Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

UNIVERSIDAD DE JAÉN

IDENTIFICACIÓN DE PERFILES FALSOS EN REDES SOCIALES

Alumno: Juan Carlos Ortal Rienda

Tutor: Luis Alfonso Ureña López

Tutor Adj.: Flor Miriam Plaza Del Arco

Dpto.: Escuela Politécnica Superior de Jaén

Índice

1. Introducción

En este apartado se presentarán todos los campos relacionados con el desarrollo de este proyecto de fin de grado; desde las plataformas objetivo, diferencias entre perfiles y su evolución hasta herramientas y filosofías de trabajo.

1.1. Procesamiento del Lenguaje Natural

Es un campo de las ciencias de computación el cual emplea la Inteligencia Artificial y la lingüística y que tiene como objetivo comprender y automatizar interacciones entre máquina y humano por medio del desarrollo de herramientas y mecanismos eficaces y eficientes que actúen como enlaces entre los dos.

En su inicio el procesamiento del lenguaje se basaba en reglas diseñadas de forma convencional, pero desde finales de 1980 se produjo una revolución de los mismos y se introdujeron herramientas de Aprendizaje Automático.

Este campo de la Inteligencia Artificial este sujeto a un aspecto extremadamente complejo que es el lenguaje humano ya que su comprensión lleva normalmente a ambigüedades en diferentes niveles: léxico, referencial, estructural y pragmático. Además, debe evolucionar al unísono con la lengua lo que conlleva a constantes cambios y el establecimiento de una estructura fija de análisis resulta en una pérdida de precisión ya que debe ser dinámico y adaptativo.

En todos los proyectos relacionados con el lenguaje natural y que tengan como objetivo su procesamiento estarán sujetos a estos cambios o dificultades por lo éste no será una excepción.

1.2. Aprendizaje Automático

Otra rama de la Inteligencia Artificial es el Aprendizaje Automático. Este campo nació junto con la Inteligencia Artificial y la búsqueda de agentes que aprendiesen por si solos. A día de hoy el Aprendizaje Automático está fuertemente ligado a la extracción de datos el cual los utiliza como entrada y los analiza para poder extraer patrones y de ellos realizar un proceso de extracción del conocimiento.

Junto con el Procesamiento del Lenguaje Natural serán las dos grandes herramientas que utilizaremos en nuestro trabajo de fin de grado.

1.3. Redes sociales

Las redes sociales son básicamente la plataforma de opinión del siglo 21. Conforman estructuras en Internet integradas por usuarios u organizaciones que comparten intereses o valores comunes y en las cuales se crean relaciones entre los mismos de forma rápida. Además, en ellas podemos encontrar jerarquías de usuarios sobre los temas o aspectos en los que se interactúan.

Desde su creación hasta la actualidad, la influencia en la población de estas redes es cada vez mayor. Esto es debido a que el usuario comenzó la interacción en ellas de forma básica y cada vez se ha aumentado más el contenido, significado y tiempo de interacción; por lo que ahora el entorno online social de los usuarios puede influir directamente en la vida de los mismos.

Mas concretamente, la diferencia de interacción también está ligada a las características personales: lugar de procedencia, genero, edad, corriente política… Esto ha provocado que como en muchos otros ámbitos, diferentes usuarios comenzasen a especializarse en corrientes ideológicas o políticas o simplemente solo dedicar su contenido a un tema específico.

Por el contrario, un gran número de usuarios de perfil “normal” o “básico” actualmente también las utilizan para un aprendizaje o una formación, por ejemplo, ha proliferado

en la pandemia de COVID-19 donde los perfiles orientados a la cocina adquirieron una mayor influencia junto a otros muchos.

Este panorama ha hecho que organizaciones y usuarios públicos, como políticos, deportistas o personajes similares tengan un perfil en las redes y como consecuencia tengan un poder mediático en ellas, ya que al tener miles o millones de seguidores pueden influir en su opinión de forma inmediata.

1.4. Bots y Humanos

Esta situación crea jerarquías y pueden enfrentarse entre ellas; lo cual abre otro frente en las redes sociales, las búsquedas de estas situaciones para beneficio propio o particular, ya sea desde una empresa o un individuo.

Aquí es cuando intervienen otros tipos de usuarios los cuales tienen como objetivo cambiar la opinión de otros usuarios para obtener una remuneración ya sea monetaria o no y con un posible objetivo malicioso. Estos usuarios pueden ser humanos pero los más importantes para nuestro trabajo son los automatizados:

Los tipos de usuarios automatizados o bots que podemos encontrar son:

- Chat Bot: Su misión es la de crear y mandar mensajes de forma automática. Estos mensajes pueden ser desde una simple bienvenida hasta una pequeña conversación con otro usuario.
- Crawlers: Su misión es la de recopilar información de forma masiva. Son utilizadas por los grandes buscadores para indexar información sobre los sitios webs.
- Bots Spam: Se encarga de repetir el envío de ciertos mensajes de forma masiva, la cual puede ser publicidad o información negativa para realizar phishing. Estos bots también podemos verlos en las llamadas a teléfonos móviles.
- Scrapers: Diseñados para el robo informático buscando encontrar información sensible del objetivo.

- Bot social: Pueden simular ser un usuario con una suplantación de identidad y como consecuencia utilizar el poder mediático a su favor para engañar o realizar marketing publicitario o por lo contrario estafar al usuario.

En estos últimos profundizaremos más ya que son el objetivo del proyecto. La labor anteriormente mencionada también es importante para nuestro análisis, ya que la propia opinión expresada por el usuario suplantado influye en los seguidores, pero normalmente la suplantación de identidad se realiza a usuarios que tienen muchos seguidores y suelen ser detectadas en poco tiempo.

Por el contrario, la versión opuesta es mucho más interesante; la creación masiva de cuentas automatizadas para la expansión de una opinión o la publicidad de objetos o servicios. El creador de estos bots suplanta o inventa identidades realistas o cercanas a la realidad para que los usuarios humanos tengan una mínima confianza hacia ellos. Si el perfil del bot creado está vacío o no tiene seguidores o no tiene contenido es mucho más sospechoso que una con estas características. De modo que el creador busca las características o las automatiza y con ello crea un bot general el cual crea otros bots.

Estos bots no tuvieron una repercusión extremadamente grande hasta las elecciones de estados unidos de 2016, pero diferentes fuentes sitúan el inicio en las elecciones de 2010.

Estos bots tenían varias misiones:

- La promoción de tuits con el retuiteo automático
- La promoción de toda fuente que sea afín a su opinión, compartiéndola y dándole apoyo.
- La creación de opiniones automáticas para combatir a usuarios que opinan de forma diferente y poder convencer a otros usuarios y atraerlos a su corriente de pensamiento.

El objetivo principal es desinformar a los usuarios. Según investigaciones del MIT:

- Solo el 6% de las cuentas bot difundieron el 31% de la información de baja credibilidad y 34% de todos los artículos de poca veracidad fue creado por ellos.
- Cambiaron los porcentajes de retuit de los usuarios humano de modo que había un 70% de posibilidades de que un usuario humano retuitease una noticia falsa creada por un bot.
- La utilización de bots para aumentar el posicionamiento de tuits era vital antes de la noticia citada, se vuelva viral y se pueda confirmar su veracidad.

Como consecuencia de estos bots se estima por diferentes fuentes, que un 3,2% de los votos de Trump fue conseguidos por los bots sociales.

Pongamos otro ejemplo más específico para poder ver en un ejemplo real cual es el comportamiento de estos bots.

En el atentado terrorista del 22 de marzo de 2017 se produjo un tuit de orientación xenófoba que fue destacado. Podemos observarlo en la Ilustración 1.

Ilustración 1: Tuit Destacado

Este usuario aparentaba ser un usuario normal que “solamente” expresaba su opinión, pero en realidad se trataba de un bot de origen ruso el cual contaba con 50.000

seguidores. Esta misma cuenta ha cambiado de objetivo a lo largo de su existencia: inmigrantes, interrupción del embarazo, pensamientos contrarios a los demócratas estadounidenses… La cuenta fue detectada como bot por la plataforma y fue borrada.

Otro aspecto interesante es que después de hacer un borrado de cuentas pertenecientes a esta red de bots, la plataforma Twitter comenzó a desconfiar de usuarios humanos o de organizaciones que apoyaban a estas cuentas, como el canal de noticias Russia Today.

1.5. Motivación

Como consecuencia al intento de monopolizar de forma automática la opinión en cualquier red social, primeramente, desinforma a los usuarios humanos y los introduce en su pensamiento convirtiéndolos en secuaces de su pensamiento y a posteriori, cuando la existencia de bots imposibles de distinguir por un usuario humano es normal, la red social pierde credibilidad por lo que los propios dirigentes de estas redes invierten para poder eliminar y devolver la confianza a su plataforma.

Primeramente, y debido directamente a las consecuencias de la proliferación de bots sociales el desarrollo de proyecto para su detección es mayor y la utilización del procesamiento del lenguaje es vital, ya que con esta herramienta podremos detectar emociones, opiniones, expresiones… todas características fundamentales y en las cuales bots y humanos difieran enormemente.

Además, otra herramienta fundamental para el desarrollo será la Minería de Datos y el tratamiento inteligente de información.

En una perspectiva general, el desarrollo de proyectos de esta índole amplía enormemente el interés por estos campos a personas ajenas a ellos e informa a los mismo de peligros o simples realidades que se dan en su entorno diario y los cuales pueden influir en su vida sin su permiso o su consciencia.

Para mi propia perspectiva, estos dos campos componen los pilares en los que he sido formado por lo que la utilización de ambos y la ampliación de conocimientos en los mismo resulta de gran interés para mí. En menor medida el desarrollo en lenguajes especialmente útiles en este campo como Python con una investigación de librerías, métodos y configuraciones necesarias para proyecto son muy atractivas para mí y me serán muy útiles en el futuro.

1.6. Propósito

El propósito general de proyecto es el desarrollo de un clasificador de perfiles utilizando las herramientas anteriormente nombradas que pueda distinguir entre perfiles automatizados y perfiles humanos utilizando características extraídas de los mismos. Como añadido este clasificador se deberá presentar en una aplicación web la cual esté al alcance de un usuario o cliente.

1.7. Objetivos

Los objetivos generales a conseguir en el proyecto son:

- Lectura y síntesis de artículos relacionados o similares al proyecto a desarrollar.
- Realización de una planificación de tareas sobre el desarrollo.
- Selección de lenguajes de programación y entorno de desarrollo.
- Realización de un sistema que parta desde archivos base y extraiga toda la información requerida.
- Estudio de herramientas de desarrollo software para aplicaciones web y selección de la más adecuada.
- Desarrollo de una aplicación web compatible con el clasificador.
- Documentación del desarrollo realizado para todos los objetivos anteriores.

1.8. Resultados

Finalizado el proyecto se deberían haber conseguido los siguientes aspectos:

- Investigación de fuentes externas que plasmen ideas o proyectos relacionados con nuestro proyecto y la extracción de conclusiones sobre la misma.
- Estudio de las características más representativas extraídas o no de fuentes externas útiles para la detección de bots y desarrollo del clasificador.
- Diseño de un sistema que automáticamente limpie, sintetice y finalmente extraiga todos los aspectos necesarios para la entrada del clasificador.
- Desarrollo de un clasificador capaz de conseguir el propósito de nuestro proyecto.
- Una aplicación web que utilice nuestro clasificador con la menor complejidad para nuestro cliente o usuario final.
- Una memoria la cual contenga todo el desarrollo del proyecto y sintetice todo lo realizado y concluido.

1.9. Costes

1.9.1. Costes hardware

El equipo utilizado para todo el desarrollo es un laptop ASUS por un valor de 699 euros junto a un disco duro externo TOSHIBA de 50 euros de coste para la realización de copias de seguridad.

1.9.2. Coste software

Los costes de software están fundamentados en las licencias requeridas en los programas utilizados para el desarrollo:

- Sistemas Operativo Windows 10 Home: 0 € (adquirido con laptop)
- Entorno de desarrollo Anaconda: 0 €
- Paquete Django para desarrollo web: 0 €
- Paquete Microsoft Office 2019: 18.5 €

1.9.3. Coste de personal

Estos costes son los asociados al coste salarial de diferentes empleados encargados del desarrollo del proyecto. Para la realización de este proyecto se necesitarían 2 empleados:

- Analista o investigador: el cual será encargado del diseño del sistema y de la recopilación de información
- Programador: encargado de desarrollar e implementar todo el sistema.

Según la estipulación de publicada y actualizada el 8 de septiembre de 2021 en BOE el sueldo base promedio para analista es de 23.819 euros al año. Para el rol de programador junior el sueldo promedio se baraja entre los 16.000 y los 22.000 euros al año por lo que nosotros escogeremos la media de las dos cantidades, 19.000 euros al año.

Puesto que la duración media de desarrollo oscila entre los 2 y 4 meses. El sueldo promedio de los dos roles sumara una cantidad de:

- Programador: 5428,571 euros para un periodo de 4 meses
- Analista: 6805,428 euros para un periodo de 4 meses

1.9.4. Coste total

Puesto que todo el software utilizado es libre los costes asociados al proyecto solo se basarían en el coste hardware y coste de personal los cuales sumados ascenderían a una suma total de 12982,999 euros.

2. Planificación de desarrollo

2.1. Inicio y contexto

En primer lugar, describiremos cual es el punto de partida. En 2019 se realizó un concurso desde PAN 2019 *[1]* el cual otorgaba un premio monetario al modelo que alcanzase una mayor precisión en la clasificación de perfiles de Twitter desde un dataset inicial el cual estaba dividido en dos idiomas: inglés y español. En nuestro proyecto utilizaremos este mismo dataset y crearemos un modelo que tenga el mismo objetivo; la detección y entrenamiento en la clasificación de perfiles en dos clases bot o humano.

Para el desarrollo necesitamos definir cuáles van a ser las herramientas más útiles para este objetivo. Como herramienta más importante utilizaremos la Minería de Datos: con ella podremos desarrollar el modelo clasificador; pondremos en competición a una batería de algoritmos que se entrenarán y se podrán a prueba para poder arrojar resultados y así seleccionar al más eficaz.

Para la creación de un dataset aprovechable por la minería se necesitarán la definición y extracción de características las cuales sean representativas y puedan dar un nivel de entropía de información válido utilizado técnicas de recuperación de información.

Para poder extraer características referidas al contenido necesitamos procesar el lenguaje y el significado del mismo. Para esta tarea la mejor herramienta es el Procesamiento del Lenguaje Natural. Aunque esta herramienta es muy potente sigue siendo dependiente del idioma por lo que podremos encontrar problemas asociados al mismo como ambigüedades léxica, sintácticas o semánticas entre otras muchas, las cuales son extremadamente difíciles de interpretar por una máquina.

2.2. Tareas y Planificación

En primer lugar, mostraremos el diagrama de caso de uso general del proyecto representado en la ilustración 2:

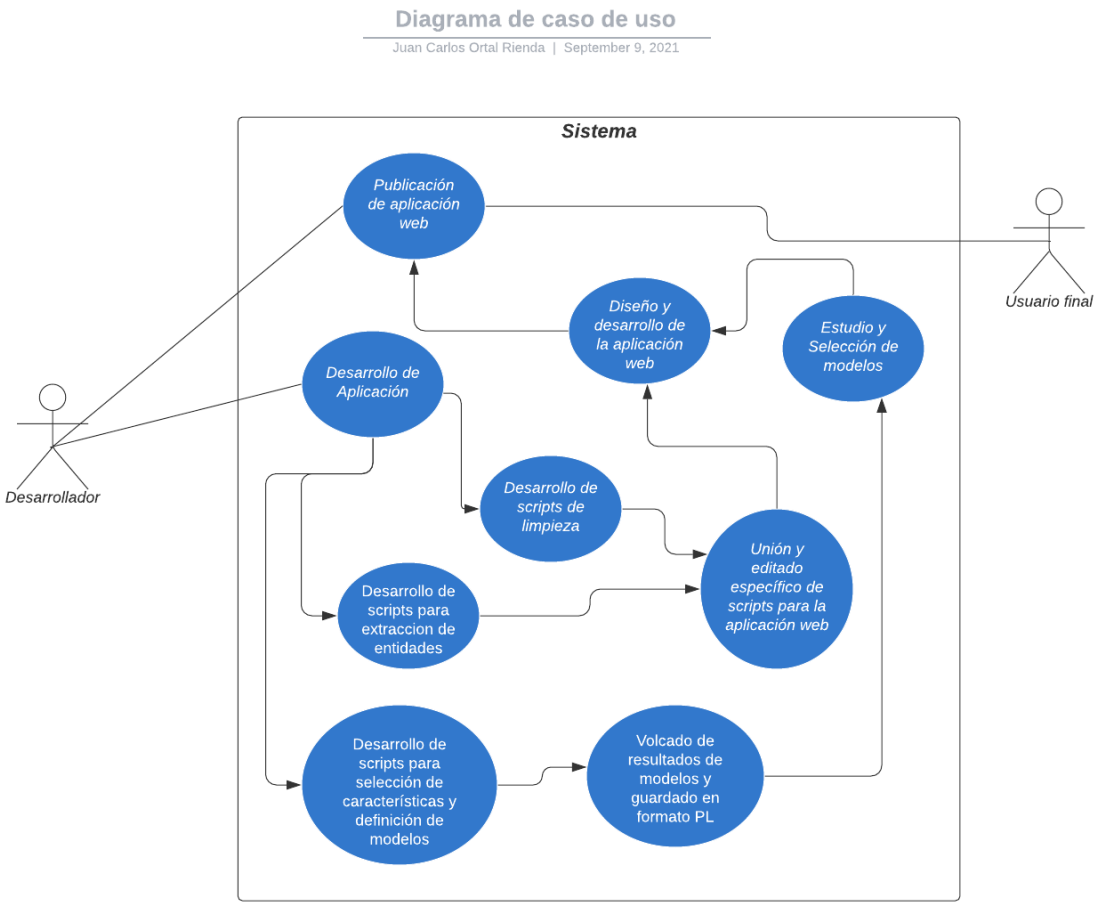

Ilustración 2: Diagrama de casos de uso

Para el desarrollo del proyecto crearemos un flujo de trabajo definiendo las diferentes tareas a realizar. La metodología ágil utilizada para el desarrollo de tareas es KANBAN la cual es una metodología de evolutiva o incremental de la cual extraeremos solo su forma de trabajo, pero no utilizaremos los tableros asociados a esta filosofía.

