Tu correo, legítimo o no deseado

Vicente Cambrón Tocados
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
viccamtoc@alum.us.es, vicentect10@gmail.com

José Luis García Marín
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
josgarma31@alum.us.es, joselugarciamarin2406@gmail.com

***Resumen*—El objetivo principal de este proyecto consistía en la elaboración de un filtro para correos electrónicos no deseados mediante el empleo de técnicas de procesamiento del lenguaje natural. Con dicho propósito y para el cumplimiento del mismo, se establecieron otro tipo de objetivos para los desarrolladores del proyecto como el aprendizaje de nuevas técnicas de procesamientos del lenguaje natural para la realización de dicho filtro de correos no deseados, así como técnicas para analizar y evaluar los filtros que de había desarrollado. Dicho análisis permitiría también conocer nuevas métricas para evaluar el trabajo realizado además de conocer cuál de los filtros realizados era el que mejor clasificaba los correos como legítimos o no deseados.**

y un resumen de las conclusiones obtenidas.

***Palabras clave*—Inteligencia Artificial, otras palabras clave...**

I. Introducción

El objetivo de la realización de este proyecto surge de la necesidad de incorporar nuevos mecanismos de filtrado a los gestores de correos electrónicos para evitar la recepción de correos Spam. Los correos basura (Spam, en inglés) también conocidos como correos no deseados o correos no solicitados hace referencia a los mensajes de correo electrónico no deseados, no solicitados o cuyo remitente se desconoce, habitualmente de tipo publicitario, que son enviados de manera masiva perjudicando de alguna o otra manera al receptor del mensaje [1].

El envío de este tipo de correos ha aumentado de manera exponencial en los último años debido mayormente al interés de las empresas en dar a conocer sus productos. Por este motivo, es necesario realizar filtros de correos no deseados más firmes y certeros que permitan a los gestores de correos electrónicos desechar aquellos correos no deseados para los usuarios, mejorando así su experiencia de usuario.

Actualmente, existen multitud de técnicas y métodos aplicables para la creación de estos filtros de correo no deseados que se requieren hoy en día. En este proyecto, nos centraremos en una rama de la inteligencia artificial en concreto, Procesamiento del Lenguaje Natural. Dentro del campo del procesamiento del lenguaje natural nos centraremos en la aproximación empírica que usan técnicas basadas en el análisis estadístico de grandes volúmenes de texto. Es una aproximación ascendente, ya que los modelos se construyen a partir del propio corpus de texto del que se dispone [2]. Algunas de las técnicas que se usarán y que se irá explicando

Fig. 1. Ejemplo de una clasificación realzada mediante el modelo KNN

a lo largo del artículo son KNN, Naive Bayes multinominal y modelos de bolsa de palabras. Además, también se han utilizado otro tipo de técnicas sobre las nombradas con el fin de mejorar los filtros obtenidos.

Con el propósito de proporcionar el mejor filtro se realizó análisis exhaustivo para conocer cuál de los filtros desarrollados era el que realizaba la clasificación entre correos legítimos y no deseados de manera más eficiente y con un mayor número de aciertos.

Tras la utilización de todas estas técnicas se obtuvo el resultado final y objetivo de este proyecto de investigación.

En los siguientes apartados de este articulo, se explicará de manera detalladas las técnicas y métodos que han sido utilizados para la construcción del filtro de correos no deseados, así como menciones a trabajos relacionados con este proyecto. Por otro lado, se añadirá la información correspondientes a los materiales y herramientas software que han sido necesarias durante las fase de desarrollo. También, se proporcionará información sobre la metodología de trabajo seguida durante la elaboración del proyecto. Además, se adjuntará un aparatado en el que se enumeraran cuales son los resultados obtenidos y otro apartado en el que se especificarán cuales son las conclusiones obtenidas tras finalización el proyecto.

II. Preliminares

En esta sección, se comentará de manera breve las técnicas y métodos que han sido empleados. Por otro lado, también se hará mención a trabajos realizados por terceras personas que se asemejan al propósito de nuestro proyecto.

