

Una vez que tenemos extraídas las características podremos ejecutar el sistema clasificador, basta con situarnos en la carpeta que contiene el script del clasificador y ejecutar el siguiente comando:

python classificator.py

Este comando ejecutará el sistema con los valores por defecto (que se explicarán a continuación). Si se quiere ejecutar con otros valores, se puede ejecutar el comando que se muestra en la siguiente ilustración (Ilustración B.2.), y donde aparecerán todos los parámetros que se pueden modificar.

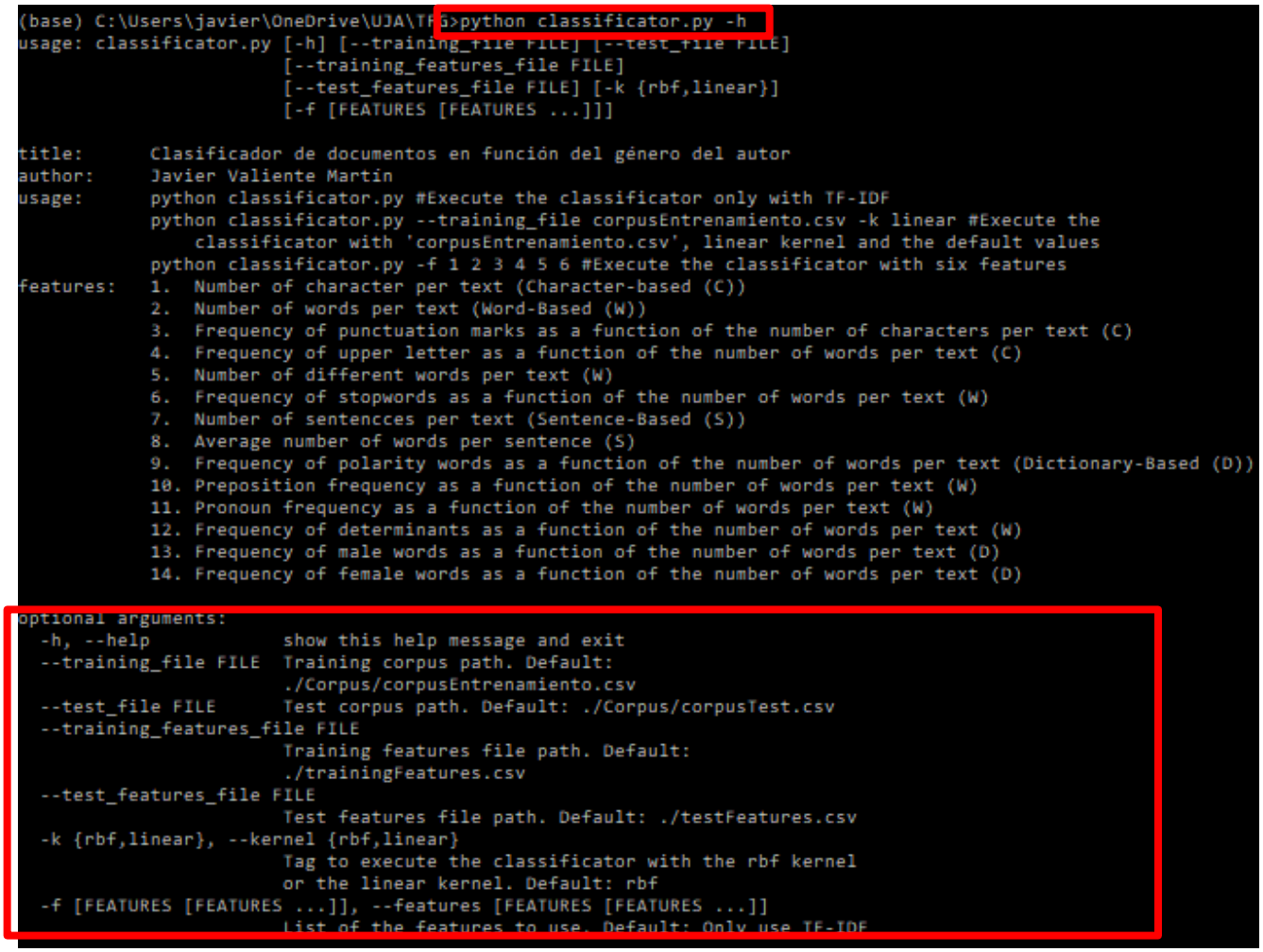


Ilustración B.2. Ayuda del script del clasificador

Los parámetros que pueden ser modificados a la hora de ejecutar el sistema son los siguientes:

- Corpus de entrenamiento: corresponde al parámetro '--training_file FILE', es posible elegir un fichero que contenga un corpus de entrenamiento válido, como el especificado en el capítulo 5 punto 5.6.2. Para ello habrá que indicar la ruta donde se encuentra el fichero. El valor por defecto de este parámetro es el corpus principal de entrenamiento utilizado para el esta fase del proyecto.
- Corpus de test: corresponde al parámetro '--test_file FILE', al igual que con el corpus de entrenamiento se puede proporcionar un fichero que contenga un corpus para el test siempre que tenga el formato especificado en el mismo capítulo y punto citado en el parámetro anterior. Se debe indicar la ruta donde

se encuentra el fichero y el valor por defecto de este parámetro es el corpus de test creado para las pruebas principales del proyecto.

- Fichero con las características del corpus de entrenamiento: corresponde al parámetro '—training_features_file FILE'. Permite utilizar un fichero que contenga las características extraídas del corpus de entrenamiento que se pase como parámetro. Se debe indicar la ruta donde se encuentra el fichero y debe tener el formato especificado en el capítulo 5 punto 6.3. El valor por defecto es el fichero que contiene las características extraídas del corpus de entrenamiento utilizado.
- Fichero con las características del corpus de test: corresponde al parámetro '—test_features_file FILE'. Permite utilizar un fichero que contenga las características extraídas del corpus de test que se pase como parámetro. Se debe indicar la ruta donde se encuentra el fichero y debe tener el formato especificado en el capítulo 5 punto 6.3. El valor por defecto es el fichero que contiene las características extraídas del corpus de test utilizado.
- Núcleo del clasificador: corresponde al parámetro '-k' o '--kernel {rbf, linear}'. Permite ejecutar el clasificador con uno de los dos núcleos que se proporcionan: RBF o Linear. El núcleo por defecto es RBF.
- Características: corresponde al parámetro '-f [FEATURES]' o '--features [FEATURES]'. Este parámetro fue explicado en el capítulo 5 punto 6.5. Permite elegir las características a utilizar para clasificar los textos del total de 14 posibles que se proporcionan (todas ellas explicadas en el capítulo 5 punto 6.3.). La característica por defecto es TF-IDF.

Los corpus que se pasen como parámetro deben de ser los mismos que se han utilizado para extraer las características.

Por último se muestran algunos ejemplos de ejecución del clasificador:

- Ejecuta el clasificador con los valores por defecto.

```
python classificator.py
```

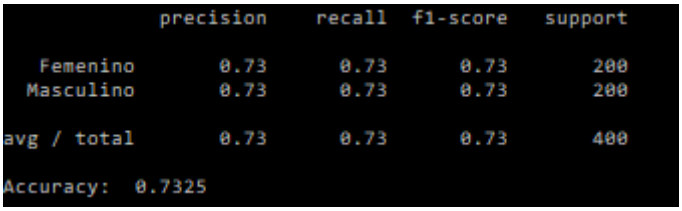
- Ejecuta el clasificador con los corpus especificados en la orden y el núcleo linear. El resto serán los valores por defecto.

python classificator.py --training_file *corpusEntrenamiento.csv* --test_file *corpusText.csv* -k *linear*

- Ejecuta el clasificador con el núcleo RBF y las primeras 6 características. El resto serán los valores por defecto.

python classificator.py -k *rbf* -features *1 2 3 4 5 6*

Una vez se haya ejecutado el clasificador se visualizarán en pantalla los resultados, tal y como se muestran en la siguiente ilustración (Ilustración B.3.)



	precision	recall	f1-score	support
Femenino	0.73	0.73	0.73	200
Masculino	0.73	0.73	0.73	200
avg / total	0.73	0.73	0.73	400