En su lugar, emplearemos el software gratuito de ASANA el cual permite la planificación de todas las tareas de forma simple y organizada en el tiempo y un seguimiento de su evolución. En la ilustración 3 podemos ver la planificación de tareas, fecha límite de finalización y prioridad.

Elección del lenguaje y entorno de programación	Juan	4 jul	Medio	En curso
Lectura y Análisis de artículos externos	Juan	11 jul	Alto	
Limpieza de ficheros iniciales	Juan	18 jul	Bajo	
Definición de metodologías de extracción de entidades en diferentes idiomas 1	Juan	21 jul	Alto	
Volcado de entidades globales y unitarias por perfil	Juan		Bajo	
Elección de características	Juan	18 jul	Alto	
Extracción de características	Juan	1 ago	Alto	
Análisis de características	Juan	14 ago	Medio	
Definición de selectores de características	Juan	18 jul	Medio	
Definición de algoritmos	Juan	18 jul	Medio	
Entrenamientos de modelos con diferentes selectores 2	Juan	8 ago	Alto	
Volcado de resultados en ficheros				
Recopilación de medidas en ficheros Excel para comparativa				
Elección de plataforma para aplicación web	Juan	18 jul	Alto	
Diseño de la aplicación web	Juan	30 jul	Medio	
Desarrollo de la aplicación web	Juan	27 ago	Alto	

Ilustración 3: Planificación de tareas

Además, podemos apreciar el diagrama de secuencia de tareas para una mejor visión global:

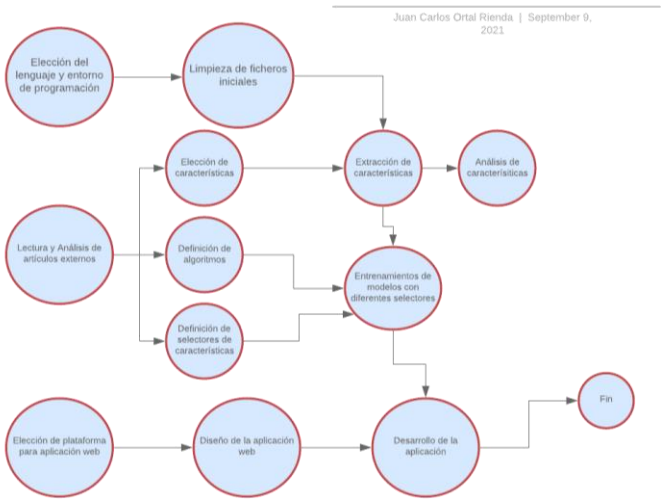

Ilustración 4: Diagrama de secuencia

Seguidamente podemos ver la línea de dependencia y secuencia en la línea temporal. Ilustraciones 5 y 6.

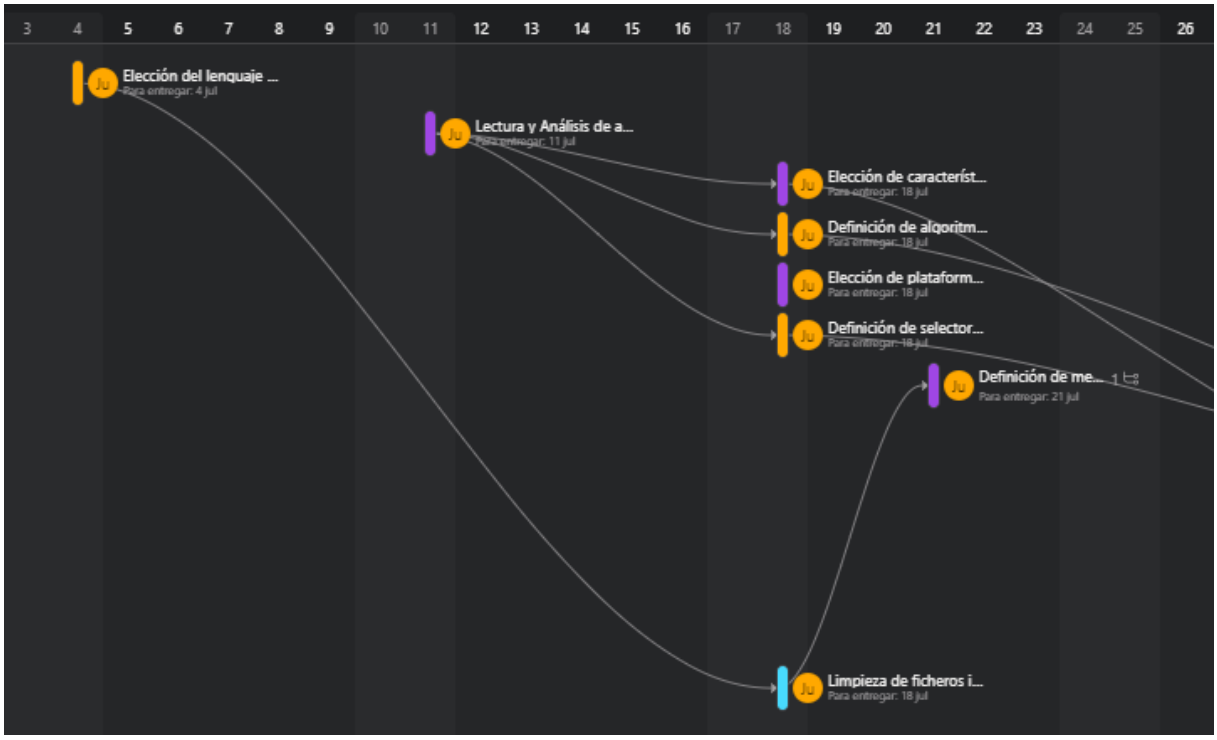

Ilustración 5: Cronología de tareas 1

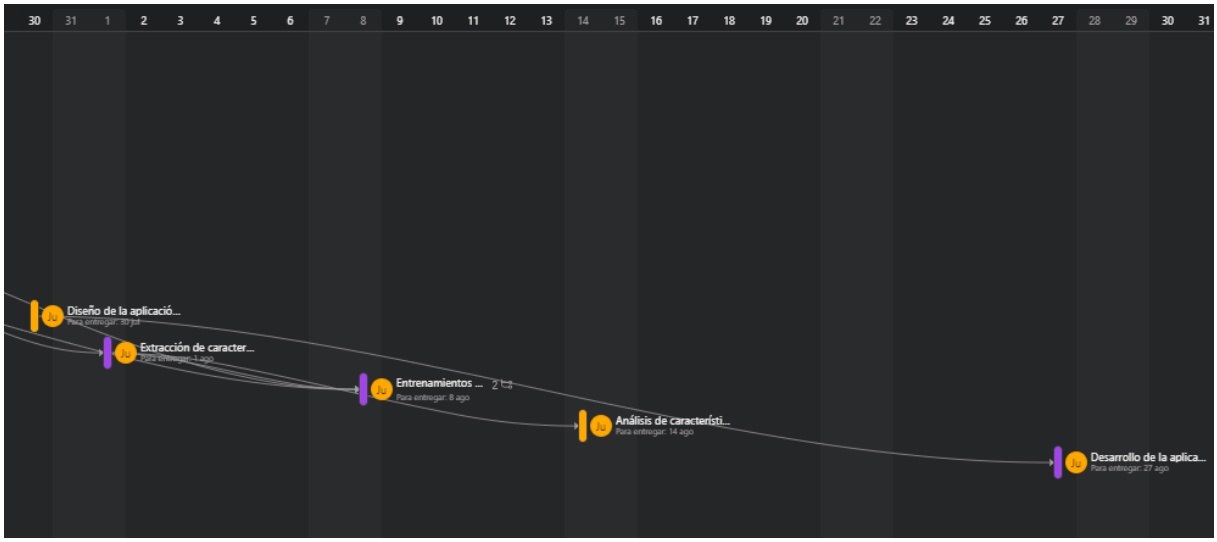

Ilustración 6: Cronología de tareas 2

Como podemos observar las diferentes tareas ilustradas anteriormente, tienen dependencias entre ellas de modo que las líneas que las unen expresan su secuencia. Por ejemplo, la lectura y análisis de los artículos y fuentes externas es necesaria para una elección de características a extraer, la definición de algoritmos, y definición de selectores de características.

En la ilustración 7 podemos ver una representación por trimestres.

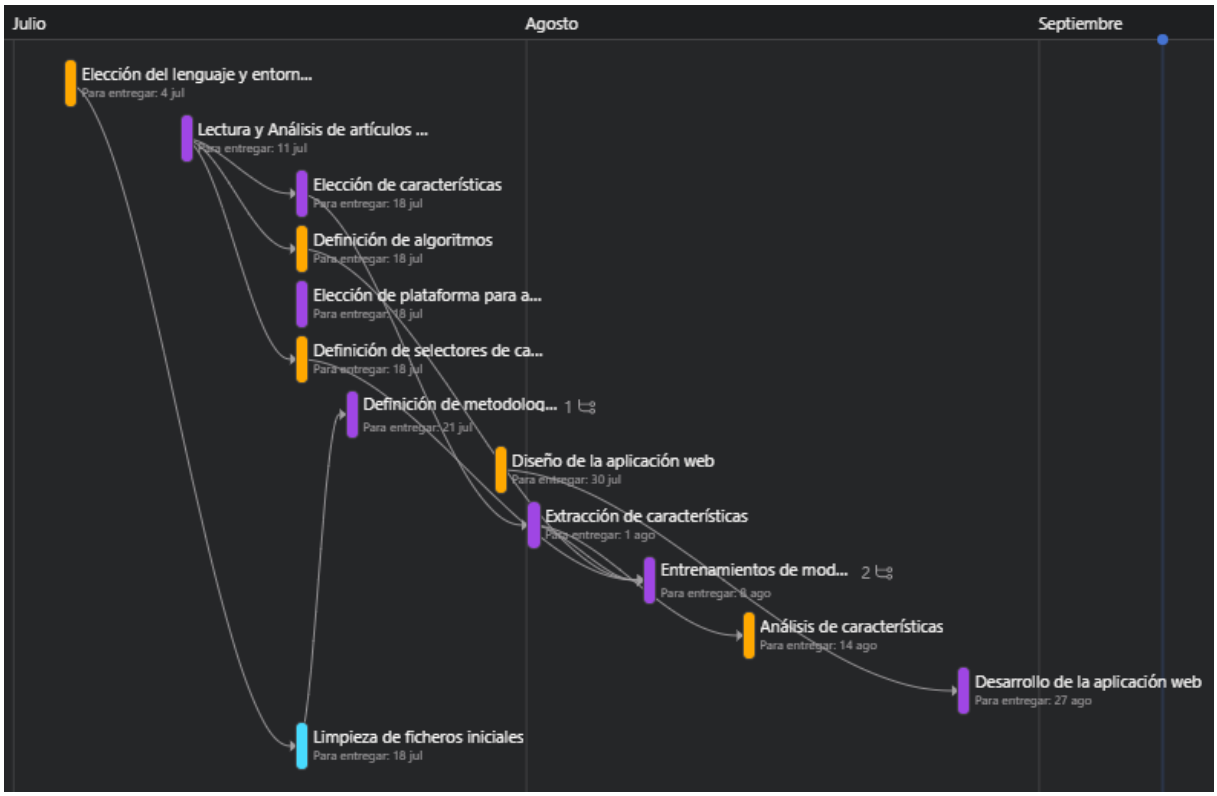

Ilustración 7: Cronología de tareas trimestral

Esta selección de tareas contempla diversas ventajas: Los resultados marcan las metas de las diferentes tareas por lo que el inicio y finalización están bien marcados y solo dependen del auto o ejecutor de las mismas. Desde el completado de la tarea 1 se puede contemplar la evolución del proyecto y se pueden realizar cambios que ayuden en la realización de tareas posteriores.

Al conocer la estructura de los resultados de cada una de las tareas se puede trabajar en las siguientes de forma general, lo cual es una ventaja, pero no se puede avanzar en gran medida por la dependencia entre ellas y el riesgo a iteraciones costosas en tiempo sin valor alguno.

2.3. Propuesta de solución

En el siguiente apartado se describe la solución al problema planteado para este TFG. La solución final será el desarrollo de un conjunto de script automatizados que cumplan de las diferentes tareas planificadas de forma individual para la batería de archivos inicial. Posteriormente y para la aplicación web, todas las tareas serán unificadas en funciones y relacionadas en archivos para un procesamiento único de los archivos subidos por los usuarios.

3. Desarrollo y resolución de tareas

En este apartado se describirán las resoluciones de las diferentes tareas siguiendo la planificación anterior. Las tareas relacionadas con la aplicación web se describirán en el apartado 5. Las tareas realizadas de forma paralela serán descritas en las tareas precedentes a su inicio.

3.1. Elección del lenguaje y entorno de programación

Para la resolución de este apartado nos vamos a basar nuestra propia experiencia en el grado de ingeniería informática. Para los dos campos predominantes en la propuesta de solución: el Procesamiento del Lenguaje Natural, Aprendizaje Automático y Minería de Datos hemos utilizado anteriormente el lenguaje Python puesto que dispone de librerías muy potentes y específicas para el Procesamiento del Lenguaje Natural. Para Aprendizaje Automático y Minería de Datos hemos utilizado anteriormente software WEKA el cual actualmente no es acto para un desarrollo automático. Puesto que el lenguaje para el Procesamiento del Lenguaje Natural esta elegido y Python es totalmente apto para los requerimientos elegiremos este para todo el proyecto.

Como entorno de programación usaremos el ya instalado desde el cursado de la asignatura especifica Anaconda el cual es un software potente y sencillo en el cual no es complicado la instalación de librerías o cualquier tipo de dependencias.

Con ello damos por completa la primera tarea de nuestra planificación.

Seguidamente se realizarán de forma paralela la limpieza de ficheros iniciales y la lectura y análisis de artículos externos.

3.2. Trabajo relacionado

Este apartado refiere a la resolución de la tarea “Lectura y análisis de artículos externos”.

Las características que analicemos en los siguientes archivos o artículos sobre modelos creados deben estar dentro del contexto o entrada que tenemos para el nuestro, es decir, no podemos inventar datos que no tenemos o no podemos conseguir.

3.2.1. Efthimion, P. G., Payne, S., & Proferes, N. (2018):

Bot Detection Techniques to Identify Social Twitter Bots from Southern Methodist University [2]

A continuación, se describirá la premisa desde la que pase este articulo:

3.2.1.1. Premisa

En este artículo se describe el mismo problema que en nuestro proyecto. Se realiza una introducción en la cual se describe la plataforma Twitter y las diferentes herramientas utilizadas específicamente en esta plataforma, hashtags, menciones y retuits. Seguidamente se describe como se ha popularizado el uso de cuentas bots y como han influido en los resultados de diferentes eventos políticos. Estas secciones tienen un gran parecido con el prólogo de nuestro proyecto.

3.2.1.2. Datos iniciales

Los datos iniciales de este proyecto es un dataset llamado Creci-2017 en el cual se proporcionan 3 grupos diferentes de bots y 1 de usuarios. Los tipos de bots son:

- Social Spambots
- Traditional Spambots
- Fake followers
- Real users

Y se analiza las distribuciones de clases con diferentes graficas.

3.2.1.3. Análisis

Como punto más importante del artículo se narra cual es la filosofía de detección que los autores van a tomar para la detección de estos bots. Desde un principio, se hace gran énfasis en que la información dada por la cuenta es de gran utilidad para alcanzar sus objetivos y que son las áreas más básicas de extracción de información.

Seguidamente, los campos de extracción se dividen en 3 campos:

- Perfil de usuario: es la característica que individualiza cada usuario del resto. Se hace hincapié en aspectos como la imagen de perfil la cual atrae atención de los demás usuarios y es un signo descriptivo del mismo. Otro aspecto, y en el cual me encuentro en desacuerdo con el artículo es que se impone la idea de que un bot no sigue a ningún otro usuario y que, por lo tanto, las cuentas con menor número de cuentas seguidas tienen un alto riesgo de ser bots, lo cual es una contradicción ya que en el propio dataset se tiene una clase nombrada como fake followers los cuales contradicen esta teoría al tener como misión única seguir cuentas.
  Entre otras muchas características se imponen algunos ratios de seguidores y seguidos: 100:1 o 50:1 el origen de estos ratios no es explicado.

- Actividad de cuenta: se destaca como un gran indicativo de clasificación y se destaca la hora de publicación y periodos de publicación los cuales para cuentas automatizadas es muy indicativo de bots spamers o similares.
- Minería de texto: solo se destaca el uso de la distancia Levenshtein la cual es el número mínimo de operaciones requeridas para transformar una palabra en otra cambiando caracteres de la misma. Esta distancia es utilizada para la evasión de filtros de spam o filtros de repetición.

Después de este análisis las características extraídas de la batería de perfiles son:

- Ausencia de Id
- Ausencia de imagen de usuario
- Ausencia de nombre
- Menor de 30 seguidores
- Geolocalización
- Idioma inglés
- Descripción con links
- Menos de 50 tuits
- 2:1 ratio de amigos y seguidores
- Mas de 1000 seguidores
- Nunca a tuiteado
- 50:1 ratios amigos seguidores
- 100:1 ratio amigos seguidores
- Ausencia de descripción
- Distancia Levenshtein menor que 30

Estas características no podrían ser extraídas en el dataset con el que contamos, pero igualmente analizaremos la viabilidad de las mismas de una forma generalizada salvo por la distancia Levenshtein.

Como clasificadores destacados solo se utiliza una regresión logística con una máquina de soporte vectorial únicamente.

3.2.1.4. Conclusiones

Tras la lectura, muchos de las formulaciones hechas en el artículo están poco justificadas y meramente se toman por validas: la imposición de ratios o el número mínimo tuits. Pero como se ha mencionado anteriormente la extracción de información se ha orientado a la composición de características booleanas lo cual genera poca entropía de información. Finalmente, el modelo consigue unos valores de precisión altos, pero no se describe la relación prueba test que se utiliza.

En general, en un artículo interesante, pero a la vez simple. Únicamente se ha hecho un estudio de características generales de cuentas de Twitter con una extracción de características básicas, lo único destacado es la distancia Levenshtein la cual era desconocida para mí.