En primer lugar, hablaremos sobre las técnicas y métodos empleados.

A. Métodos y técnicas empleados

Para la elaboración de los filtros, se ha empelado la tarea de clasificación de documentos, problema muy frecuentado en el procesamiento del lenguaje natural, campo de la inteligencia artificial que permite a los ordenadores entender y comprender el lenguaje humano.

Para la realización de la tarea de clasificación, se es necesario emplear dos tipos de modelos, un modelo de lenguaje y un modelos de clasificación.

- Modelo de lenguaje: nos permiten poder representar los documentos que se usarán para tanto para representar los correos que se usarán para entrenar el algoritmo que realzará la tarea de clasificación como para representar los nuevos correos que deben ser clasificados como legítimos o no deseado.
- Modelos de clasificación: Dado un conjunto de datos de entrenamiento, junto con las etiquetas objetivo, la clasificación determina la etiqueta de clase para un caso de prueba no etiquetado. [4]

En este caso, el modelo de lenguaje que se ha decidido utilizar es el modelo de bolsa de palabra que "dado un vocabulario finito de términos V prefijado, cada documento se representa como el vector de la cantidad de veces que ocurre cada término en el documento" [2].

En cuanto los modelos de clasificación, modelos que nos permiten clasificar los correos entrantes como legítimos o no deseados, hay que destacar la utilización de dos tipos de estos modelos.

- El primero de ellos es el modelo de clasificación Naive Bayes multinominal, "un clasificador probabilístico fundamentado en el teorema de Bayes" [3].

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

- Él otro de los modelos que clasificación que se han empleado el modelo KNN (K Nearest Neighbors), que se consiste en buscar los k documentos más cercanos al documento a clasificar para a partir de ellos poder clasificar el nuevo documento como las clase predominante de los k documentos más cercanos. Dicho modelo ha sido utilizado para varios k, es decir, se ha probado la precisión del algoritmo para un conjunto de k vecinos diferentes. Además, los k que han sido escogido han sido siempre números impares con el objetivo de evitar los empates entre las clases a clasificar.

Por otro lado, hay que destacar que se han aplicado una serie de técnicas a los métodos mencionados anteriormente se han incorporado una serie de técnicas que nos con el fin de intentar mejorar estos métodos. Las técnicas utilizadas son las siguientes:

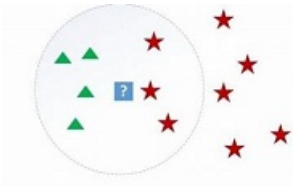

Fig. 2. Ejemplo de una clasificación realzada mediante el modelo KNN

- Eliminación de ruido: conjunto de tareas de normalización de texto específico que a menudo tienen lugar antes de la tokenización. Algunas tareas de eliminación de ruido son:
  - Eliminar encabezados de archivos de texto, pies de página
  - Eliminar HTML, XML, etc. marcado y metadatos
  - Extraer datos valiosos de otros formatos, como JSON [8]
- Tokenización: es un paso que divide cadenas de texto más largas en piezas más pequeñas o tokens. Los trozos de texto más grandes pueden ser convertidos en oraciones, las oraciones pueden ser tokenizadas en palabras, etc. El procesamiento adicional generalmente se realiza después de que una pieza de texto ha sido apropiadamente concatenada. La tokenización también se conoce como segmentación de texto o análisis léxico. A veces la segmentación se usa para referirse al desglose de un gran trozo de texto en partes más grandes que las palabras (por ejemplo, párrafos u oraciones), mientras que la tokenización se reserva para el proceso de desglose que se produce exclusivamente en palabras [9].
- Normalización: se refiere a una serie de tareas relacionadas destinadas a poner todo el texto en igualdad de condiciones: convirtiendo todo el texto en el mismo caso (superior o inferior), eliminando la puntuación, convirtiendo los números a sus equivalentes de palabras, y así sucesivamente. La normalización pone todas las palabras en pie de igualdad, y permite que el procesamiento proceda de manera uniforme [10].

