



# **Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español**

**Proyecto de Fin de Grado en ETS de Ingeniería Informática de modalidad específica**

**Realizado por: Roi Arias Rico**

**Dirigido por: Raquel Martínez Unanue**

**Codirigido por: María del Soto Montalvo Herranz**

**Curso académico: 2021/2022 Convocatoria de defensa:**



# **Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español**

**Proyecto de Fin de Grado en ETS de Ingeniería Informática de modalidad específica**

Realizado por: Roi Arias Rico

Dirigido por: Raquel Martínez Unanue

Codirigido por: María del Soto Montalvo Herranz

Fecha de lectura y defensa del proyecto:.....

## Resumen del proyecto

Los efectos adversos de los medicamentos constituyen un problema de salud muy importante tanto a nivel económico como sanitario. A pesar de que existen instituciones dedicadas a recoger las posibles sospechas de efectos adversos a los medicamentos, los casos detectados suelen ser inferiores a los reales. Por ello, una de las posibilidades de detectar efectos adversos está relacionada con las redes sociales debido a su amplio uso y disponibilidad de datos. En este proyecto se trata de recopilar los mensajes de los usuarios en una red social y extraer los efectos adversos de los medicamentos mencionados. Debido a los avances en aprendizaje automático, hoy en día se pueden emplear diferentes algoritmos que permitan automatizar y clasificar la existencia de este tipo de relaciones. A pesar de los avances, no es un proceso fácil ya que se trata de documentos desestructurados. Las dificultades de este tipo de procesos se basan en que los textos no tienen una gramática clara, con abundantes errores de ortografía o presencia de símbolos como *emojis* que dificultan la tarea de identificación. Por eso, en este tipo de documentos se inicia con un proceso de limpieza de textos para evitar los errores de identificación. A menudo, es acompañado de etapas de preprocesamiento como tokenización, lematización, etiquetado PoS o *stemming*. Una vez que disponemos de un texto sin tantos errores, se procede al reconocimiento de entidades, ya sea mediante algoritmos de aprendizaje automático o mediante la creación de reglas o patrones de búsqueda. Este trabajo se apoya en el empleo de diccionarios elaborados *ad hoc* para este trabajo y del diccionario UMLS para términos médicos. Una vez identificados los términos, se inicia la clasificación que permite detectar la presencia o ausencia de efectos adversos y la capacidad de predicción.

## **Lista de palabras clave**

Efectos adversos de medicamentos; procesamiento de lenguaje natural; minado de datos; aprendizaje automático.

## **Abstract**

### **System for the detection of adverse drug effects in biomedical texts in Spanish**

The adverse effects of medications are a very important health problem both economically and health-wise. Despite the fact that there are institutions dedicated to collecting possible suspicions of adverse drug effects, the cases detected are usually lower than the real ones. Therefore, one of the possibilities of detecting adverse effects is related to social networks due to their extensive use and availability of data. This project tries to collect the messages of the users in a social network and extract the adverse effects of the mentioned medicines. Due to advances in machine learning, today different algorithms can be used to automate and classify the existence of this type of relationship. Despite the advances, it is not an easy process since it deals with unstructured documents. The difficulties of this type of process are based on the fact that the texts do not have a clear grammar, with abundant spelling errors or the presence of symbols such as emojis that make the identification task difficult. For this reason, this type of document begins with a text cleaning process to avoid identification errors. It is often accompanied by pre-processing steps such as tokenization, stemming, PoS tagging, or stemming. Once we have a text without so many errors, we proceed to the named-entity recognition, either through machine learning algorithms or through the creation of rules or search patterns. This work is supported by the use of dictionaries prepared ad hoc for this work and the UMLS dictionary for medical terms. Once the terms have been identified, the classification begins, which allows detecting the presence or absence of adverse effects and its predictive capacity.

## **Keywords**

Adverse effects; natural language processing; data mining; machine learning.

# Índice

1. Introducción .....	1
1.1 Descripción del problema .....	1
1.2 Motivación .....	3
1.3. Objetivos .....	4
1.4 Estructura del trabajo.....	5
2. Antecedentes .....	7
2.1 Extracción de la información aplicada al dominio biomédico .....	7
2.2 Revisión de las fuentes de información biomédicas .....	8
2.3 Reconocimiento/Extracción de entidades a través de técnicas de Aprendizaje Automático.....	9
2.3.1 Aprendizaje supervisado .....	10
2.3.2 Aprendizaje no supervisado .....	11
2.3.3 Espacios vectoriales.....	12
2.3.4 Enfoque seleccionado .....	12
3. Planteamiento del problema .....	15
3.1 Teoría de la extracción de conceptos de texto no estructurados.....	15
3.2 Herramientas utilizadas .....	16
3.3 Construcción de diccionario de medicamentos .....	17
3.3.1 CIMA.....	18
3.3.2 Vademecum .....	20
3.4 Construcción de diccionario de términos médicos .....	22
3.4.1 MedDRA- (español) .....	22
3.4.2. Diccionario de términos médicos de la Real Academia Nacional de Medicina .....	24
3.5 Minado de mensajes en Twitter .....	25
4. Descripción del sistema desarrollado.....	31
4.1 Esquema de la solución.....	31
4.2 Limpieza de texto .....	31
4.3 Preprocesamiento .....	34
4.3.1 Tokenización.....	34
4.3.2 Lematización/Stemming.....	35
4.3.3 Stopwords .....	37
4.4 Reconocimiento de entidades (NER).....	38