3.2.2. Puertas, E., Moreno-Sandoval, L. G., Plaza-del-Arco, F. M., Alvarado-Valencia, J. A., Pomares-Quimbaya, A., & Alfonso, L. (2019):

Bots and Gender Profiling on Twitter from Pontificia Universidad Javeriana [3]

3.2.2.1. Premisa

Este articulo hace una introducción menor al anterior y clasifica los bots desde un primer momento como diseñados para realizar actividades malignas y cambiar la opinión en diferentes dominios. También hace referencia a las elecciones de Estados Unidos en 2016 y al reclutamiento de terroristas gracias a ellos.

Seguidamente se hace referencia a nuestro mismo punto de partida PAN 2019. Como objetivo de este artículo es la creación de un clasificador con la misma misión que el nuestro.

3.2.2.2. Datos iniciales

Los datos iniciales de este articulo son los mismos que los iniciales de este proyecto.

3.2.1.3. Análisis

En el análisis y propuesta de solución de este artículo se abarcan dos estrategias diferentes:

- La extracción de características desde el vocabulario y términos utilizados en la redacción de tuits.
- Características extraídas en aspectos específicos de la plataforma: hashtags, menciones, retuits y emojis.

A continuación, se hace un preprocesamiento de archivos para una limpieza y selección de información a estudiar y se procede directamente a la extracción de características:

En su selección de características podemos ver:

- Media de longitud de palabra por tuit
- Curtosis de la variable anterior
- Cantidad de emojis utilizados en todo el perfil
- Número de hashtag por tuit
- Número de menciones por tuit
- Número de URLs por tuit
- Número de retuit por tweet
- Diversidad léxica de todos los tuits
- Longitud del tuit
- Curtosis de la variable anterior
- Número de tuits con primera persona del singular
- Número de tuits con segunda persona del singular
- Número de tuits con tercera persona del singular
- Número de tuits con primera y segunda persona del plural
- Número de tuits con tercera persona del plural

Además, se hace referencia a los clasificadores que van a ser utilizados en el modelo:

- Naive Bayes
- Gaussian Naive Bayes
- Complement Naive Bayes
- Logistic Regression
- Random Forest

El procedimiento de entrenamiento del modelo se realiza con un porcentaje del 60% de ejemplos para entrenamiento y un 40% restante para test. De lo que podemos deducir que se va hacer un modelo exigente con un porcentaje de test alto.

Los resultados de este modelo podemos verlo en dos vertientes: teniendo como objetivo el género o teniendo solo como objetivo la clase bot. Los resultados son los siguientes:

Los modelos finales más precisos podemos observarlos en la tabla 1:

Type	Language	Acc	Best Model
BOT	en	0.91	RF
GENDER	en	0.81	RF
BOT	es	0.90	RF
GENDER	es	0.75	LR

Tabla 1: Resultados Algoritmos Art. 2

Como precisión del mejor modelo para la clasificación de bot, representado en la tabla 2:

	Precision		**Recall**		**F1-Score**		**Support**	
Class	en	es	en	es	en	es	en	es
0	0.97	0.96	0.85	0.82	0.91	0.88	620	460
1	0.86	0.84	0.98	0.97	0.92	0.90	620	460
Micro avg	0.91	0.89	0.91	0.89	0.91	0.89	1240	920
Macro avg	0.92	0.90	0.91	0.89	0.91	0.89	1240	920

Tabla 2: Resultados Modelo Bot Art. 2

Como precisión del mejor modelo para el género, representado en la tabla 3:

	Precision		**Recall**		**F1-Score**		**Support**	
Class	en	es	en	es	en	es	en	es
0	0.79	0.76	0.84	0.72	0.81	0.74	310	540
1	0.83	0.73	0.77	0.78	0.80	0.76	310	540
Micro avg	0.81	0.75	0.81	0.75	0.81	0.75	620	1080
Macro avg	0.81	0.75	0.81	0.75	0.81	0.75	620	1080

Tabla 3: Resultados Modelo Género Art. 2

Por lo que podemos observar en las ilustraciones anteriores se consiguen resultados de precisión altos por lo que podemos deducir que los modelos creados realmente funcionan y que pueden ser muy útiles para nuestro proyecto.

3.2.2.4. Conclusiones

Es muy interesante poder ver un punto de vista similar al que nosotros debemos desarrollar y afrontando exactamente el mismo problema.

Algunas de las características podrían ser utilizadas en nuestro proyecto ya que son fáciles de extraer. Las peores candidatas para ello son las referidas a la persona de la que habla o de las que se habla. Con respecto a los algoritmos utilizados elegiremos 3 de ellos: Naive Bayes, Logistic Regression y Random Forest.

3.2.3. Dickerson, J. P., Kagan, V., & Subrahmanian, V. S. (2014, August):

Using Sentiment to Detect Bots on Twitter from University of Maryland [4]

3.2.3.1. Premisa

En este artículo se realiza una breve introducción sobre el tanto por ciento de bots encontrados en diferentes plataformas. En ámbito político se centra en un campo nuevo, las elecciones indias. Seguidamente se explica una nueva filosofía de análisis de sentimientos en la detección de bots y que puede ser la respuesta ante otros proyectos que tuvieron grandes dificultades de detección.

Por último, presenta los 4 grandes campos de extracción de características:

- Técnicas de análisis de sentimientos
- Métricas y estudio de vecindario
- Métricas sintácticas de los tuits
- Modelos semánticos
- Gráficos teóricos y métricas de otro tipo

3.2.3.2. Datos iniciales

Este proyecto parte desde un dataset recolectado en las elecciones indias de 2013 a 2014 que cuenta con 550.000 cuentas de Twitter con 7.7 millones de tuits. Al tener un número muy alto de perfiles podemos destacar que el modelo tendrá un alto interés.

3.2.3.3. Análisis

El propio articulo describe al proyecto como SentiBot el cual aborda diferentes frentes de extracción de conocimiento, comentados anteriormente los cuales sintetiza y agrupa en los siguiente 4 campos:

- Sintaxis de Tuit
- Semántica de Tuit
- Actitud del usuario
- Propiedades del vecindario del usuario

En cada uno de estos ámbitos va a proceder a extraer diferentes características:

Sintaxis de Tuit

- Media de Hashtag
- Media de menciones
- Media de número de enlaces externos
- Media de número de características especiales

Semántica de Tuit

Para este apartado deberemos definir los tópicos los cuales son entidades a las que hace referencia un tuit o un autor en toda su cuenta. Estos van a ser utilizados para poder analizar las diferentes actitudes hacia el mismo tópico desde diferentes usuarios. Las características extraídas son:

- Media de sentimientos hacia un tópico
    - Cada juicio que se hace hacia este tópico sea bueno o malo, se suma y se divide entre la cantidad de juicios

- Overall de media de sentimientos por tópico general
    - Realizadas todas las medias de todos los usuarios que mencionan un tópico se realiza una media general del propio tópico.

- Polaridad de Sentimientos por tópico
    - Por cada usuario y entidad se calcula el tanto por ciento de tuits positivos y negativos.

- Overall de Polaridad de sentimientos por tópico
    - Se realiza una media global sobre la positividad y negatividad del usuario y todos los tópicos de los cuales ha expresado un juicio no neutral

- Rango de contracción
    - Por cada tópico calcularemos el valor X+ que será la fracción de tweets positivos que ha realizado el usuario y un valor X- con la fracción negativa. Además, es calculado este mismo valor, pero para todos los usuarios llamado Y+ e Y-. El rango de contracción de usuario para este tópico se calculará con la siguiente expresión matemática:

$$\mathrm{CR}(u,t) = x_t^+ y_t^- + x_t^- y_t^+.$$

- Rango de acuerdo
    - Por cada tópico y ya calculados los valores anteriores, el rango de acuerdo será:

$$\mathrm{AR}(u,t) = x_t^+ y_t^+ + x_t^- y_t^-$$

Esta característica expresa el rango de acuerdo que tienen todos los usuarios hacia una sola entidad.

- Rango de disonancia
    - Por cada usuario se realiza la sumatoria de la división de todos sus rangos de contradicción y sus rangos de acuerdo.

$$\mathrm{DR}(u) = \sum_{t \in \mathrm{TOI}} \mathrm{CR}(u,t)/\mathrm{AR}(u,t).$$

- Fuerza positiva por usuario
    - Sumatoria de todos los tweets positivos con todas las entidades mencionadas por el mismo

- Overall de fuerza positiva
    - Media de todas las fuerzas positivas de todos los usuarios

- Fuerza negativa
    - Sumatoria de todos los tweets negativos con todas las entidades mencionadas por el mismo

- Overall de fuerza negativa
    - Media de todas las fuerzas negativas de todos los usuarios

Actitud de Usuario

- Propagación de tuit
    - Se calcula la frecuencia de tuiteo que tiene el usuario en periodos de 24 horas y la entropía generada en cada tuit.

- Frecuencia de tuit
    - Número de tuits generados en un periodo de tiempo

- Frecuencia de repetición de tuit
    - Número de veces que el usuario a repetidos tuits anteriores

- Geolocalización
    - Si es posible, agregar geolocalización de cada tweet.

- Flip y Flop de sentimientos

  - Se define dos nuevos conceptos:
    - Flip: Si el usuario ha realizado un juicio negativo sobre un tópico y seguidamente hace un juicio positivo tendremos un Flip.
    - Flop: Si el usuario ha realizado un juicio positivo sobre un tópico y seguidamente hace un juicio negativo tendremos un Flop.

- Varianza de sentimientos por tweet
  - Por cada tópico se hace una media de la variación de sentimientos.

Propiedades del vecindario del usuario

- In-degree
  - Número de seguidores

- Out-Degree
  - Número de seguidos

- In/Out ratio
  - Número de seguidores entre los seguidos

- Rango de contradicción entre los vecinos
  - Similar al rango de contradicción anterior pero ahora solo se calculará los valores Y+ e Y- de los vecinos del usuario.

- Rango de acuerdo entre los vecinos
  - Similar al rango de acuerdo, pero solo con los vecinos

- Rango de disonancia con los vecinos
  - Rango de disonancia con los vecinos calculado con los dos valores anteriores

A continuación, se presentan los algoritmos que van a ser utilizados:

- Gaussian Naive Bayes
- Support Vector Machine
- Random Forest
- Extremly randomized trees
- Adaboost
- Gradient boosting

Por último, se realizan dos apartados más, la comparativa de datos entre los bots y los humanos; cuáles son sus valores y su representación, y las conclusiones y resultados de los diferentes modelos que se han utilizado. Estos datos serán comparados con los de nuestro modelo.

3.2.3.4. Conclusiones

El trabajo realizado en este artículo es de alto valor y a mi parecer me es muy interesante, como se aborda el problema desde frentes ya estudiados y se abren nuevos con nuevos puntos de vista y gracias a ellos se consiguen resultados increíbles. Este articulo sin duda, marcará la línea de pensamiento de nuestro proyecto.

3.3. Limpieza de ficheros iniciales

En primer lugar, se presentará cual es el contexto desde el que parte el proyecto, aunque ya se ha comentado anteriormente. Los archivos tienen extensión XML. Cada archivo representa a un usuario y contienen todos sus tweets recopilados entre etiquetas de lenguaje HTML. Aquí podemos ver un ejemplo de un archivo representado en la ilustración 8:

```
<author lang="en">
      <documents>
            <document><![CDATA[1:7 Wherefore she went after
their families: of Sered, the family of the priests, and the
light shine upon thy head; for I fear the LORD.]]></document>
            <document><![CDATA[And he put his hand over the
host: and they gave out of the house; and the tent, and his
sons'; and they saw; and, behold, they shall lament her: the
daughters of the man clothed with scarlet.]]></document>
```

Ilustración 8: Ejemplo datos iniciales

Nuestra primera tarea es la limpia de estos archivos. Todas las etiquetas deben ser borradas y solo debe dejarse el contenido de los tweets únicamente.

Podemos observar el script utilizado en la ilustración 9:

```
for fichero in ficheros:
    archivo = open(ejemplo_dir + "/" + fichero.name,"r",encoding="utf8")
    nombre_Nuevo = fichero.name.strip().replace("xml","txt")
    resultado = open("pan19/Archivos_EN_Limpios/" + nombre_Nuevo,"w",encoding="utf8")
    contenido = archivo.readlines()
    contador = 0
    del contenido[0]
    del contenido[len(contenido)-1]
    del contenido[len(contenido)-1]
    for linea in contenido:
        #print(linea)
        linea2 = linea.strip().replace("<documents>","",)
        linea3 = linea2.strip().replace("<document><![CDATA[","")
        linea4 = linea3.strip().replace("]]></document>","")
        if len(linea4) >= 3:
            resultado.write(linea4)
            contador += 1
            resultado.write("\n")
    resultado.close()
```

Ilustración 9: Script limpia inicial

Ejecutado este script obtenemos todos los archivos con solo el texto. Una porción de uno de estos archivos se muestra en la ilustración 10:

```
65:18 But be ye far from the Philistines.
4:29 And rose up, and went out.
24:13 My son, keep my mouth hath spoken, saying, Within three years, year after the Midianites.
```

Ilustración 10: Resultados limpia inicial

Con la finalización de esta tarea y la lectura y análisis de artículos concluimos las principales tareas iniciales de nuestro proyecto. Nuestro cronograma de tareas queda de la siguiente forma representada en la ilustración 11.

Ilustración 11: Actualización cronología tareas 1

3.4. Definición de metodología de extracción de entidades en diferentes idiomas.

La siguiente tarea a realizar es la extracción de entidades la cual realizaremos con un script especial para ello y que tendrá diferencias en las librerías. Los aspectos más destacados de la versión para inglés son:

- Utilizaciones de las bibliotecas de NTLK para la extracción de entidades y limpieza y filtrado de palabras:
    - maxent_treebank_pos_tagger → english.pickle
    - maxent_ne_chunker → english_ace_multiclass.pickle
    - Tree bank Word Tokenizer
    - Stop words de NTLK

Para la versión española:

- Utilización de la biblioteca Spacy para la extracción de entidades con la carga del archivo es_core_news_sm
- Utilización de la biblioteca NTLK para la limpieza y filtrado de palabras
    - Stop words de NTLK

Ejecutado estos scripts conseguimos un número alto de entidades por lo que se tiene que realizar un filtrado con las más repetidas y con un tanto por ciento a elegir. Es por ello que se realiza una ordenación por repetición y se escogen el 25% más repetido.

Definimos solo las entidades de cada archivo que están dentro de este 25% para un ahorro de procesamiento.

Podemos observar los dos archivos resultantes de los scripts en las ilustraciones siguientes:

Entidades globales agrupadas, ilustración 12:

great new us picard united states learn york white brexit washington knock
rican keep spain see developer wow thanks cool colorado asian want indones
shion cheese malaysia yo system philadelphia dc jesus region rock yewwinfo
t industry exquisite hour favorite top sitecore tech believe clean courage
fragrancedirect belfast ca pizza build night cant fair added redrum cher s
nce netherlands bannon pessimist inigo republican florence visit pleasure
arkansas shall radiation hunt eastern grey create northwestern beer santa

Ilustración 12: Ejemplo de entidades globales

Entidades por archivos agrupadas, ilustración 13:

1008c35dc72c34ead679c539a0ed7c24
flight american new scottish manhattan york kezzang69 georgian jo_caulfield english city chambers curve gardens aw biggest
100e80cf6283b11b25f05f4d673947ea
italy amy_siskind earth america india iran somalia russia indonesian fahrenthold us china german germany eu british chinese new york united states kans
10142b769515b97369d36b9c0c47383b
virginia new york romanian washington can us florida south carolina america cuban charlottesville politico american cuba white house rainmaker1973 dc
1014403fecf2a8ac15264e80e6513450
north south east west canadian chinese beijing slovenia lake bought short go
1023e968a8aa69cf5b659aa57a478f64
chicago new bloomberg eight boston

Ilustración 13: Ejemplo de entidades por archivo

Con la creación de estos 4 archivos, 2 por cada uno de los idiomas se da por realizada esta tarea y la subtarea ligada.

3.5. Elección de Características

Después del estudio de los diferentes artículos procedemos a la selección que características van a ser elegidas finalmente. Para ello vamos a seguir una filosofía hibrida escogeremos las mejores características “antiguas”; las extraídas en los dos primeros artículos y que son de una simple y básica extracción y como hemos comprobado son útiles en detección.

Y además las extraídas en el tercer artículo pues siguen ámbitos “nuevos” con perspectivas nuevas y que incluyen el procesamiento de sentimientos y análisis de lenguaje que puede ser determinante para la clasificación.

Esta filosofía no quiere decir que se recolecten todas las características o no se creen nuevas. Las características no elegidas son:

Desde el artículo 1:

- La característica más interesante no elegida es la distancia Levenshtein la cual supondría un gasto computacional enorme en nuestro proyecto y puede que las ejecuciones sean insoportables para el equipo hardware de desarrollo.

Desde el artículo 2:

- Las variables referidas a la persona que escribe el tuit pueden generar mucho ruido puesto que la referencia a pronombres personales puede ser ambigua y aunque interesante no es elegida además por su coste computacional.
- Las variables de curtosis podrían ser más interesantes extraerlas como medidas analísticas en el análisis de características.

Desde el artículo 3:

- Todas las características no elegidas no son compatibles con el dataset de entrada desde el que parte el proyecto.

Descartadas todas las anteriores características las características elegidas finalmente son las mostradas en la tabla 4:

Semánticas	Sintácticas
• Rango de contradicción • Rango de acuerdo • Rango de disonancia • Media de positivismo • Media de negativismo • Flip Flop de sentimientos	• Media de hashtags encontrados • Media de RT encontrados • Media de Link Exteriores • Media de menciones • Diversidad Léxica • Media de longitud de palabra • Extensión media de tuit • Repeticiones de tuit

Tabla 4: Características finales

En el desarrollo del script para la extracción de características encontramos una nueva librería que puede extraer en nivel de emociones que existan en el tweet. Esta librería es text2emotions, la cual va a arrojar 5 emociones diferentes:

- Happy
- Angry
- Surprise
- Sad
- Fear

El problema de esta librería es que solo es válida para inglés por lo que solo servirá para el modelo referente a este idioma.

3.6. Extracción de características

Con estas últimas características obtenemos 18 características diferentes las cuales tendrán que ser extraídas en un solo script y volcadas en un archivo.

Aspectos a destacar del script para inglés:

- Utilización de la biblioteca NTLK.Sentiment llamada Sentiment Intensity Analyzer la cual con un tuit como entrada devuelve 4 valores: Positivo, Negativo, Neutral y Composición. Los cuales son solo usados los dos primeros.
- Utilización de el mismo tokenizador utilizado en la extracción de entidades.