B. Trabajo Relacionado

Hoy en día, todos los gestores de correo electrónico ofrecen una funcionalidad de filtrado de correos no deseados. En este apartado del artículo haremos mención a algunos proyectos software en los que se desarrollaron filtros Spam que se usan en la actualidad.

En primer lugar, empezaremos hablando de MailWasher Free, un programa claro que funciona con todos los programas de correo electrónico. MailWasher filtra sus mensajes de correo antes de que se descarguen en su computadora (o dispositivo móvil). Verá una vista previa del correo y puede marcar los mensajes y eliminarlos antes de recuperarlos. Actualmente, se encuntra disponible para Windows, Android

y MacOS y dispone de una versión gratuita para una cuenta de correo electrónico, versión de pago para varias cuentas. [5].

Por otro lado, SPAMfighter es un filtro de spam para Outlook, Outlook Express, Windows Mail, Thunderbird y Windows Live Mail. Se produce después de que el desarrollador de software se asoció con Microsoft para crear el filtro antispam ”más fuerte, seguro y efectivo” del mercado. Esencialmente, SPAMfighter bloqueará el spam de su bandeja de entrada. Si, por casualidad, aún recibe un correo electrónico no deseado, se le insta a informarlo. Esto ayudará a eliminarlo de los buzones de todos los demás miembros de la comunidad con solo un clic. En general, SPAMfighter proporciona funciones fáciles de usar y está comprobado que obtiene resultados. Disponible en dos versiones diferentes, SPAMfighter Standard y SPAMfighter PRO, la tecnología de bloqueo de spam del programa es galardonada y lo ayudará a limpiar su bandeja de entrada de inmediato [6].

En último lugar, hablaremos de SpamSieve, una aplicación disponible solamente para macOS, que funciona como complemento a tu gestor de correo, y que filtra de manera inteligente todo el spam que podría llegar a tu inbox después de haberse saltado los clásicos filtros incluidos por defecto en los diferentes servidores de los servicios de correo que existen. SpamSieve puede ayudarte en este proceso gracias a su filtrado bayesiano, el cual mueve los correos que detecte que no son deseados a la carpeta de Spam. Es importante destacar esa parte de su funcionamiento ya que muchos usuarios sienten miedo a poder perder algo importante al instalar está clase de filtros, pero al simplemente mover los correos y no eliminarlos por completo, puedes estar seguro de poder recuperar un falso positivo. Además SpamSieve aprende con el tiempo, así que gracias a su potencial y un mínimo de entrenamiento por tu parte, con el tiempo podrás decirle adiós al spam prácticamente para siempre [7].

III. Materiales y Herramientas

En esta sección, nos centraremos en el lenguaje de programación utilizado para el desarrollo del código y de las librerías que han sido usadas para la implementación de dicho código.

En primer lugar, comenzaremos hablando del lenguaje de programación escogido para la realización del proyecto. La inteligencia artificial en general, y el machine learning en particular, está creciendo a un ritmo vertiginoso. El mercado necesita un número cada vez más grande de programadores capaces de aportar soluciones rápidas y funcionales, y una de las mejores formas de conseguirlo es a través de Python [11] 3. Algunas de la ventajas que nos proporciona la utilización de este lenguaje de programación son:

- Sencillez y facilidad de aprendizaje
- Es open source
- Abarca a una gran comunidad
- Ofrece multitud de librerías
- Versatilidad y flexibilidad
- Excelente representación de los datos

Fig. 3. Ejemplo de una clasificación realzada mediante el modelo KNN

A continuación, se nombrarán y se añadirá una breve explicación de las librería de Python que han sido utilizadas durante el desarrollo del proyecto.