4.5 Clasificación de efectos adversos.....	40
4.5.1 Extracción de características ( <i>feature extraction</i> ).....	41
4.5.2 Algoritmos considerados.....	43
4.6. Desarrollo de una aplicación basada en el sistema .....	46
5. Metodología de desarrollo y diseño .....	49
5.1 Metodología .....	49
5.2 Planificación .....	49
5.3 Requisitos del sistema.....	51
5.3.1 Requisitos funcionales.....	51
5.3.2 Requisitos no funcionales .....	52
5.4 Casos de uso .....	52
6. Evaluación del sistema .....	57
6.1 Métricas para la extracción de información.....	57
6.2 Resultados .....	59
6.2.1 <i>Corpus</i> de prueba .....	59
6.2.2 Limpieza.....	59
6.2.2 Reconocimiento de entidades.....	60
6.2.3 Clasificación de textos .....	61
7. Conclusiones.....	67
7.1 Conclusiones.....	67
7.2 Posibles mejoras y líneas futuras de desarrollo.....	68
8. Presupuesto y cálculo de costes .....	71
8.1 Descripción del proyecto.....	71
8.2 Cálculo de costes.....	71
8.3 Presupuesto .....	73
Bibliografía .....	75
Listado de siglas, abreviaturas y acrónimos.....	79
Anexo .....	81
Anexo A Guía de instalación.....	81
Anexo B Manual de usuario .....	83



# Indice de figuras

Figura 1. Pipeline de detección de efectos adversos .....	14
Figura 2. Ejemplo de entrada del archivo DICCIONARIO_ATC.xml .....	19
Figura 3. Ejemplo de entrada del archivo DICCIONARIO_PRINCIPIOS_ACTIVOS.xml .....	19
Figura 4. Ejemplo de estructura en CIMA de nombres comerciales de medicamentos .....	20
Figura 5. Distribución de menciones de marcas comerciales agrupadas por ATC.....	27
Figura 6. Distribución de menciones de principios activos agrupadas por ATC .....	29
Figura 7. Ejemplos de mensajes extraídos de la red social Twitter .....	30
Figura 8. Pipeline PLN para detección de efectos adversos.....	31
Figura 9. Ejemplo de preprocesamiento de tweets con eliminación del usuario .....	33
Figura 10. Ejemplo de tokenización .....	34
Figura 11. Componentes de la librería SpaCy (Fuente: SpaCy.io) .....	35
Figura 12. Ejemplo de reconocimiento de entidades empleando stemming .....	37
Figura 13. Definición de la entidad "dosis" .....	39
Figura 14. Expresión matemática de la similitud mediante Jaccard .....	40
Figura 15. Fórmula TF-IDF .....	42
Figura 16. Expresión matemática de la regresión logística .....	43
Figura 17. Función logística .....	44
Figura 18. Pantalla con los resultados finales del sistema .....	47
Figura 19. Diagrama de Gantt del proyecto .....	50
Figura 20. Matriz de confusión.....	57
Figura 21. Resultados de la limpieza del corpus .....	59
Figura 22. Extracto de un texto tras NER .....	60
Figura 23. Resultados del reconocimienro de entidades.....	60
Figura 24. Número de efectos adversos detectados en los tweets. ....	61
Figura 25. Matriz de confusión obtenida en la regresión logística .....	65
Figura 26. Curva ROC resultado de la Regresión Logística .....	65

# Indice de tablas

Tabla 1. Algoritmo estadístico de aprendizaje .....	14
Tabla 2. Distribución de tipos semántica de MedDRA.....	24
Tabla 3. Clasificación de marcas comerciales según apariciones en la red social .....	26
Tabla 4. Clasificación de principios activos según apariciones en la red social .....	28
Tabla 5. Ejemplos de preprocesamiento de emojis .....	34
Tabla 6. Distribución del esfuerzo .....	51
Tabla 7. Caso de uso.....	53
Tabla 8. Flujo básico del caso de uso .....	54
Tabla 9. Flujo alternativo 1.....	54
Tabla 10. Flujo alternativo 2.....	55
Tabla 11. Resultados de la precisión de los algoritmos .....	62
Tabla 12. Resultados de los algoritmos.....	63
Tabla 13. Resultados de la validación cruzada .....	63
Tabla 14. Comparativa de resultados TF-IDF vs TF-IDF+Word2Vec.....	66
Tabla 15. Costes de personal .....	71
Tabla 16. Costes de equipamiento.....	72
Tabla 17. Costes de software .....	72
Tabla 18. Costes de material fungible y otros gastos.....	73
Tabla 19. Presupuesto del proyecto.....	73





# 1. Introducción

## 1.1 Descripción del problema

De acuerdo con la Organización Mundial de la Salud (OMS), una reacción adversa al medicamento (RAM) se puede definir como una respuesta a un fármaco que es nociva y no intencionada y que tiene lugar cuando este se administra en dosis utilizadas normalmente en seres humanos para la profilaxis, diagnóstico o tratamiento de una enfermedad, o para la modificación de una función fisiológica. Como consecuencia de ello, se puede definir la Farmacovigilancia como la actividad de salud pública que tiene por objetivo la identificación, cuantificación, evaluación y prevención de los riesgos del uso de los medicamentos una vez comercializados[1].