Aspectos a destacar del script para español:

- Utilización de la biblioteca sentiment_analysis_spanish para la valoración de los tweets positivos o negativos.

Todos los demás cálculos se hacen de forma aritmética.

Ejecutados estos dos scripts obtenemos dos archivos de resultados diferentes para los dos idiomas:

Porción de resultados para inglés mostrados en la ilustración 14:

```
0.03125,0.03125,1.0,0.433,0.122,13.0,0.016529,0.008264,7.454545,
1.0,0.066116,0.247934,0.438017,0.008264,0.0,0.0,0.008264,0.0,hum
an
0.060967,0.046176,5.1399,0.145167,0.131875,17.394231,0.0,0.01923
1,10.317308,1.0,0.173077,0.846154,0.519231,0.005577,0.0,0.016827
,0.007981,0.027212,human
0.012104,0.037896,0.319408,0.226,0.0,16.839286,0.0,0.0,9.357143,
1.0,0.196429,0.732143,0.303571,0.0,0.0,0.014911,0.002232,0.01857
1,human
```

Ilustración 14: Ejemplo características inglés

Porción de resultados para español mostrados en la ilustración 15:

```
0.055556,0.055556,12.0,0.02955,0.303783,12.933333,0.0,0.342857
,6.67619,1.0,0.12381,0.352381,0.114286,human
0.009434,0.009434,4.0,0.054621,0.745379,17.91,0.0,0.05,9.02,1.
0,1.0,0.0,1.0,bot
0.051095,0.051095,14.0,0.028491,0.289691,14.146789,0.0,0.40367
,7.155963,1.0,0.036697,0.40367,0.357798,human
0.025641,0.025641,8.0,0.006329,0.141819,16.8,0.0,0.490909,8.38
1818,1.0,0.045455,0.354545,0.209091,human
```

Ilustración 15: Ejemplo características español

Con la finalización de esta tarea queda realizado todo el proceso de extracción y limpieza de información de todos los archivos iniciales por lo que a partir de este punto la información extraída será resolutiva, es decir compondrá conclusiones a nuestro proyecto y podrán ser expuestas en el apartado final.

Nuestro diagrama de tareas quedaría de la siguiente forma representado en la ilustración 16:

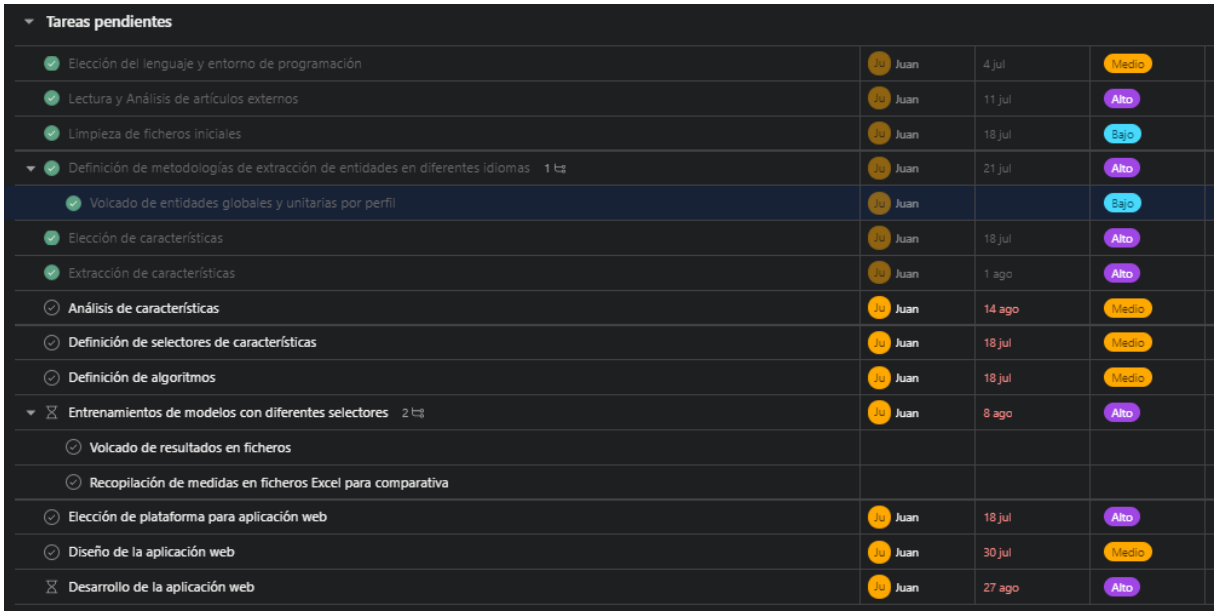

Ilustración 16: Actualización cronología de tareas 2

3.7. Análisis de características

En esta sección se analizarán los valores de las características representadas y podremos evaluar los resultados. Para la visualización de datos y representación utilizaremos la herramienta Tableau. Además, compararemos los resultados con uno de los artículos anteriormente mencionados. Esta es la relación de clase desde la que partimos en inglés y seguidamente español, tablas 5 y 6 respectivamente:

Clase	
bot	1.880
human	2.049

Tabla 5: Equilibrio clases inglés

Clase	
bot	1.478
human	1.500

Tabla 6: Equilibrio clases español

Analizaremos los datos obtenidos en los diferentes datasets. En primer lugar, compararemos las características de semánticas entre bots y humanos.

Destacar que para el análisis de los sentimientos positivos o negativos en todos los tuits de cada usuario se emplean bibliotecas ajenas al desarrollo del proyecto y que son diferentes entre los idiomas. Es por ello que estos evaluadores influirán enormemente en los datos arrojados a continuación.

- Rango de contradicción

Clase	
bot	0,18756
human	0,23930

Tabla 7: Media conteo rango de contradicción inglés

En primer lugar, analizaremos el promedio de valores de la característica. En este caso los humanos tienen un mayor promedio de contradicción que los bots como podemos ver en la tabla 7.

Seguidamente veremos la gráfica por estratos, representada en la ilustración 17:

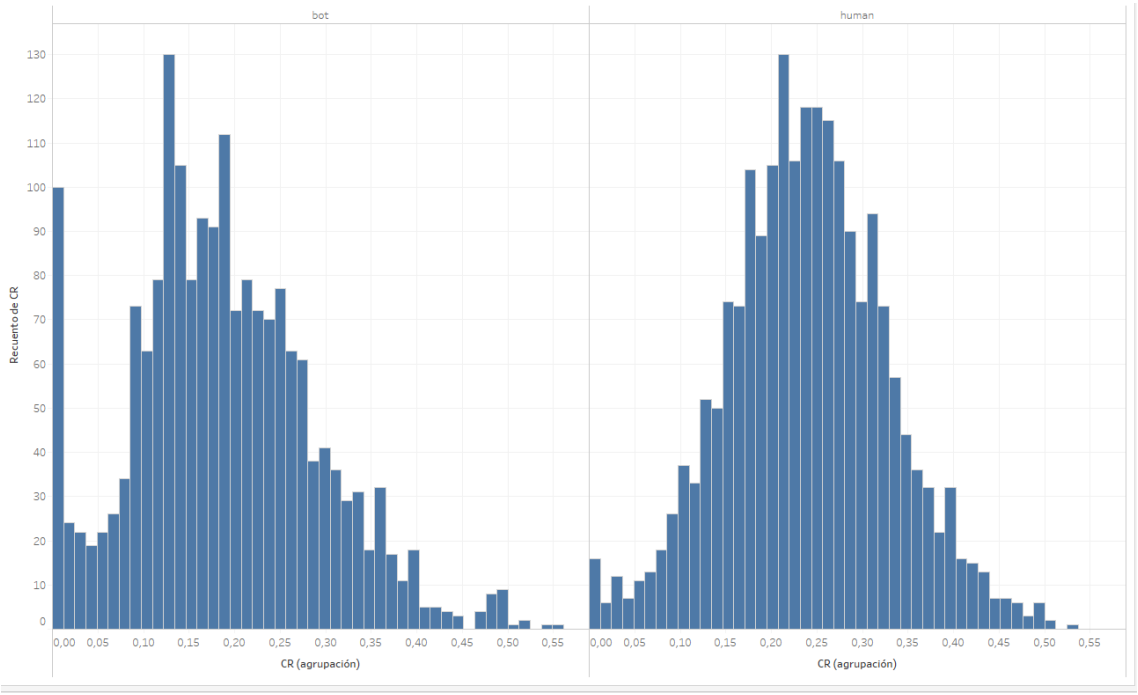

Ilustración 17: Estratos de recuento rango de contradicción inglés

Podemos observar el parecido de las clases, pero destaca que muchos de los bots de nuestro dataset tienen un valor 0 de contradicción junto con los demás usuarios. Por el contrario, hay bots que demuestran tener un rango de contradicción más altos que superan el umbral de los humanos.

Mostraremos ahora los resultados para español en la tabla 8:

Clase	
bot	0,17744
human	0,21562

Tabla 8: Media conteo rango de contradicción español

El promedio extraído es muy parecido al arrojado por el dataset de inglés.

Visualizaremos los diferentes estratos, a través de la ilustración 18:

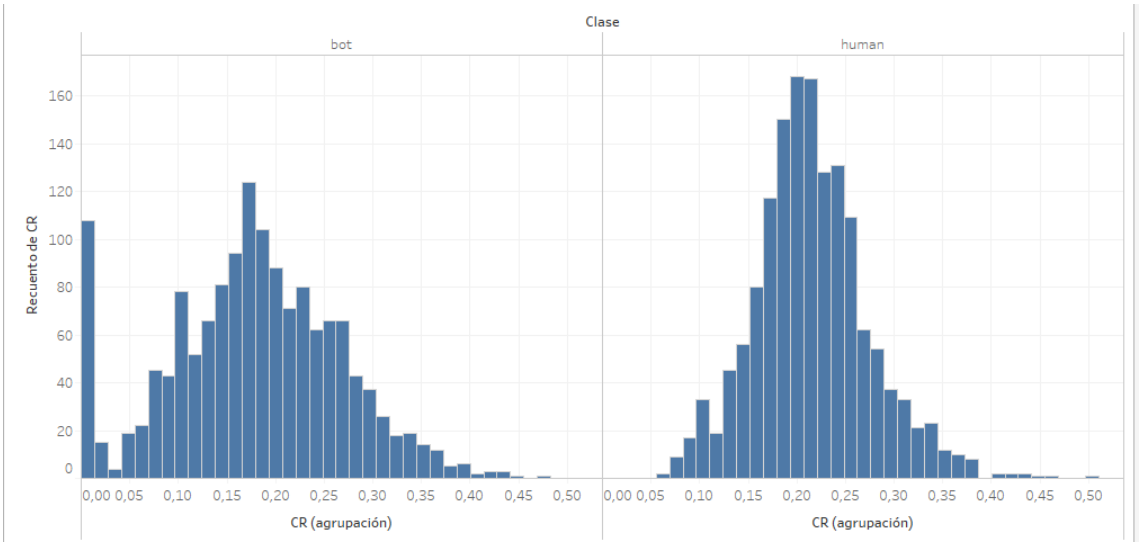

Ilustración 18: Estratos de recuento rango de contradicción español

En la anterior gráfica podemos observar las mismas características; la cantidad de bots sin un rango de contradicción es mucho mayor y la mayoría de humanos se agrupan en una pequeña franja de valores.

- <u>Rango de acuerdo</u>

Clase	
bot	0,39035
human	0,40423

Tabla 9: Media de rango de acuerdo inglés

Se puede apreciar como los valores están más altos en el rango de acuerdo. Además, los promedios de valores están más cercanos observados en la tabla 9. Seguidamente visualizaremos la gráfica por estratos, representados en la ilustración 19:

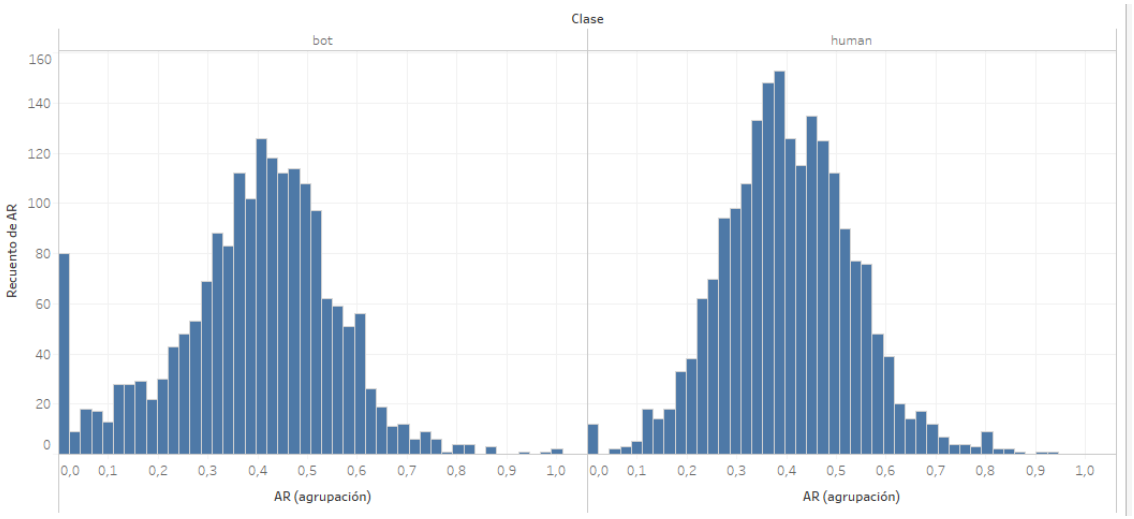

Ilustración 19: Estratos de recuento rango de acuerdo inglés

Podemos observar los mismos patrones que en el rango de contradicción; los valores nulos están mucho más presentes en los bots que en los humanos y los máximos valores se dan en esta misma clase, aunque particularmente la diferencia aquí es menor.

Los valores para español son los siguientes, representados en la ilustración 20:

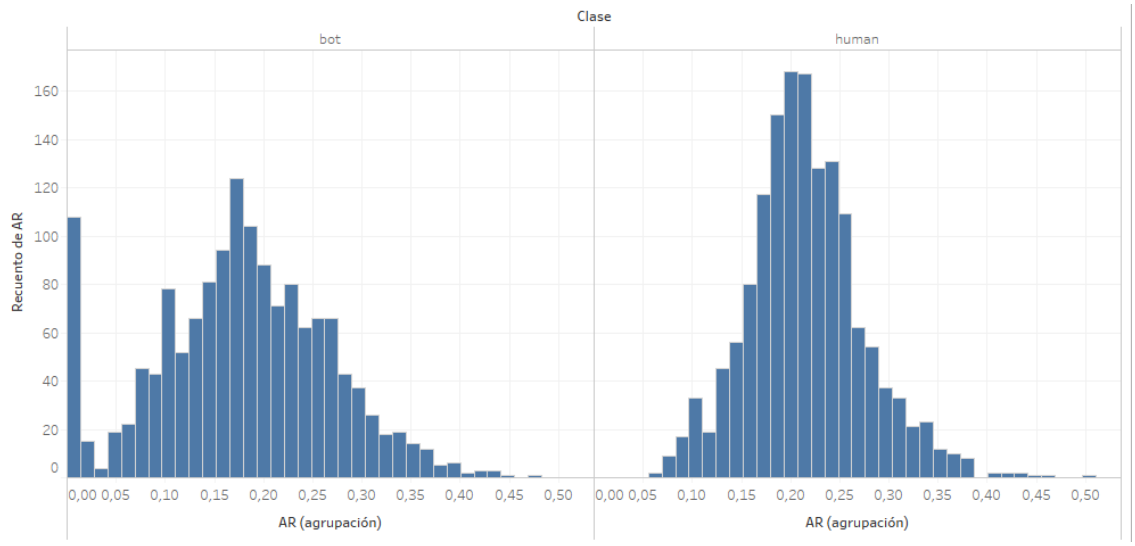

Ilustración 20: Estratos de recuento rango de acuerdo español

Obtenemos la misma gráfica que para el rango de contradicción. Esto es debido a que nuestro evaluador de frases para español es diferente al de inglés. Este evaluador solo da como resultado un solo dato en una escala de 0 a 1 por lo que se toma como nivel de positivismo de la frase. El nivel de negativismo de esta frase será el inverso al dado 1 – valor obtenido. Por el contrario, el evaluador de inglés separa los valores de las dos características lo cual es más interesante.

- Rango de disonancia

Clase	
bot	7,603
human	6,176

Tabla 10: Media de rango de disonancia inglés

Esta característica es derivada de las dos anteriores y presenta una media inversa. Los bots tienen un promedio mucho superior de disonancia como se puede observar en la tabla 10. Seguidamente observarnos la gráfica de estratos de recuento en la ilustración 21:

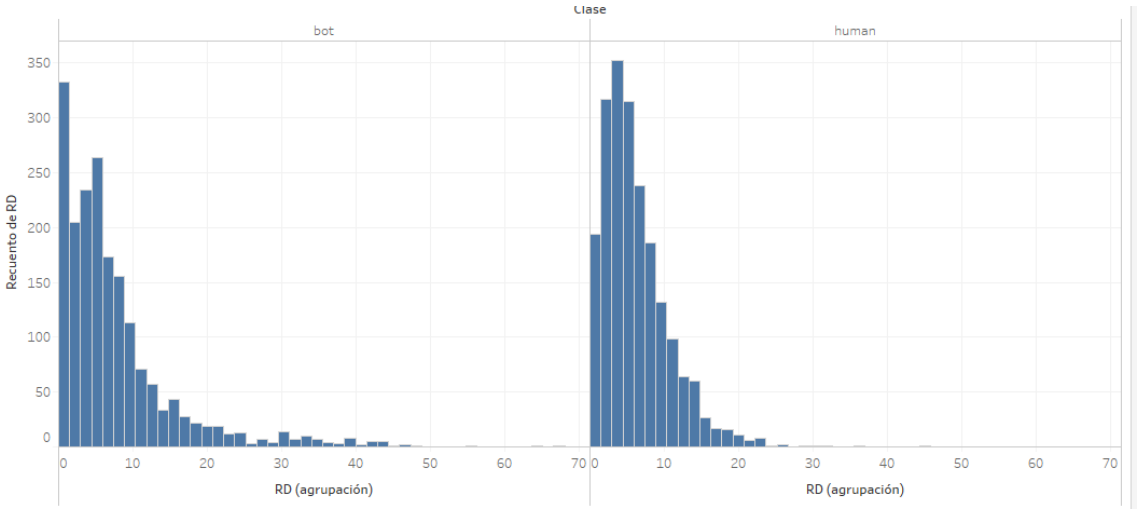

Ilustración 21: Estratos de recuento rango de disonancia inglés

Podemos observar la característica de valores nulos los cuales están un poco más emparejados. Similar a lo detallado anteriormente, los valores más extremos de

disonancia son alcanzados por los bots, aunque por pocos de ellos. Por el contrario, la mayoría de humanos se agrupan en 3 estratos y no alcanzan valores extremos.