Accuracy: 0.7325

Ilustración B.3. Resultados de una ejecución del sistema

Anexo C: Mantenimiento del sistema

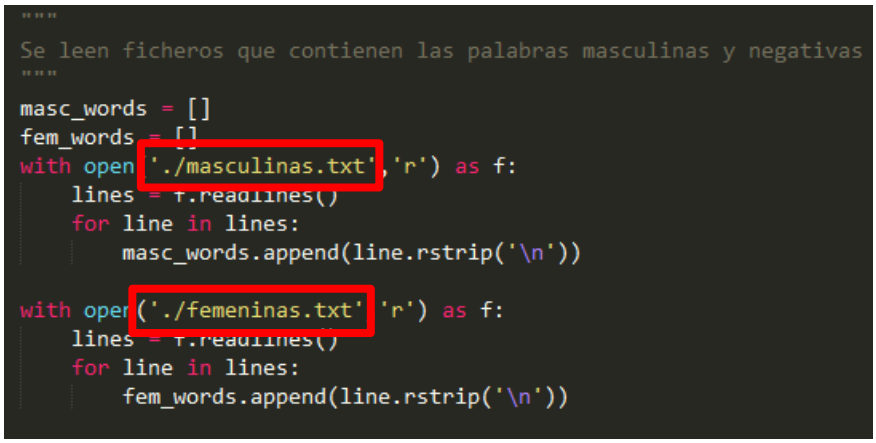
En este anexo se explica cómo se pueden cambiar los lexicones que se han utilizado, para de esta forma poder utilizar otros al gusto del usuario para intentar obtener mejores resultados.

Los lexicones utilizados, ya explicados en el capítulo 3 punto 2 son:

- Lexicón de palabras identificativas del género masculino, fichero con el nombre ‘masculinas.txt’.
- Lexicón de palabras identificativas del género femenino, fichero con el nombre ‘femeninas.txt’.
- Lexicón de palabras con polaridad (iSOL), formado por:
    - Lexicón de palabras positivas, fichero con el nombre ‘positivas_mejorada.csv’.
    - Lexicón de palabras negativas, fichero con el nombre ‘negativas_mejorada.csv’.

Para sustituir alguno de estos lexicones, basta con modificar brevemente el código del script llamado ‘featuresExtraction.py’, situado en la carpeta raíz del proyecto, cambiando la ruta del lexicón que queramos cambiar por la ruta del nuevo lexicón a utilizar.

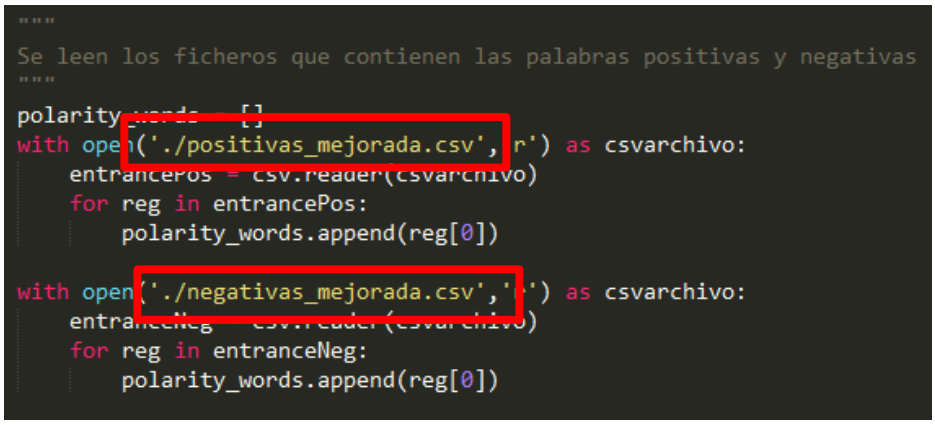
A continuación se muestran dos ilustraciones (Ilustración C.1. e Ilustración C.2.), en las que se puede observar la parte del código que habría que modificar para cambiar cualquiera de los 4 lexicones.



```
"""
Se leen ficheros que contienen las palabras masculinas y negativas
"""
masc_words = []
fem_words = []
with open('./masculinas.txt','r') as f:
    lines = f.readlines()
    for line in lines:
        masc_words.append(line.rstrip('\n'))

with open('./femeninas.txt','r') as f:
    lines = f.readlines()
    for line in lines:
        fem_words.append(line.rstrip('\n'))
```

Ilustración C.1. Código lectura lexicones de palabras identificativas de ambos géneros.



```
"""
Se leen los ficheros que contienen las palabras positivas y negativas
"""
polarity_words = []
with open('./positivas_mejorada.csv', 'r') as csvarchivo:
    entrancePos = csv.reader(csvarchivo)
    for reg in entrancePos:
        polarity_words.append(reg[0])

with open('./negativas_mejorada.csv','r') as csvarchivo:
    entranceNeg = csv.reader(csvarchivo)
    for reg in entranceNeg:
        polarity_words.append(reg[0])
```

Ilustración C.2. Código lectura lexicones de palabras con polaridad.

Anexo D: Índice de ilustraciones

Anexo E: Índice de tablas