- Scikit-learn es una biblioteca para aprendizaje automático de software libre. Incluye varios algoritmos de clasificación, regresión y análisis de grupos entre los cuales están máquinas de vectores de soporte, bosques aleatorios, Gradient boosting, K-means y DBSCAN. Está diseñada para interoperar con las bibliotecas numéricas y científicas NumPy y SciPy [12].
- NumPy es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas [13].
- OS proporciona una forma portátil de utilizar la funcionalidad dependiente del sistema operativo [14].
- Email es una biblioteca para administrar mensajes de correo electrónico. No está diseñado específicamente para realizar ningún envío de mensajes de correo electrónico a SMTP (RFC 2821), NNTP u otros servidores; esas son funciones de módulos como smtplib y nntplib [15].
- Re proporcionaoperaciones de coincidencia de expresiones reales similares a las que se encuentran en Perl [16].
- Collections implementa tipos de datos de contenedores especializados que proporcionan alternativas a los contenedores, dictados, listas, conjuntos y tuplas integrados de propósito general de Python [17].
- Pickle implementa protocolos binarios para serializar y desserializar una estructura de objetos Python. ”Pickling” es el proceso por el cual una jerarquía de objetos Python se convierte en un flujo de bytes, y ”unpickling” es la operación inversa, mediante la cual un flujo de bytes se convierte de nuevo en una jerarquía de objetos [18].
- Bs4 utilizada para analizar documentos HTML (incluyendo los que tienen un marcado incorrecto). Esta biblioteca crea un árbol con todos los elementos del documento y puede ser utilizado para extraer información [19].

Fig. 4. Ejemplo de una clasificación realzada mediante el modelo KNN

IV. Metodología

En este apartado se va a detallar cuál ha sido la metodología de trabajo seguida durante el proyecto.

A. Conjunto de entrenamiento y conjunto de prueba

En primer lugar, se partió de un conjunto de correos que ya habían sido clasificados como legítimos o no deseados previamente. Para la creación del los filtros de correo no deseados se decidió dividir este conjunto de correo ya clasificados en dos subconjuntos.Para la realización de esta tarea se utilizó la librería split-folders. El primero de ellos se trató del conjunto de entrenamiento, el cual contendría aquellos correos que se utilizaría para que los algoritmos de clasificación. El otro de los conjuntos es el de prueba, que se utilizó para comprobar la eficacia de los algoritmos de clasificación detectando cuales de las clasificaciones habían sido verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. La separación del conjunto inicial de correos se realizó de manera aleatoria dejando un 20% de los correos para el subconjunto de prueba y un 80% de los correos para el subconjunto de entrenamiento.

B. Vocabulario

Una vez separados los correos que se usaría para el entrenamiento y para las pruebas, el siguiente paso que se dió fue definir un el vocabulario de palabras que utilizarían los modelos de lenguaje. Para ello, el primer paso fue la extracción de los cuerpos de cada uno de los correos, tarea que pudo llevar a cabo gracias al paquete email. Posteriormente, dichos cuerpos fueron escritos y unificados que un único archivo txt que nos facilitaría el trabajo posteriormente. Debido a que los correos estaban separados en legítimos y no deseados, se obtuvieron dos de estos ficheros. A continuación, para obtener el vocabulario se leyeron los ficheros y se llevo a cabo una separación de las palabras contenidas en dicho fichero (empelando el método split de la librería re). A medida que se iban separando las palabras, estas iban siendo añadidas a un conjunto, con el objetivo de evitar palabras repetidas, y además se eliminaron aquellas palabras que contuvieran caracteres no alfanuméricos y dígitos. Una vez realizado estos pasos, se obtuvo el vocabulario completo y con el fin de evitar tener que realizar esta secuencia de acciones para poder usar el vocabulario se decidió escribir dicho vocabulario en un fichero JSON, mediante el método dump de la librería pickle, para que este pudiera ser reutilizado.