Analizaremos ahora los resultados para español:

Clase	
bot	28,9310
human	29,7433

Tabla 11: Media rango de disonancia español

La media de disonancia cambia enormemente en español, no por lo valores ya que ellos están influenciados por la escala utilizada por el evaluador, si no por el acercamiento de promedios, observador en la tabla 11. Representaremos el recuento en estratos en la ilustración 22:

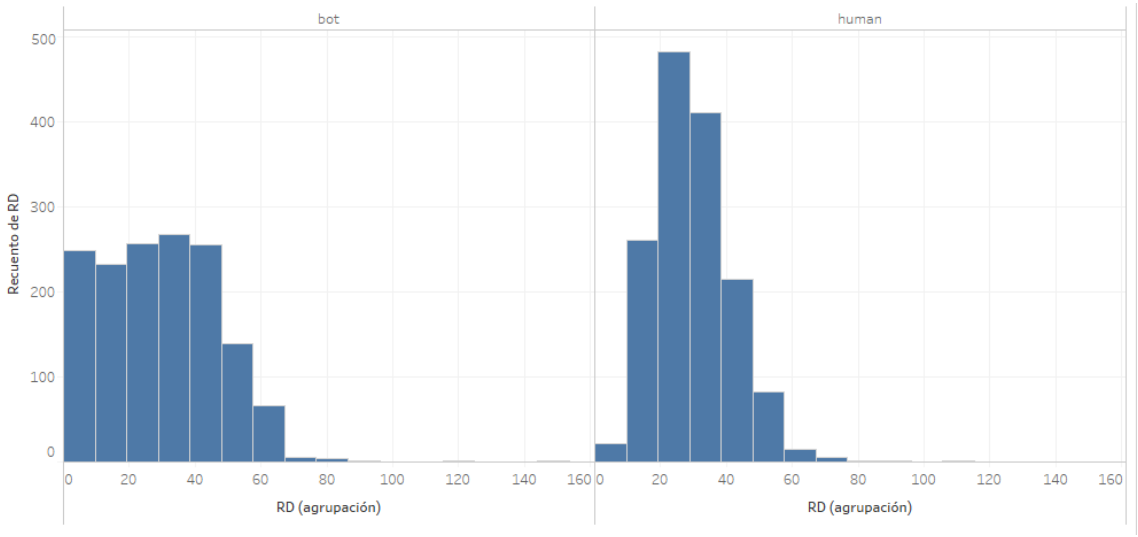

Ilustración 22: Estratos de recuento rango de disonancia español

Tendríamos unas gráficas similares entre los dos idiomas.

Podremos reafirmar dos hipótesis anteriores y concluir:

- Los valores de acuerdo y contradicción tienen un mayor promedio en los usuarios humanos.

- En los rangos de disonancia encontramos valores diferentes para cada idioma. Para inglés el promedio el liderado por los bots mientras que en español por los humanos con una menor diferencia.
- Los valores más extremos son alcanzados por los bots y los muchos de ellos obtienen un valor 0.

Podemos concluir que la mayoría de usuarios humanos expresan algún tipo de opinión con respectos a su contenido. En contra punto, los bots pueden tener valores de disonancia mayores, pero hay un gran Número que no expresa ningún tipo de opinión.

- Media de fuerza positiva y negativismo

Clase	Prom. NEG	Prom. POS
bot	0,11817	0,10831
human	0,12778	0,14176

Tabla 12: Media de positivismo y negativismo inglés

El promedio de positivismo y negativismo es mayor en los humanos observado en la tabla 12. Por lo que se puede observar el nivel de negativismo general es mayor que el de positivismo.

Resultados para español:

Clase	Prom. POS	Prom. NEG
bot	0,1042	0,3025
human	0,0611	0,2743

Tabla 13: Media de positivismo y negativismo español

Se puede observar que los valores para español son simétricos por el aspecto del evaluador anteriormente comentado observado desde la tabla 13. Sin embargo, los valores de positivismo son inversos en este dataset, los humanos tienen una menor media.

- Flip Flop de sentimientos

En la ilustración 23 podemos observar las medias de Flip Flop para inglés:

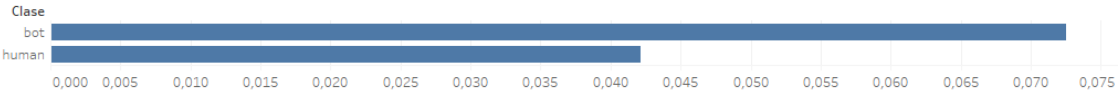

Ilustración 23: Diagrama de barras media Flip Flop inglés

Para español en la ilustración 24:

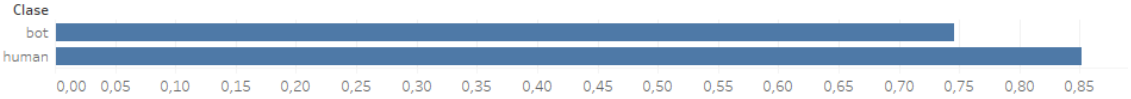

Ilustración 24: Diagrama de barras media Flip Flop español

La media es diferente lo cual sigue la tendencia con las variables anteriores.

Haciendo referencia al tercer articulo analizado, se concluía que el Flip Flop de sentimientos es menor en los bots que en los humanos, podemos verlos en la ilustración 25:

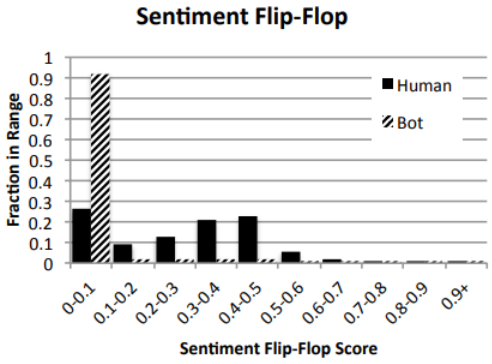

Ilustración 25: Diagrama de barras Art.3

Podemos ver los diferentes segmentos en la ilustración 26:

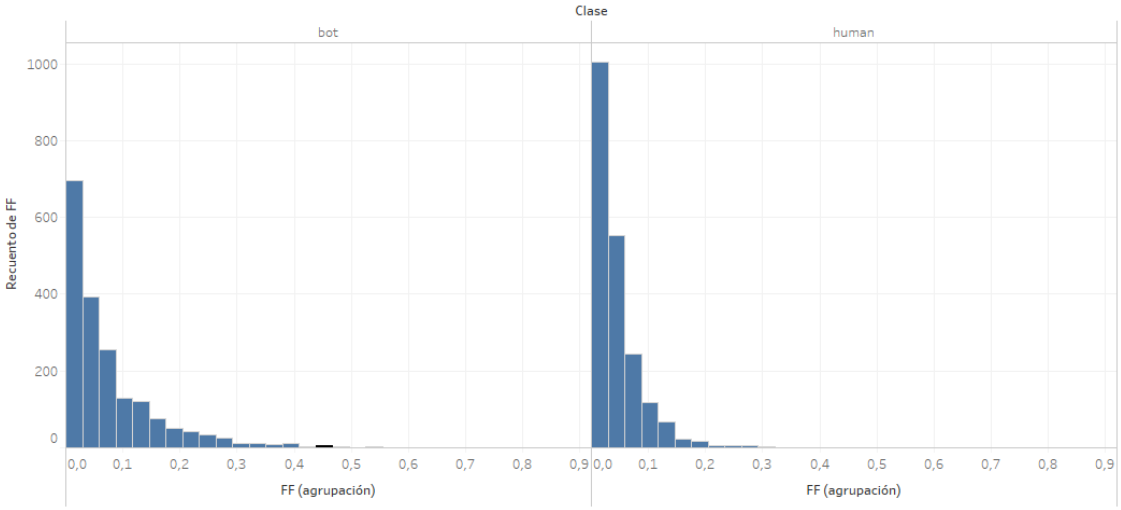

Ilustración 26: Estratos de recuento Flip Flop inglés

Los valores para español están representados en la ilustración 27:

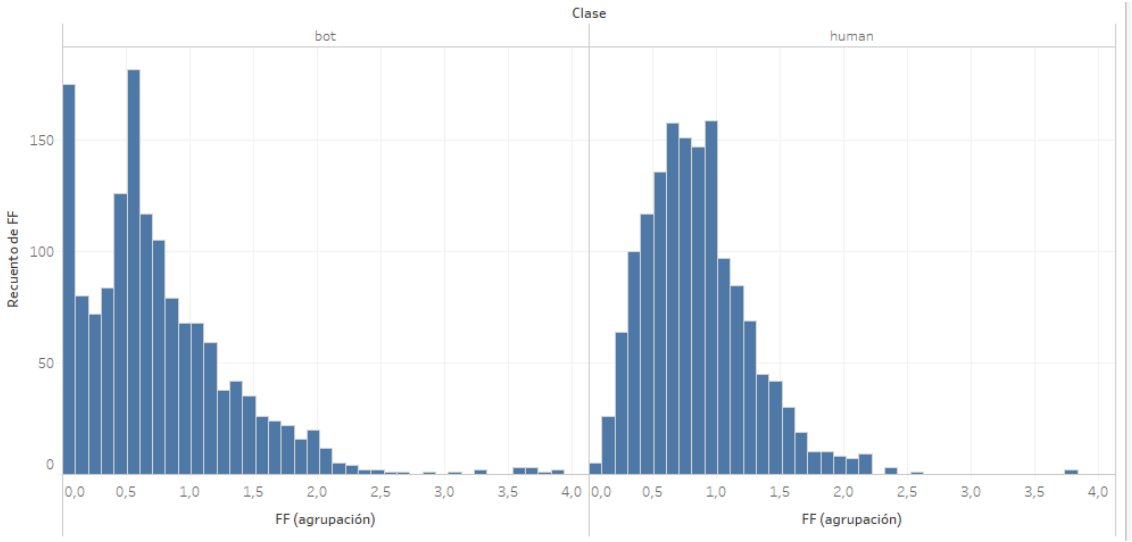

Ilustración 27: Estratos de recuento Flip Flop español

Tal y como se mostraba en la gráfica extraída del artículo en los dos estudios los bots son los usuarios que mayoritariamente no tienen esta característica, es decir, con un valor 0. En español se da la misma conclusión que en nuestro artículo de referencia,

los humanos tienen un mayor promedio, sin embargo, en inglés son los bots los que destacan con una gran diferencia.

Concluimos que todas las características referentes a la semántica están ligadas al evaluador utilizado en cada idioma. Para nuestros selectores de características podrá ser más interesante el evaluador de inglés ya que presenta una mayor entropía al evaluar positivismo y negativismo de forma diferente.

A continuación, vamos analizar las características sintácticas:

- Referentes al contenido del tuit

En la siguiente gráfica representa el promedio de cada una de las características:

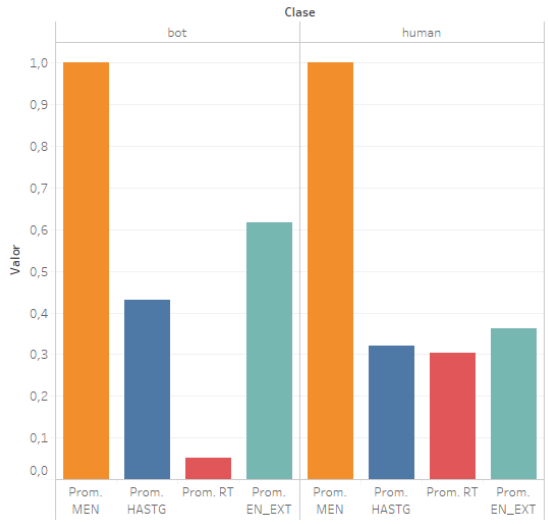

Ilustración 28: Diagrama de barras medias contenido tuit inglés

Observamos en la ilustración 28, que la media de menciones es la misma para bots que para humanos. La cantidad de hashtags es parcialmente mayor para los bots. La cantidad de retuits es mucho mayor en los humanos que en los bots y el número de enlaces externos es inverso; es mayor en bots que en humanos.

Para español en la ilustración 29:

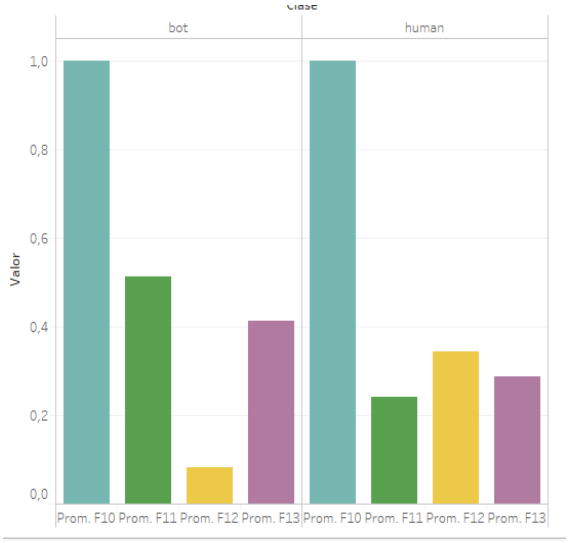

Ilustración 29: Diagrama de barras medias contenido tuit español

Podemos ver como las características repiten los mismos patrones que para inglés. Los bots superan a los humanos en Número de hashtags y enlaces externos, por el contrario, los humanos superan en retuits a los bots.

- <u>Diversidad Léxica</u>

En la tabla 14 podemos observar la media de diversidad por clase:

Clase	
bot	6,0944
human	5,6748

Tabla 14: Media diversidad léxica inglés

La media de diversidad léxica es bastante parecida entre las dos clases, representada en la tabla 14. Seguidamente podemos observar, en la ilustración 30, la medida por conteo y valor.

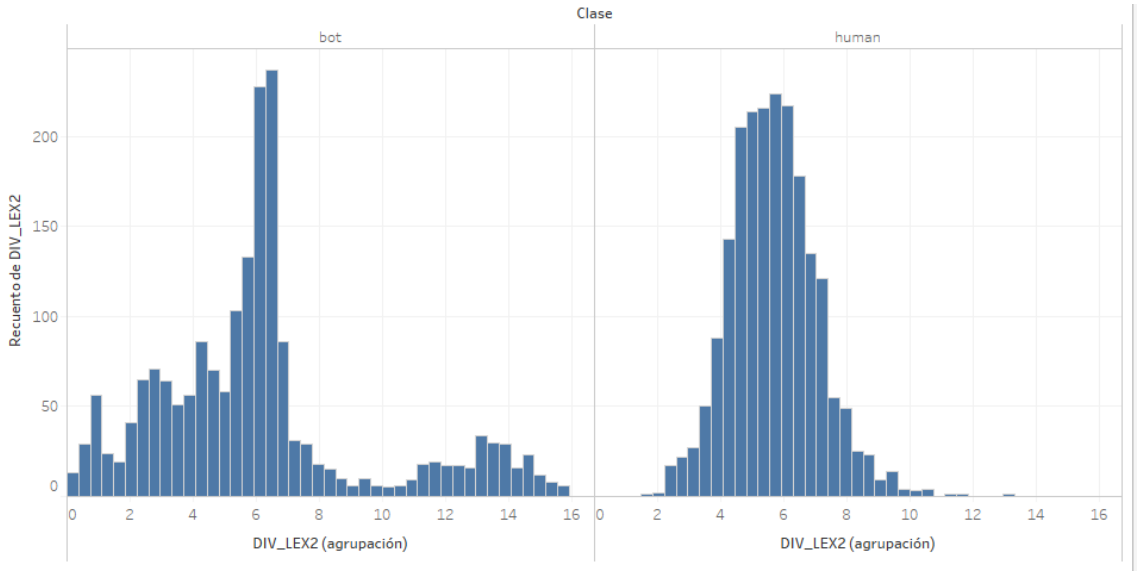

Ilustración 30: Estratos por conteo diversidad léxica inglés

Se representa una agrupación de los bots en dos grupos: el primer grupo con una diversidad léxica media, donde se encuentran la mayoría de cuentas y el segundo con una diversidad léxica mucho mayor o mucho menor y con pocos usuarios. Por parte de los humanos la gran mayoría de usuarios tienen una diversidad léxica parecida y no llegan a tener un valor tan grande.

Valores para español, descritos en la tabla 15:

Clase	
bot	4,8280
human	5,2480

Tabla 15: Media diversidad léxica español

Por el contrario, en español la diversidad léxica es mayor en los humanos.

Seguidamente podemos observar la medida por conteo y valor en la ilustración 31.

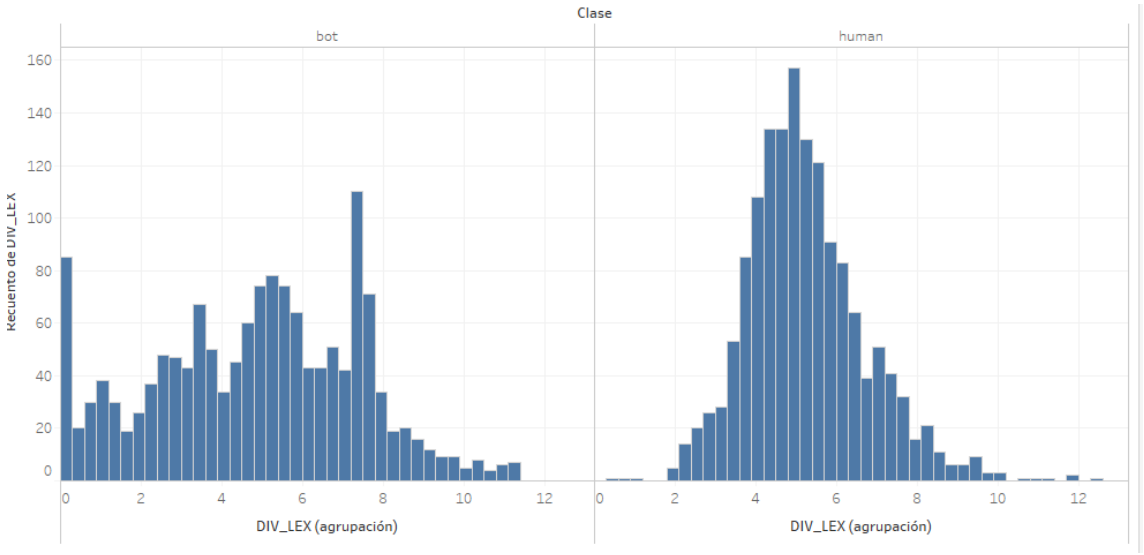

Ilustración 31: Estratos de recuento diversidad léxica español

En esta característica también podemos observar el patrón de los valores 0 ya que los bots son los únicos que han tenido una diversidad léxica de 0. Como visión general, los bots reparten más su diversidad léxica y alcanzan valores más extremos y los humanos se agrupan más en los mismos estratos.

Para un mayor análisis tendremos que analizar las dos últimas características sintácticas.

- Extensión de tuit

El promedio de extensión podemos observarlo en la tabla 16:

Clase	
bot	17,008
human	14,024

Tabla 16: Media extensión tuit inglés

El promedio de extensión es mayor en los bots que en los humanos. Adjuntaremos la visualización por estratos en la ilustración 32.

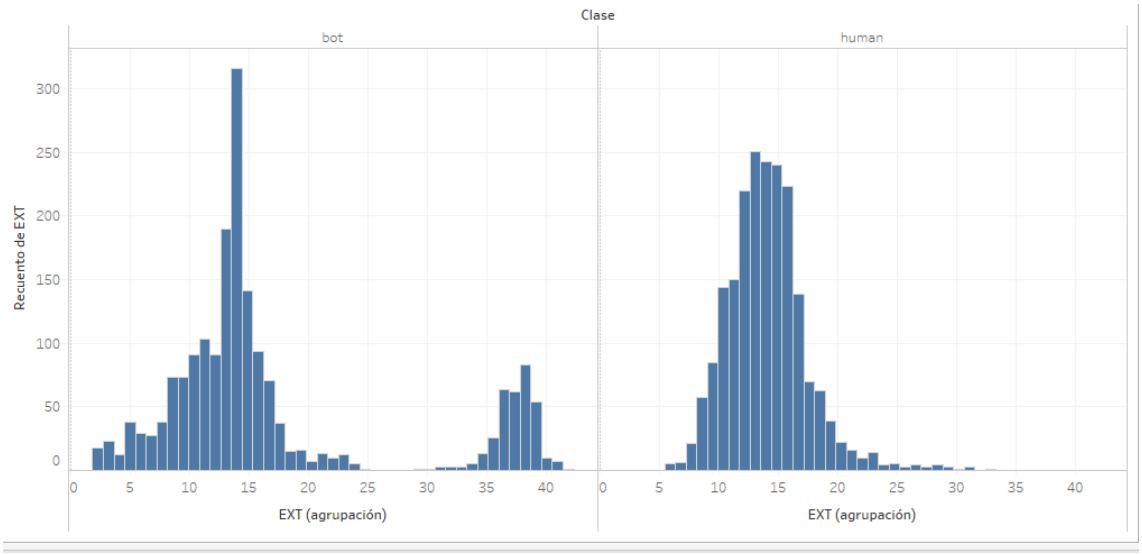

Ilustración 32: Estratos de recuento extensión de tuit inglés

La gráfica es muy similar a la adjunta de diversidad léxica por lo que podemos deducir están ligadas. Para un mayor análisis utilizaremos la última característica.

Resultados para español, representados en la tabla 17:

Clase	
bot	13,0454
human	12,8463

Tabla 17: Media extensión de tuit español

Igual que en inglés los bot tienen una mayor extensión, pero hay una menor diferencia entre las clases. Observaremos en la ilustración 33 su representación en estratos:

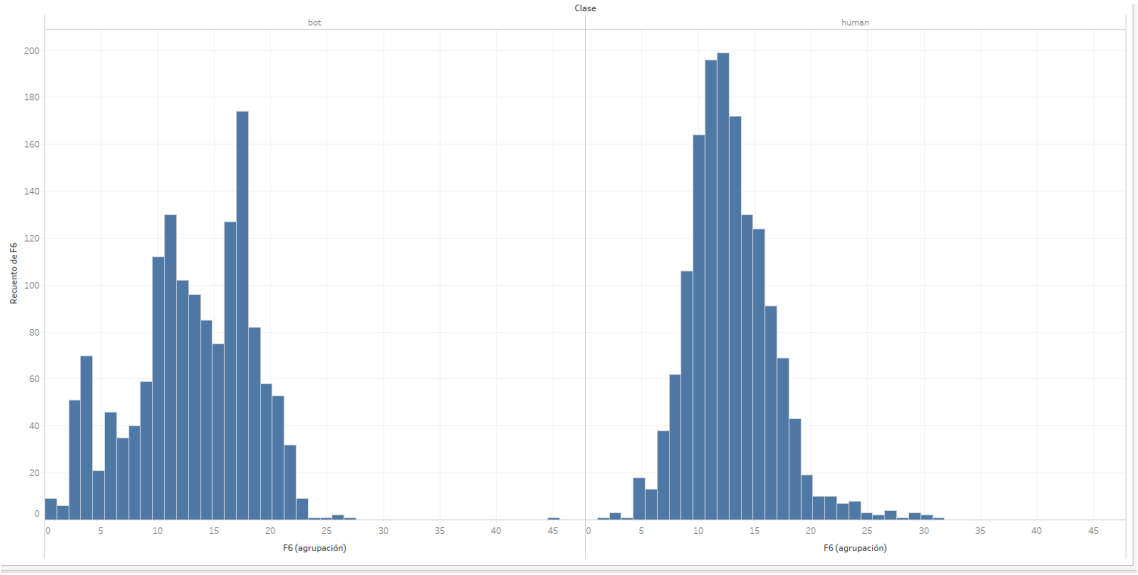

Ilustración 33: Estratos de recuento extensión de tuit español

Por lo que podemos ver en la gráfica las dos clases son bastante parecidas, aunque estén organizadas de forma diferente. Como diferencia entre idiomas la extensión es mayor inglés por parte de los bots y no presentan dos grupos de valores. Además, la extensión está más igualada en español.

- Repetición de tuit

El diagrama de promedio de repetición se muestra en la ilustración 34:

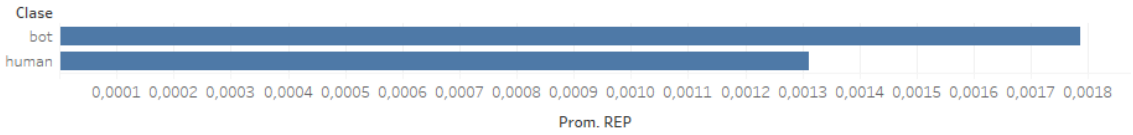

Ilustración 34: Diagrama de barras media repetición de tuit inglés

Podemos observar una gran diferencia entre las clases, los bots obtienen un promedio mucho mayor.

Datos para español representados en la ilustración 35:

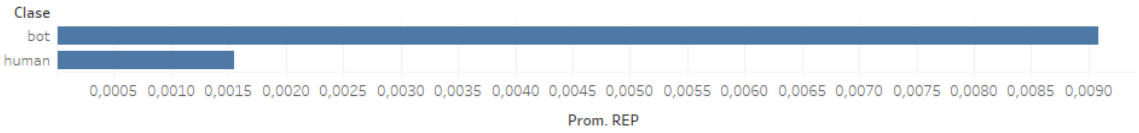

Ilustración 35: Diagrama de barras media repetición de tuit español

La diferencia es aún mayor.

Concluyendo sobre todas las características referidas al contenido podemos ver como los bots superan en 5/7 aspectos a los humanos.

- <u>Análisis sobre emociones</u>

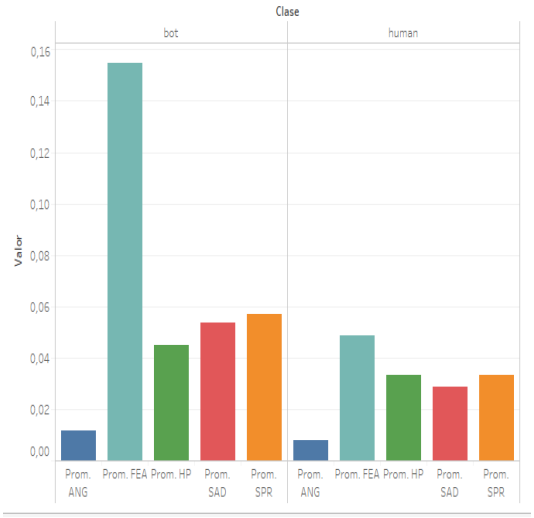

Ilustración 36: Diagrama de barras media emociones

En esta gráfica, ilustración 36, podemos observar la suma de valores recogidos por cada una de las emociones estudiadas, en ella aparece una gran diferencia entre diferentes emociones, pero destaca el miedo, el cual tiene un valor mucho mayor. Puede que esta característica sea buena para los modelos siguientes. Estas características son únicas de este idioma.

Como reflexión general sobre el estudio de características:

- Los bots difieren y destacan en numerosas características y son parecidos en muy pocas a los humanos.
- Muchos de los bots utilizados en el dataset no expresan ningún tipo de sentimiento, pero si generan contenido, lo que puede llevarnos a la creencia de que pueden ser simplemente spamers de información.

- En los bots que expresan un sentimiento u opinión cambian de parecer más frecuentemente que los humanos.
- El análisis de la fuerza positiva o negativa de los bots está fuertemente ligado a las herramientas utilizadas para ello. Además, hay aspectos como la ironía o las negaciones, aspectos ligados al lenguaje y a la personalidad del usuario que pueden cambiar completamente el sentido del contenido expresado.
- En los aspectos referidos al contenido, la filosofía de creación de los bots hace que tengan tendencia a utilizar muchos más hashtags y enlaces externos. Esto puede ser debido a que los hashtags atraer la atención de los usuarios al contenido del bot y él los redirige hacia lugares externos con algún fin. Estos aspectos pueden ser también causa de los bots de tipo spam.
- La repetición de tuit es mucho mayor en bots que en humanos, sin embargo, los bots tienen una mayor diversidad léxica que los humanos.
- Sobre las emociones, hay una gran diferencia en el cálculo del miedo lo cual puede ser interesante para la selección de características y creación de modelos.

3.8. Definición de algoritmos y selectores

La realización unísona de estas dos tareas se produce debido a que los selectores configurarán entradas para los algoritmos los cuales configurarán modelos. Estos componentes estarán recogidos en un mismo proceso.

Seguidamente procedemos a la creación de un script que tenga como objetivo la selección de características de forma automática y el entrenamiento de diferentes modelos con diferentes algoritmos.

Los algoritmos son elegidos desde los conocimientos aprendidos en todas las asignaturas relacionadas con la especialización de tratamiento inteligente de información. Además, se han incluido algoritmos que han sido utilizados en los artículos analizados anteriormente.

Los selectores aprendidos desde la rama no son compatibles con el software utilizado, pero nos dan un punto de partida para hacer una búsqueda y recopilación de diferentes selectores populares para aprendizaje automática desarrollados en Python extraídos desde diversas fuentes y foros.

En primer lugar, vamos a definir los diferentes selectores de características a utilizar:

- Defecto: Utilizando todas las características disponibles
- Sklearn → Select K best
    - Utilizando chi2 con K =14 / K = 8
    - Utilizando ftest con K = 14 / K = 8
    - Utilizando Mutual Info con K = 14 / K = 8
- Sklearn → Logistic Regression
- Selección de características con Random Forest
- Linear Support Vector Classification (SVC)
- Selección con Extra Trees Classifier
- Selección por cálculo de Varianza

Segundo lugar, definiremos los diferentes algoritmos a utilizar:

- Logistic Regression
- Linear Discriminant Analysis
- K Neighbors Classifier
- Decision Tree Classifier
- Gaussian Naive Bayes
- Support Vector Machine (SVM)
- Random Forest
- Bagging Default
- Bagging with SVM
- Bagging with Random Forest
- Multilayer perceptrón
- Adaboost

3.9. Entrenamiento de modelos y realización de experimentos

En primer lugar, se realizará un análisis TF-IDF de los documentos y se introducirá en los algoritmos seleccionados para poder establecer un punto de partida de mejora, estos modelos serán solo temporales y no se incluirán en el proyecto final.

De entre todos los modelos calculados el más destacado se puede observar en las tablas 18 y 19, inglés y español respectivamente:

```
RF: 0.526958(0.0473042)
0.5269581522525789 3[[102  95]
 [  110 86]]              precision    recall  f1-score

         bot       0.52      0.50      0.50       197
       human       0.51      0.52      0.52       196

    accuracy                           0.53       393
   macro avg       0.53      0.53      0.53       393
weighted avg       0.53      0.53      0.53       393
```

Tabla 18: Resultados TF-IDF inglés

```
BG: 0.472563 (0.527437)
0.4325631233487843[[74  82]
 [ 67 75]]              precision    recall  f1-score

         bot       0.49      0.44      0.51       156
       human       0.45      0.49      0.43       142

    accuracy                           0.47       298
   macro avg       0.47      0.46      0.47       298
weighted avg       0.47      0.46      0.47       298
```

Tabla 19: Resultados TF-IDF español

Concluido el establecimiento del punto de partida se analizarán los modelos con las características extraídas.

Cada uno de los procedimientos de selección de características será probada con todos los algoritmos. Los resultados de todos los diferentes modelos y selectores están adjuntos en los archivos del proyecto. Los más destacados de todos ellos son:

Para inglés:

Características

Las características más elegidas por todos los selectores es el número de menciones con una elección de 9/9 y la Número de retuits y enlaces externos con una elección de 7/9 de ellos.

La característica menos elegida es la repetición de tuits con una selección de 1/8 de los selectores. Esta selección se debe a la gran diferencia entre clases tal y como se estudió en los apartados anteriores.

Con respecto a las características de positivismo y negativismo no destacan como elegidas, pero se prioriza el positivismo al negativismo en una relación 4/9 a 2/9.

La característica de flip flop tiene una elección baja 4/9.

La emoción más elegida es el miedo ya que obtiene una relación 6/9 y tiene una gran diferencia con el resto.

Selectores

Los selectores menos restrictivos son: Chi2, Ftest y Mutual Info. con una selección de 14 características.

Los selectores de características más restrictivos son Ext. Tree y eliminación recursiva con una selección de 3 características:

- Ext. Tree
    - Número de menciones
    - Número de retuit
    - Número de enlaces externos
- Eliminación recursiva
    - Rango de contradicción
    - Número de menciones
    - Nivel de miedo

Según los valores de precisión de modelos el selector más eficiente es Varianza con un promedio de 92.6 % y el selector menos eficiente es Eliminación Recursiva con 89.75%.

Modelos

El máximo de precisión alcanzado por un modelo es 97% y el mínimo alcanzado es 73%.

Los algoritmos con un promedio de precisión mayor son:

- Random Forest con una media de 95.3 %
- BG-RF con una media de 95%

Las configuraciones por defecto en las cuales se cogen todas las características obtenemos una precisión del 97% con un algoritmo de Random Forest y Bagging.

Los resultados para los selectores más restrictivos son:

- Ext. Tree
    - Se obtienen varios modelos con un porcentaje del 94% (KNN, RF, BG, BG-RF)
- Eli. Recursiva
    - Se obtienen varios modelos con un porcentaje del 91% (LR, SVM, BG-SVM, MLPC, ADA)

Para el selector más eficiente, Varianza se alcanza el máximo de precisión en 3 de sus modelos y alcanza el máximo de algoritmo en 6/12.

Concluimos que nuestro mejor modelo es un algoritmo de Random Forest con un selector Varianza con las características:

- Rango de acuerdo
- Rango de disonancia
- Extensión de tuit
- Diversidad léxica
- Número de menciones
- Número de hashtags
- Número de retuits
- Número de enlaces externos
- Nivel de miedo

Parámetros de salida representados en la tabla 20:

```
RF: 0.970304 (0.007412)
0.9669211195928753[[189   8]
 [  5 191]]              precision    recall  f1-score

         bot       0.97      0.96      0.97       197
       human       0.96      0.97      0.97       196

    accuracy                           0.97       393
   macro avg       0.97      0.97      0.97       393
weighted avg       0.97      0.97      0.97       393
```

Tabla 20: Resultados mejor modelo inglés

Para español:

Características

Las características más seleccionadas son: número de retuits con una selección de 9/9 y el número de enlaces externos y Número de hashtags con una selección de 7/9.

La característica menos elegida es el número de menciones con una selección de 1/9, lo cual es totalmente contrario al inglés la cual es la más elegida. En este idioma, el nivel de positivismo tiene una mayor selección que el negativismo, 4/9 a 3/9.

En comparación al otro idioma, la repetición de tuit consigue un valor mayor 3/9 pero siguen encontrándose entre las peores.

A lo referido al flip flop la selección en español es mayor 6/9 mayor que en inglés. La diferencia entre las medias no es muy grande y la única diferencia era la cantidad de humanos con un flip flop 0, por lo que no podemos concluir claramente a que es debido la selección.

La diversidad léxica obtiene el mismo valor de selección entre los dos idiomas 6/9.

Selectores

Los selectores menos restrictivos son Chi2, Ftest y Mutual Info. con una selección de 9 características.

El selector más restrictivo es Ext. Trees con solo dos características elegidas:

- Número de retuits
- Número de enlaces externos

El selector más eficiente es según los valores de precisión de sus algoritmos es Mutual Info. con una precisión media de 85.83%. El menos eficiente es la Eliminación Recursiva con un 81.08%

Modelos

El máximo de precisión alcanzado por un algoritmo es 93%.

Las configuraciones por defecto alcanzan una precisión del 93% con un algoritmo de BG-RF.

El selector más restrictivo consigue una precisión máxima de 87% por un algoritmo de Adaboost.

El algoritmo con un promedio de precisión más alto es BG-RF con un 89.77%, el menos es BG-SVM con una precisión de 76%.