C. Bolsa de Palabras y Naive Bayes Multinominal

El primero de los filtros que se implementó fue mediante el modelo de lenguaje bolsa de palabras (Bag of Words, BoW) y el modelo clasificación Navive Bayes multinominal. El primero de los pasos fue cargar el vocabulario que se usaría para entrenar el modelo. Para ellos se realizaban los pasos descritos previamente en la creación del vocabulario, a excepción de la existencia del fichero JSON en el que ya estaría cargado el vocabulario a utilizar y en cuyo caso se leería usando el método load de la librería pickle. Una vez conocido el vocabulario, el siguiente pasa sería entrenar el modelo con el conjuntos de correos de entrenamiento. Para empezar el entrenamiento, se utilizó el método CountVectorizer que nos permitió contar la frecuencia de las palabras del vocabulario en los documentos, es decir, de los emails. Dicha método lo aplicamos tanto para los correos legítimos como para los no deseados del conjunto de correos de entrenamiento. El siguiente paso consistía en la obtención de los corpus que consistían en una lista de los cuerpos de los mensajes para cada clase, legítimos y no deseados. Una vez que contábamos con la frecuencia de las palabras del vocabulario y con los corpus, se llevo a cabo el entrenamiento del modelo, usando el método fit_transform. Como resultado del entrenamiento, se obtuvo dos matrices, una para los correos legítimos y otra para los no deseados, que contenían los vectores de la frecuencia de las palabras por cada documento.

Para llevar a cabo la clasificación, se ha utilizado el clasificador Naive Bayes multinominal, el cual esta basado en la siguiente fórmula (1)

$$arg\ máx_{c \in C}\ (P(c) \prod_{t \in V} P(t|c)^{n_{D,t}}) \tag{1}$$

En este punto cabe destacar que para evitar los desbordamientos numéricos las probabilidades del modelo Naive Bayes han sido calculadas mediante la siguiente fórmula (2)

$$arg\ máx_{c \in C}\ (log(P(c))\ + \sum_{t \in V} n_{D,t} * log(P(t|c))) \tag{2}$$

Donde C es las posibles clases en las que se puede clasificar, en este caso Legítimo y No Deseados, V hace referencia al conjunto de términos y n es la ocurrencia de un término del vocabulario en el documento.

D. Modelo TF-IDF y Clasificador KNN

El segundo y último filtro se ha implementado a través del modelo de lenguaje Tf-idf y el modelo de clasificación KNN. Como en el anterior modelo, el primer paso consistió en realizar la carga del vocabulario. El procedimiento seguido para ello exactamente el mismo que el modelo anterior. Posteriormente, se obtuvieron el corpus de entrenamiento. Difiriendo del modelo anterior, este consistía únicamente en una lista de los cuerpos de los mensajes del todo el conjunto de entrenamiento, es decir, tanto de los correos de entrenamiento legítimos como no deseados.

Por tanto, para construir el modelo tf-idf, usamos TfidfVectorizer de la librería sklearn. A continuación se generó el modelo con el vocabulario y se entreno dicho modelo con el corpus obtenido previamente, usando el método fit_transform de TfidfVectorizer. El resultado de esto resulto una matriz de los vectores de obtenidos por el modelo de lenguaje tf-idf para cada documento de entrenamiento.

Para la clasificación de los correos, una vez entrenado el modelo, se genera un nuevo vector tf-idf a partir del modelo usando el método transform de TfidfVectorizer.

Por último, utilizamos el modelo de clasificación KNN, el cual asignará la clase mayoritaria de calcular las similitudes (3) de todos los documentos del conjuntos de entrenamientos con el nuevo correo a clasificar.

$$sim(D1, D2) = \frac{D1 \cdot D2}{\|D1\| \cdot \|D2\|} \tag{3}$$

V. Resultados

En esta sección se detallarán tanto los experimentos realizados como los resultados conseguidos:

- Los experimentos realizados, indicando razonadamente la configuración empleada, qué se quiere determinar, y como se ha medido.
- Los resultados obtenidos en cada experimento, explicando en cada caso lo que se ha conseguido.
- Análisis de los resultados, haciendo comparativas y obteniendo conclusiones.

Se pueden hacer uso de tablas, como el ejemplo de la tabla **??**.

VI. Conclusiones

Finalmente, se dedica la última sección para indicar las conclusiones obtenidas del trabajo. Se puede dedicar un párrafo para realizar un resumen sucinto del trabajo, con los experimentos y resultados. Seguidamente, uno o dos párrafos con conclusiones. Se suele dedicar un párrafo final con ideas de mejora y trabajo futuro.