El modelo más eficiente es la relación cantidad de características y precisión es un algoritmo BG con una selección de Varianza con las siguientes características:

- Rango de disonancia
- Nivel de negativismo
- Extensión de tuit
- Flip Flop de sentimientos
- Diversidad léxica
- Número de hashtags
- Número de retuits
- Número de enlaces externos

Parámetros de salida mostrados en la tabla 21:

```
BG: 0.915672 (0.017197)
0.9261744966442953[[143  13]
 [  9 133]]              precision    recall  f1-score

         bot       0.94      0.92      0.93       156
       human       0.91      0.94      0.92       142

    accuracy                           0.93       298
   macro avg       0.93      0.93      0.93       298
weighted avg       0.93      0.93      0.93       298
```

Tabla 21: Resultados mejor modelo español

3.10. Conclusiones

Después de todo el estudio y desarrollo de modelos hemos podidos observar:

Con respecto a los datos:

Los datos de los datasets desde un principio estaban divididos en idiomas, lo cual planteaba un problema ya que cada idioma debía ser tratado particularmente he iba a influir en todos los aspectos que derivasen del mismo. Una consecuencia de ello ha sido la elección de bibliotecas y scripts auxiliares para análisis y evaluación los cuales provienen de fuentes distintas y tratan los mismos aspectos como por ejemplo el nivel de positivismo y negativismo desde diferentes puntos de vista y los evalúan de diferentes formas.

Con respecto a la programación:

Python es un lenguaje simple y muy eficiente además de útil. Pero aspectos ajenos al lenguaje han sido determinantes en el cálculo de valores de características. Un ejemplo es el tratamiento de todo el contenido de tuits en forma minúscula lo cual

cambio completamente los valores en versiones donde se implantó. Con ello concluimos que todas las bibliotecas utilizadas en un proyecto como tal deben ser compatibles, es decir, los enlaces, parámetros o métodos de cada uno deben estar correctamente enlazados con los mejores tipos de datos requeridos.

Con respecto a las características:

Muchas de las características que hemos estudiado presentaban diferencias notables entre las clases como promedios o valores máximos. Además, hemos podido extraer patrones asociados a estas mismas clases

- Los valores extremos de opinión en bots ya sean de valores altos o valores 0
- El agrupamiento de humanos en los mismos estratos en diversas características sintácticas.
- En lo referido a la semántica de tuit; los bots utilizan más herramientas y no texto plano, es decir, añaden hashtags y enlaces externos con el fin de atraer y dirigir a sus objetivos.
- La mayor diversidad léxica de los bot es una característica que sorprende puesto que los humanos son más creativos que los robots por naturaleza por lo que podemos pensar que la utilización de sinónimos y antónimos en la creación de tuits podría estar automatizada y así conseguir una mayor diversidad. Con respecto a la extensión de tuit se podría dar un suceso similar con una automatización de longitud. Con respecto a la repetición, esta repetición puede estar debido a los bots de tipo spamers los cuales repiten su contenido constantemente.

Con respecto a los selectores:

De entre los 9 selectores programados el más destacado sin duda ha sido la varianza puesto que ha conseguido crear un set de características bueno eligiendo

normalmente características que en el estudio previo predecimos que serían interesante.

Hemos podido comprobar que los selectores más restrictivos han conseguido buenos resultados creando sets muy pequeños por lo que podemos concluir que hay una gran diferencia de utilidad entre las diferentes características por lo que muchas de ellas crearán ruido o añadirán poca precisión.

Con respecto a algoritmos y modelos:

En general se ha conseguido una precisión media buena. La media de precisión de todos los modelos en el idioma de inglés ha sido de 91.6% y un 83.15% para español. La diferencia entre los idiomas únicamente reside en la calidad de todas las herramientas utilizadas para tratarlos. Hemos podido comprobar que los materiales para inglés están mucho más avanzados que para español y son muchos más precisos en sus resultados, estos además han influido en los tiempos de cálculo de características puesto que los cálculos de inglés han requerido de un mayor tiempo que los de español.

Nuestro diagrama de tareas queda actualización y mostrado en la ilustración 37:

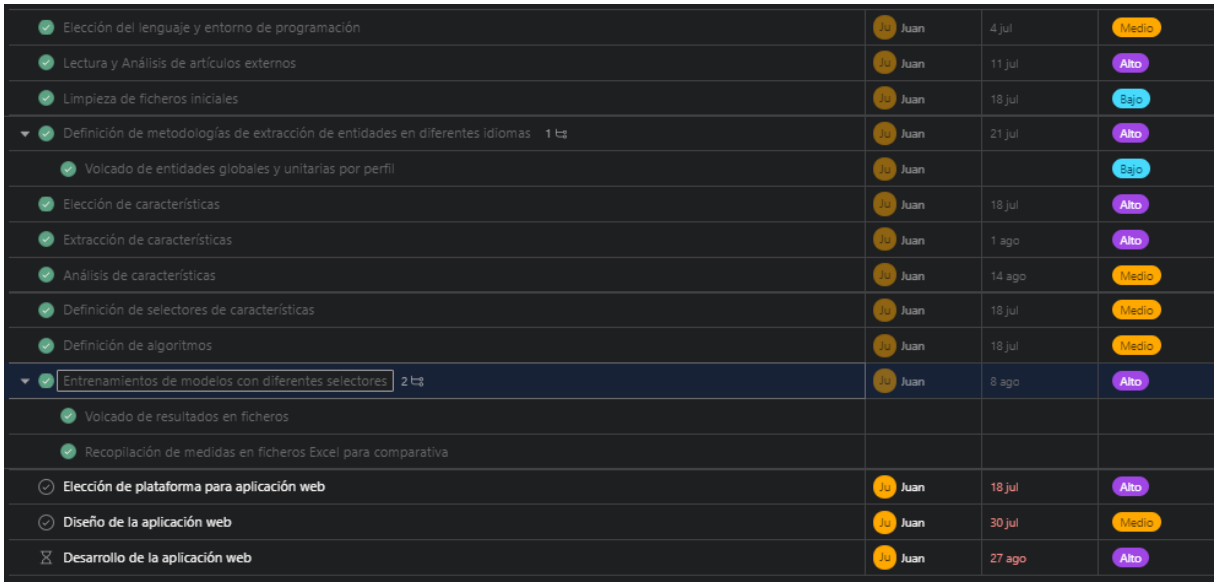

Ilustración 37: Actualización cronología de tareas 3

4. Aplicación Web

4.1. Elección de plataforma

Para completar el desarrollo completo del proyecto se propone la creación de una aplicación web donde un usuario pueda subir un archivo con tuits recopilados de un usuario y nuestros modelos puedan dan un resultado ante él.

En primer lugar, haremos un sondeo sobre las diferentes plataformas que pueden albergar una pequeña web. Como requerimiento, se necesita que la web pueda interactuar con Python y así poder ejecutar los scripts de evaluación.

Como opción más sencilla es la utilización de un sistema de gestión de contenido. Los dos más utilizados son WordPress y Joomla.

Instalamos el primer sistema, Wordpress. Para la subida y entendimiento de archivo en Python se debe instalar un plugin especial para ello y los archivos deben ser ejecutados de forma externa lo cual no es fiable. Además, como añadido los servidores FTP en los que se instala el software no funciona correctamente con los archivos Python al no admitirlos de forma segura. Debido a estos problemas se opta por abandonar esta herramienta.

La segunda opción es Joomla, pero al hacer pequeñas indagaciones concluimos que no es del todo compatible con Python y desestimamos la idea.

Por último, optamos por la opción más complicada, pero con más control y compatibilidad con Python, Django. Este software está construido es Python por lo que no habrá problemas con el lenguaje de programación. Definido nuestro entorno de desarrollo queda completa la tarea referida a la elección de entorno de desarrollo.

Realizada la instalación de Django creamos nuestra aplicación dentro del mismo para poder empezar el proyecto web. Dentro de este proyecto se nos crearan diferentes archivos que utilizaremos más tarde: formularios, modelos, urls y views.

El funcionamiento de Django es simple: desde el archivo de urls se llama a cada una de las vistas creadas y conformar con estas llamadas las diferentes urls que tendrá nuestra web. Dentro de cada vista se hace una renderización de cada página en código HTML. Además, cada vista puede gestionar parámetros, base de datos, formularios… Estas vistas nos servirán para la llamada de los scripts de limpieza y clasificación de los archivos subidos por los usuarios.

El código HTML utilizado para el proyecto puede ser creado desde cualquier editor de web por lo que utilizamos NicePage para ello. Existe un problema en la compatibilidad de estilos entre la creación de la web y la lectura de Django por lo que necesitamos un archivo puente descarado desde Boostrap de tipo CSS que consigue la aplicación de estilos. Además, se necesita editar los archivos HTML intercalando código de Django especial para poder hacer referencias a vistas externas o archivos necesarios para la visualización correcta. Podemos observar su carga en la ilustración 38:

```
{% load static %}
<!-- Bootstrap -->
<link href="{% static 'ProyectoWebApp/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
```

Ilustración 38: Carga y adición de script boostrap

4.2. Diseño

Configurada la web podemos crear nuestra web de forma simple. Puesto que nuestro proyecto web va a requerir de una funcionalidad básica el diseño no será extremadamente complejo. En primer lugar, definiremos los objetivos a alcanzar de nuestra web:

- Informar al usuario de los peligros de los bot sociales y como afectan a su vida diaria
- Presentar de forma sencilla el trabajo realizado en este proyecto.
- Diseño de una sección que explique como el usuario puede crear su propio archivo para poder subirlo a la plataforma.

- Formulario de subida a la plataforma, realización de prueba y muestra del resultado arrojado.

Definidos los objetivos a conseguir realizaremos una organización de nodos y subnodos de la web.

Debido a la simplicidad únicamente necesitaremos dos nodos principales:

- Inicio / Home: el cual introduzca al usuario desde la presentación del problema y sus causas a como este proyecto desarrollo una solución y como el propio usuario puede ponerla a prueba. Este nodo enlazara con el siguiente.

- Prueba / Test: se accederá desde el nodo principal únicamente desde la sección creada para ello. En él, el usuario tendrá un formulario de subida y tendrá la posibilidad de realizar la prueba, siempre que el archivo se tome como válido. Una vez realizado el análisis y clasificación del archivo la respuesta será mostrada en este mismo nodo.

Como objeto general se creará una barra de navegación la cual conste con un menú desde donde acceder a todas las diferentes secciones, un botón para poder acceder a los diferentes idiomas de la web y botón de vuelta a inicio.

4.3. Desarrollo

Inicio / Home

Las siguientes imágenes hacen referencia a un solo archivo HTML y se dividen las diferentes secciones de la página.

En primer lugar, una sección introductoria donde presentamos el problema y encontramos un título atractivo para el usuario. Podemos observar las diferentes secciones por idioma en español en inglés en las ilustraciones 39 y 40 respectivamente. Además, podemos observar la barra de navegación la cual queda fija en el scroll de página.

Ilustración 39: Inicio Sección 1 español

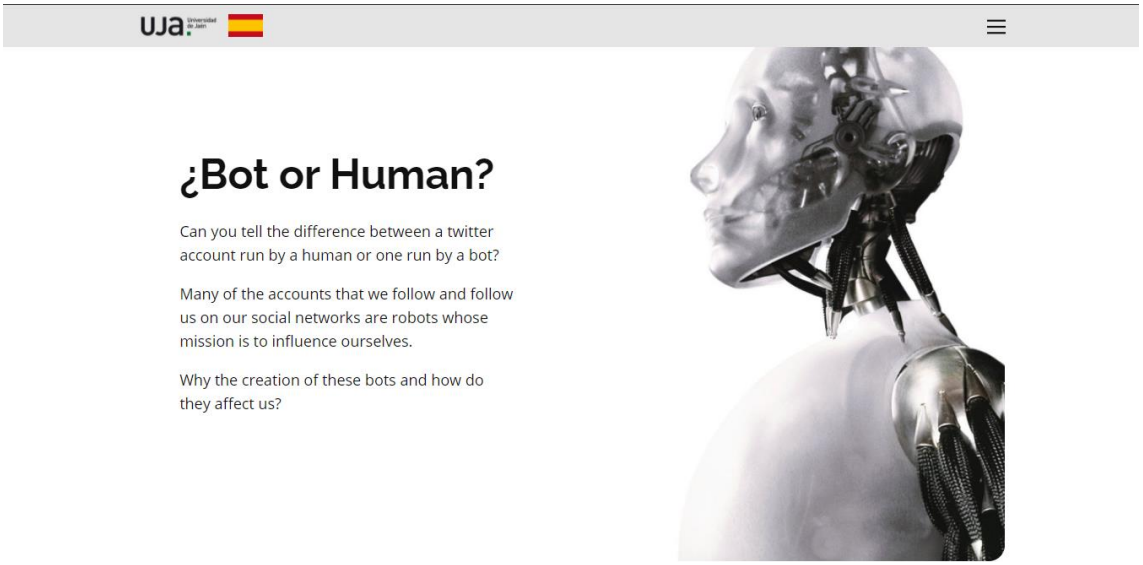

Ilustración 40: Home Sección 1 inglés

En segundo lugar, se presentan los principales objetivos de los bots sociales de una forma amena y que no sobrecargar la atención del usuario. Podemos ver las diferentes secciones por idioma en las ilustraciones 41 y 42.

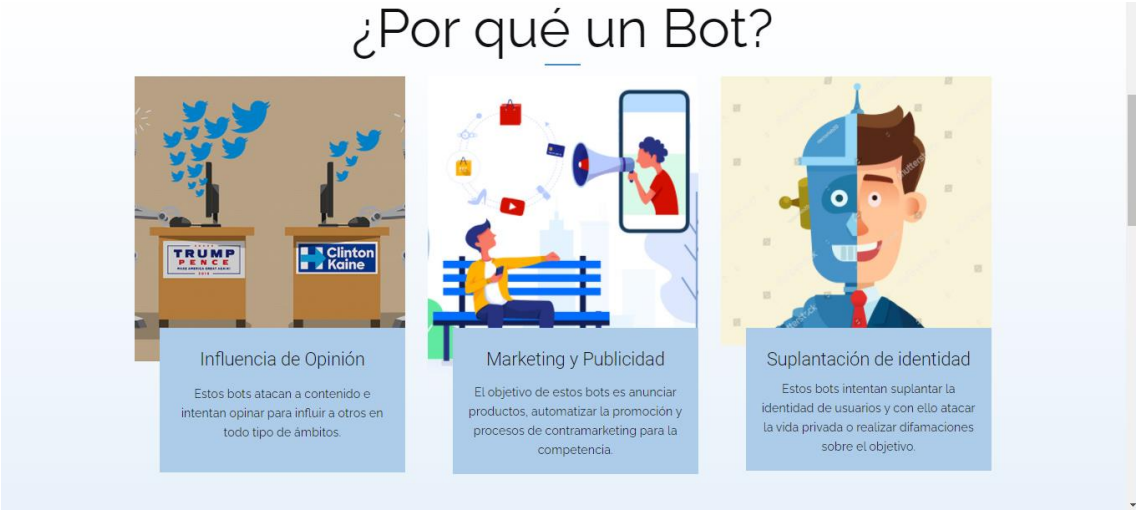

Ilustración 41: Inicio Sección 2 español

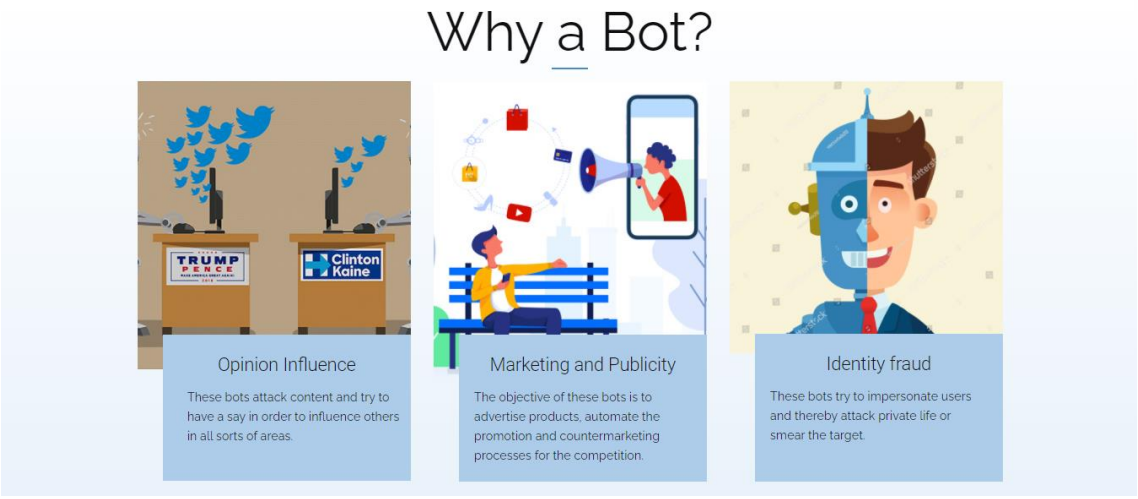

Ilustración 42: Home Sección 2 inglés

En la tercera sección se presenta un pequeño resumen del proyecto desarrollado y se introduce al usuario a realizar su propia prueba. Se observan ambas secciones por idioma en las ilustraciones 43 y 44.

Ilustración 43: Inicio Sección 3 español

Ilustración 44: Home Sección 3 inglés

Acceso a la prueba

Esta sección está incluida en Home. En ella se describe como se debe construir el archivo necesario para la realización de la prueba; se describen y se adjunta una imagen de demostración de un archivo tipo con la que se presenta al usuario la sencillez de la recopilación. Por último, destacamos un botón en el cual clicando se

accede a la página de formulario de prueba. Se pueden observar ambas secciones en las ilustraciones 45 y 46.