Referencias

[1] Correo Basura *(2022, 29 de mayo). Wikipedia, la enciclopedia libre. Fecha de consulta: 12:10, junio 7, 2022 desde* https://es.wikipedia.org/wiki/Correo_basura

[2] Profesores de la asignatura Inteligencia Artificial, Dpto. de Ciencias de la Computación e Inteligencia Artificia, Universidad de Sevilla *(7/04/2022 13:15). Procesamiento del lenguaje natural. (p. 2)*

[3] Clasificador bayesiano ingenuo *2022, 22 de febrero). Wikipedia, la enciclopedia libre. Fecha de consulta: 15:40, junio 7, 2022 desde* https://es.wikipedia.org/wiki/Clasificador_bayesiano_ingenuo

[4] Moreno, I. *Introdución a la clasificación. ¿Cómo funcionan las clasificaciones y los clasificadores?* https://www.statdeveloper.com/introduccion-a-la-clasificacion/

[5] MNI.ORG.PE *Filtros de spam. Software de filtro de spam* https://mni.org.pe/blog/filtros-de-spam/

[6] FileHippo.com https://filehippo.com/es/download_spam-fighter/

[7] Limni *Cómo fulminar el Spam de tu email con SpamSieve. Qué es SpamSieve* https://limni.net/eliminar-spam-mac-spamsieve/

[8] Mayo, M *(2018, 3 de mayo). Ciencia y Datos. Preprocesamiento de datos de texto: un tutorial en Python. Eliminación de ruido* https://medium.com/datos-y-ciencia/preprocesamiento-de-datos-de-texto-un-tutorial-en-python-5db5620f1767

[9] Mayo, M *(2018, 3 de mayo). Ciencia y Datos. Preprocesamiento de datos de texto: un tutorial en Python. Tokenización* https://medium.com/datos-y-ciencia/preprocesamiento-de-datos-de-texto-un-tutorial-en-python-5db5620f1767

[10] Mayo, M *(2018, 3 de mayo). Ciencia y Datos. Preprocesamiento de datos de texto: un tutorial en Python. Normalización* https://medium.com/datos-y-ciencia/preprocesamiento-de-datos-de-texto-un-tutorial-en-python-5db5620f1767

[11] Structuralia *(2020, 23 de octubre). La utilidad de Python para la inteligencia artificial. Python y sus ventajas para los programadores de inteligencia artificial.* https://blog.structuralia.com/python-inteligencia-artificial

[12] Scikit-learn. *(2020, 8 de noviembre). Wikipedia, La enciclopedia libre. Fecha de consulta: 11:48, junio 9, 2022 desde* https://es.wikipedia.org/w/index.php?title=Scikit-learn&oldid=130746122.

[13] NumPy. *(2022, 21 de febrero). Wikipedia, La enciclopedia libre. Fecha de consulta: 11:31, junio 9, 2022 desde* https://es.wikipedia.org/w/index.php?title=NumPy&oldid=141830635.

[14] Python *3.10.5 Documentation. The Python Standard Library. Generic Operating System Services. os — Miscellaneous operating system interfaces* https://docs.python.org/3/library/os.html

[15] Python *3.10.5 Documentation. The Python Standard Library. Internet Data Handling. email — An email and MIME handling package* https://docs.python.org/3/library/email.html?highlight=email#module-email

[16] Python *3.10.5 Documentation. The Python Standard Library. Text Processing Services. re — Regular expression operations.* https://docs.python.org/3/library/re.html?highlight=re#module-re

[17] Python *3.10.5 Documentation. The Python Standard Library. Data Types. collections — Container datatypes.* https://docs.python.org/3/library/collections.html?highlight=collections#module-collections

[18] Python *3.10.5 Documentation. The Python Standard Library. Data Persistence. pickle — Python object serialization.* https://docs.python.org/3/library/pickle.html?highlight=pickle#module-pickle

[19] Beautiful Soup. *(2020, 16 de abril). Wikipedia, La enciclopedia libre. Fecha de consulta: 11:29, junio 9, 2022 desde* https://es.wikipedia.org/w/index.php?title=Beautiful_Soup&oldid=125249372.