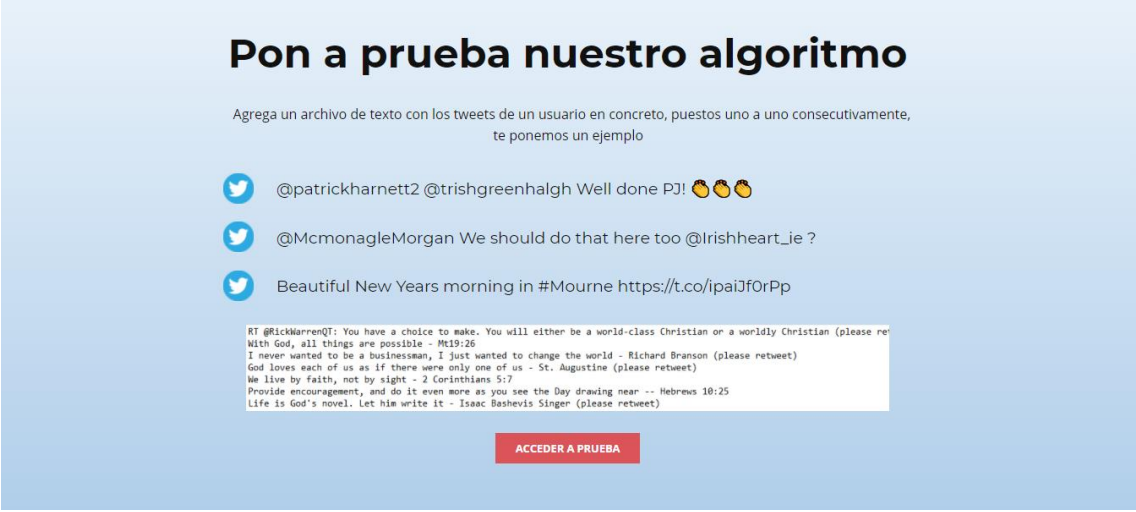

Ilustración 45: Inicio Sección 4 español

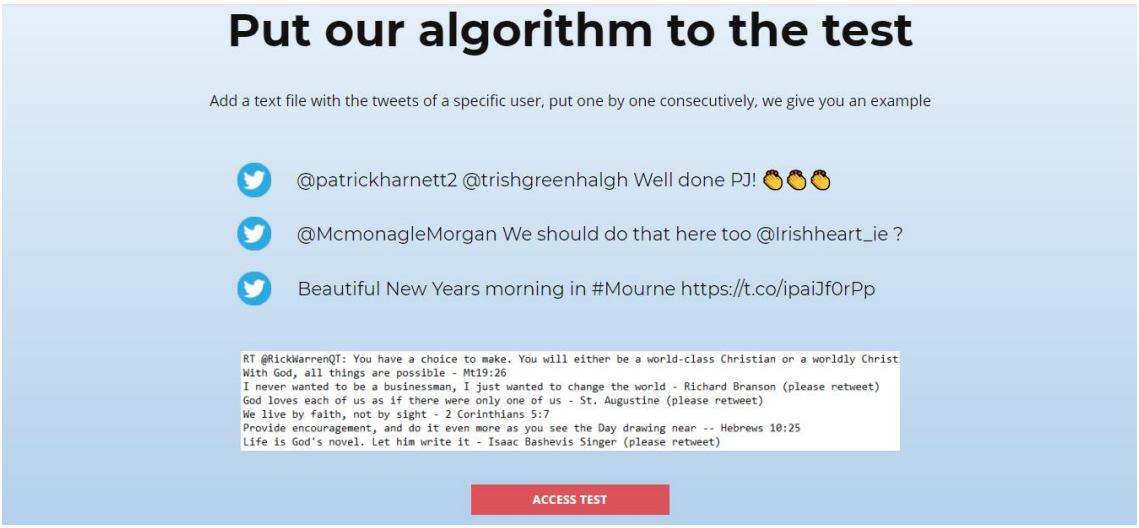

Ilustración 46: Home Sección 4 inglés

Formulario de prueba / test

En esta página podemos encontrar 3 subsecciones que describiremos de izquierda y derecha en las ilustraciones 47 y 48:

- En la primera sección se encuentra el formulario de subida de archivo. En él se debe introducir un nombre de archivo y seleccionar a través del explorador de archivos por defecto del sistema operativo el archivo a subir.
- En la segunda sección observamos un único botón en el cual se realiza la prueba, cargado el archivo este botón recargara la página y se realizara la clasificación del archivo.

Ilustración 47: Prueba Sección Prueba español

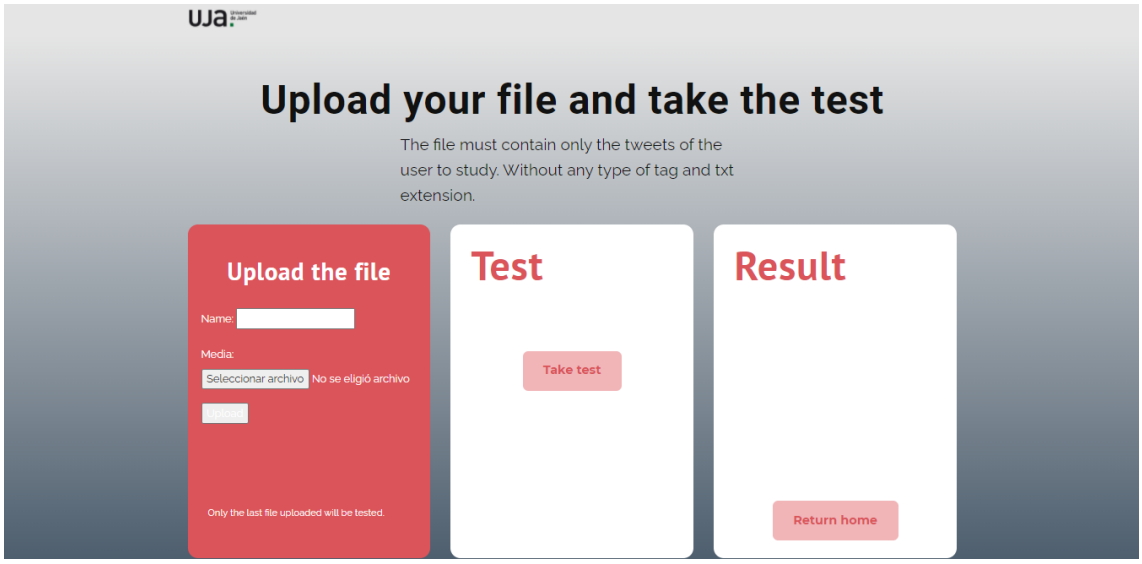

Ilustración 48: Sección Prueba inglés

- En la última sección y si la prueba se ha realizado de forma satisfactoria el resultado se muestra en esta sección y se desactiva el formulario de entrada. Podemos observar los diferentes resultados en las ilustraciones 49 y 50.

Ilustración 49: Prueba Resultado español

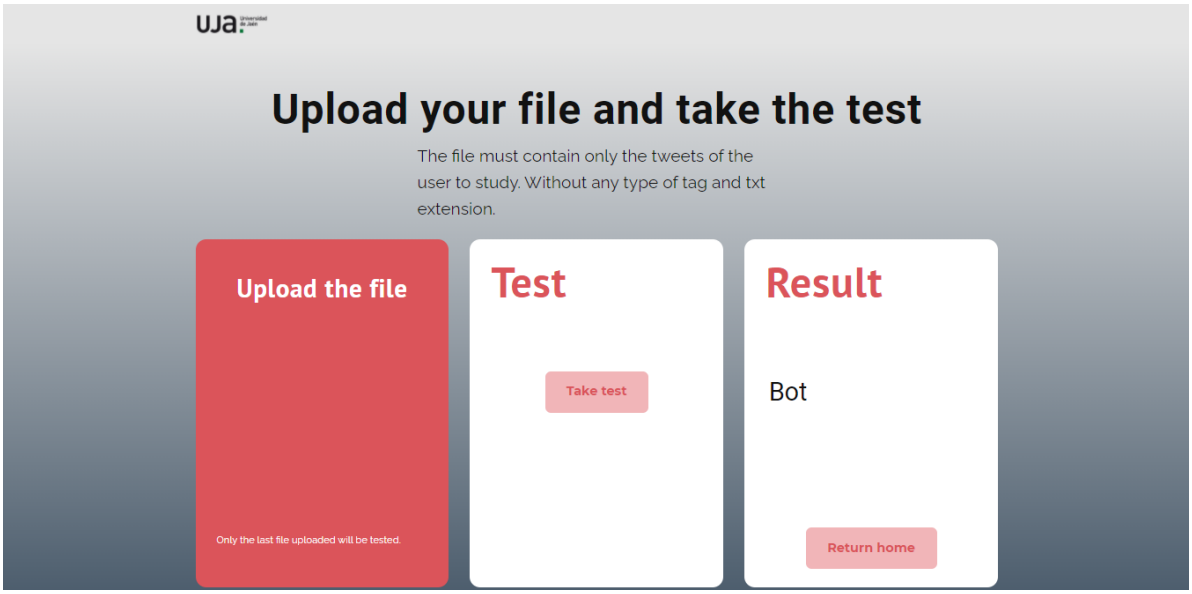

Ilustración 50: Test Resultado inglés

4.3.1 Evaluación de archivos subidos

Por último, desarrollaremos los dos scripts de evaluación para los diferentes idiomas. Para la evaluación de los archivos subidos utilizaremos una votación impar de diferentes modelos elegidos y la solución final será la decisión de la mayoría.

Para el idioma inglés utilizaremos los siguientes modelos:

- Chi2 con Random Forest
- Defecto con Bagging
- Mutual Info. con Adaboost
- Varianza con Bagging

- Varianza con Random Forest

Para español hemos seleccionado los siguientes:

- Defecto con Bagging con Random Forest
- Ftest con Bagging con Random Forest
- SVC con Random Forest
- Mutual Info. con Bagging con Random Forest
- Varianza con Bagging

Para la elección de los modelos se recogen modelos con una precisión máxima o cercana a ella y para los selectores se intenta coger una mayoría de variación de características priorizando las características más seleccionadas.

En caso de empate en alguna elección se seleccionará el selector o modelo que tenga un promedio de rendimiento mejor.

Con la finalización del desarrollo web quedan por finalizadas todas las tareas. El desarrollo completado se refleja en la ilustración 51.

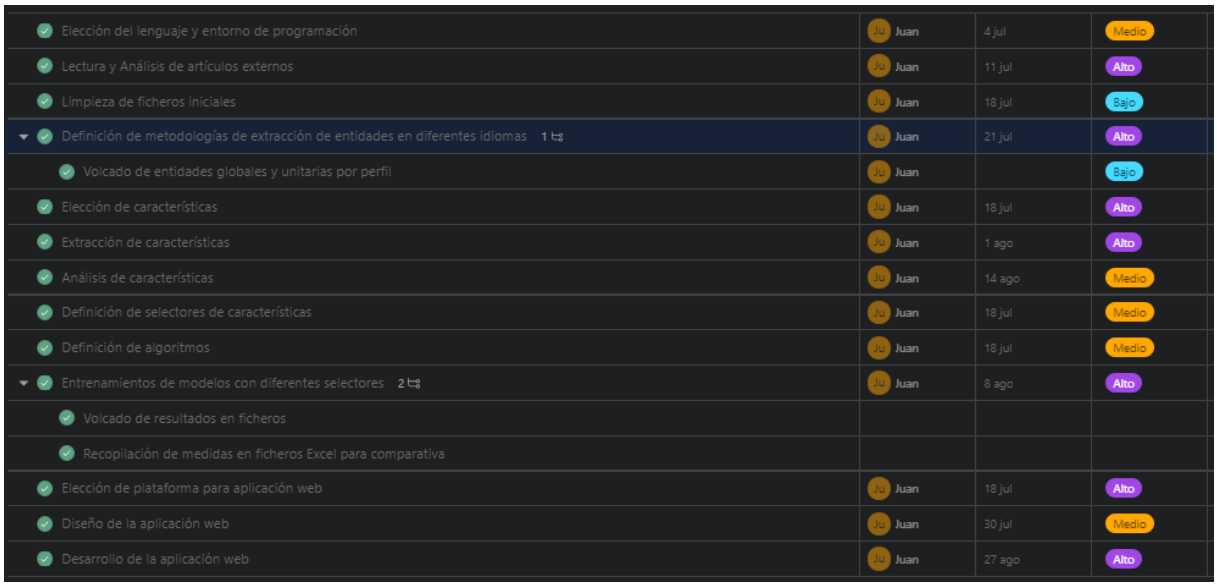

Ilustración 51: Tareas completas

5. Conclusiones Finales

De lo analizado y lo extradido:

En primer lugar, el análisis de diferentes fuentes ha sido vital para poder formar una verdadera opinión ante este gran problema que es la detección de perfiles falsos. Desde diferentes perspectivas, se ha podido abordar y se han conseguido resultados satisfactorios por parte de todos los autores, pero sin duda la aplicación de campos nuevos como los aplicados en este proyecto amplían la precisión y dinamizan las detecciones futuras. En contra parte, la inclusión de campos más complejos presenta nuevos problemas, la relación esfuerzo resultado y la complejidad de los datos etc.

En este proyecto la inclusión del lenguaje natural ha sido vital para poder extraer verdaderamente nuevo conocimiento sobre los bots y poder descubrir patrones que los defina definitivamente.

Aunque como ya ha sido mencionado, esta ciencia sigue estando ligada al aspecto más abstracto del mundo, la psicología humana.

Como mejoras y proyectos futuros:

Sin duda, hay aspectos y perspectivas que quedan fuera del desarrollo y estudio de este proyecto. La inclusión de nuevas fuentes de información en proyectos futuros podría suponer la mejora de la precisión y podrían abrir nuevos frentes de estudios que desembocasen en nuevas mejoras.

Además, la inclusión y estudio de características desestimadas por coste computacional podría suponer la mejora de precisión o inclusión de ruido.

Unos de los aspectos con mayor campo de mejora seria la aplicación y funcionalidades web las cuales podrían estar enlazadas con la propia plataforma Twitter y que el propio usuario solo tuviera que indicar que perfil quiere analizar y la web se encarga de todo el trabajo de extracción y análisis del mismo.

Del problema y su situación actual:

Los bots sociales cada día están más presentes en nuestra sociedad y son extremadamente útiles ya que el usuario medio no se centra en su detección y es guiado por ellos en muchos casos. Como herramienta de cambio de opinión son extremadamente útiles y su complejidad es cada vez mayor por lo que la dificultad de detección aumentara cada día más.

Muchas personas a día de hoy comparten el mismo propósito la detección de perfiles falsos y buscan distintas soluciones e intentas paliar este gran problema con la publicación de estudios o proyectos similares a este que realmente lleguen a la gente y puedan concienciarla.

Del proyecto y su realización:

Sin duda ha sido un gran reto en mi carrera y ha significado un culmen en mucho de los aspectos aprendidos didácticamente no solo por significar una prueba final a todo lo aprendido si no como un reto de vida que te hace replantear aspectos dentro y fuera del ámbito laboral.

Mas concretamente, la investigación realizada en los dos grandes campos de desarrollo: Minería de Datos en general y Procesamiento del Lenguaje Natural en particular ha sido muy provechosa puesto que significa una gran ampliación de conocimientos y por mi pasión a ellos una gran satisfacción personal.

En una autocrítica este proyecto particularmente me ha servido para aumentar el nivel de organización de desarrollo de proyectos informáticos y en general, aumentar la profesionalidad de mi trabajo sea en el ámbito que sea.

Con respecto a los tutores asignados destacar que su profesionalidad que es directamente proporcional a su dedicación y que han sido todo un ejemplo a seguir.

Bibliografía

[1] Daelemans, W., Kestemont, M., Manjavacas, E., Potthast, M., Rangel, F., Rosso, P., ... & Zangerle, E. (2019, September). Overview of PAN 2019: bots and gender profiling, celebrity profiling, cross-domain authorship attribution and style change detection. In *International Conference of the Cross-Language Evaluation Forum for European Languages* (pp. 402-416). Springer, Cham.

[2] Efthimion, P. G., Payne, S., & Proferes, N. (2018). Supervised machine learning bot detection techniques to identify social twitter bots. *SMU Data Science Review*, *1*(2), 5.

[3] Puertas, E., Moreno-Sandoval, L. G., Plaza-del-Arco, F. M., Alvarado-Valencia, J. A., Pomares-Quimbaya, A., & Alfonso, L. (2019). Bots and gender profiling on twitter using sociolinguistic features. CLEF.

[4] Dickerson, J. P., Kagan, V., & Subrahmanian, V. S. (2014, August). Using sentiment to detect bots on twitter: Are humans more opinionated than bots?. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)* (pp. 620-627). IEEE.

Anexo A: Manual de Instalación

Seguidamente se describirá la instalación y configuración necesaria para poder ejecutar el proyecto descrito anteriormente:

Para la ejecución de scripts necesitaremos contar con el propio Python en el ordenador destino por lo que instalamos la versión 3.7.9 de este software desde su página oficial:

- https://www.python.org/downloads/

Instalado el software requeriremos de diferentes librerías adicionales, las cuales descargaremos desde un terminal con el comando:

- pip install nltk
- pip install numpy
- pip install Sklearn
- pip install pandas
- pip install Spacy
- pip install text2emotions

Configurado todo nuestro entorno Python podremos observar nuestros archivos .py desde cualquier editor de texto o desde un software especifico.

A continuación, instalaremos el software para la visualización de la aplicación web Django en su versión 2.2.3. Obtendremos éste desde su portal web oficial o ejecutando el siguiente comando:

- pip install Django==2.2.3

Anexo B: Manual de Usuario

En este anexo se explicará todos los procesos necesarios para la visualización de la aplicación web en versión local. Con la descarga de todo el proyecto adjunto se podrá visualizar una carpeta de nombre “ProyectoWeb” la cual guarda todos los archivos necesarios para su ejecución. Para una réplica local seguiremos el siguiente proceso:

Se tendrá que replicar la estructura proporcionada en nuestro Django local por lo que crearemos un proyecto nuevo. Para ello nos situaremos en una carpeta especifica a elegir por el usuario e introduciremos el siguiente comando:

- django-admin startproject ProyetoWeb

Seguidamente crearemos una aplicación la cual tendrá por nombre “ProyectoWebApp” con la introducción del siguiente comando en terminal partiendo desde la carpeta de nuestro proyecto:

- python manage.py startapp ProyectoWebApp

Esto nos creará una nueva carpeta la cual obtendrá el nombre de nuestra aplicación.

Seguidamente, copiaremos todos los archivos ubicados en la carpeta original en nuestra nueva carpeta de proyecto sobrescribiendo todos los archivos anteriormente creados.

Por último, realizaremos una migración en el servidor. Desde la carpeta origen del proyecto introducimos el siguiente comando en consola:

- python manage.py migrate

Y arrancaremos nuestro servidor local:

- python manage.py runserver

Podremos acceder a nuestro servidor desde la url: 127.0.0.1:8000/ProyectoWebApp/ en cualquier navegador web.

Atención: los archivos prueba para la web deben seguir estrictamente los requerimientos explicados en la misma y estar redactados en el idioma de realización de la prueba.

Anexo C: Índice de Ilustraciones

Anexo D: Índice de Tablas

ORTAL RIENDA JUAN CARLOS - 26513437A

Firmado digitalmente por ORTAL RIENDA JUAN CARLOS - 26513437A
Fecha: 2021.09.10 19:07:16 +02'00'