Esta actividad recibe un elevado grado de atención tanto por las autoridades competentes en el ámbito de la autorización de comercialización de fármacos como de la propia industria farmacéutica[2].

En una situación ideal, todos los efectos adversos asociados con un fármaco son detectadas antes de la comercialización del propio fármaco. Sin embargo, esto no siempre es posible porque el número de individuos que participan en un ensayo clínico es pequeño comparada con toda la población lo que limita la capacidad de detectar efectos adversos, sobre todo si además son muy poco frecuentes. Otro motivo es la duración de los ensayos clínicos que evitan la detección de reacciones adversas a largo plazo. Por último, se debe destacar que los ensayos clínicos muestran unas condiciones ideales de utilización de fármacos, mientras que las condiciones reales sólo se manifiestan tras la autorización de comercialización del medicamento, eso significa que las posibles interacciones con otros medicamentos no se pueden predecir en cuanto a la magnitud e implicaciones para la salud de los pacientes.

Las autoridades competentes en cada país o en cada región como *Federal Drugs Administration* (FDA) en Estados Unidos o la Agencia Española de Medicamentos y

Productos Sanitarios (AEMPS) en España obligan a los profesionales sanitarios a informar de las reacciones adversas siempre que atribuyan un problema de salud a la

utilización de algún fármaco. Cada organismo nacional ha diseñado sistemas de notificación como MedWatch por la FDA o la tarjeta amarilla en el caso español con el objeto de facilitar la comunicación de estos efectos adversos. A pesar de ello, un gran inconveniente de estos sistemas de notificación es que se ha llegado a comprobar que informan menos RAM de las que realmente existen[3], incluso se ha llegado a estimar que las notificadas representan 2-10% del total[4-6].

El aumento en la disponibilidad de historiales clínicos electrónicos y la capacidad de procesar grandes volúmenes automáticamente gracias al Procesamiento de Lenguaje Natural (PLN) y algoritmos de aprendizaje automático han abierto nuevas oportunidades a la Farmacovigilancia[7]. Dentro de la Red, el gran desarrollo de las redes sociales como Twitter, Facebook, Instagram y Pinterest las han catapultado como fuentes de datos para análisis de mercado en diferentes. Sin embargo, las redes sociales no están exentas de inconvenientes para estas tareas que las redes sociales incluyen la credibilidad, frecuencia, novedad, unicidad o la importancia de los datos. Estos cinco problemas hacen muy importante seleccionar el tipo de red social. Así, Twitter al tener un límite bajo de caracteres en sus mensajes no podría tener una alta relevancia (*salience data*). Otros artículos señalan que existen muy pocos mensajes que hagan referencia a los efectos adversos de los medicamentos[8]. Esto lleva a problemas relacionados con las anotaciones porque es necesario anotar grandes volúmenes de datos para la inclusión de suficientes mensajes que contengan efectos adversos.

Algunos estudios han propugnado por la combinación tanto de los sistemas de información espontáneos como de los historiales clínicos digitales ya que aumenta la exactitud en la detección de efectos adversos[9]. En países de nuestro entorno se han establecido guías por parte de la *Association of the British Pharmaceutical Industry* (ABPI), estableciendo los datos mínimos de información necesarios para informar del efecto adverso, como que se pueda identificar al paciente, medicamento sospechoso,

efectos adversos e informador identificable (correo electrónico, nombre de usuario). En España se ha establecido un Plan de Impulso de las Tecnologías del Lenguaje en el que se abarcan diferentes campos, entre ellas el ámbito sanitario[10].

### 1.2 Motivación

Estudios en otros países han demostrado que las RAM son responsables de ingresos hospitalarios, aumentos en la estancia hospitalaria y de la mortalidad[11]. Como complemento a los sistemas de información tradicionales, la explotación de datos en redes sociales podría contribuir a su detección temprana.

A pesar de esto, la detección de efectos adversos todavía se encuentra inmadura ya que se deben crear *lexicones* propios para crear un sistema de reconocimiento de entidades y suelen estar limitadas a un grupo de fármacos objetos del estudio[12, 13]. Con el objetivo de extraer información y de detectar potenciales RAM se han empleado técnicas de PLN, no sólo en el minado de textos en redes sociales sino también en hojas de información de fármacos[14], informes médicos[15] e historiales clínicos[16].

El PLN emplea funciones estadísticas y algoritmos computacionales para analizar texto desestructurado y extraer información cuantitativa de él. Debido al pequeño tamaño de los textos de algunas redes sociales, los métodos tradicionales de PLN no son adecuados[17] por lo que se han desarrollado herramientas específicas para datos de las redes sociales[18]

Los enfoques para extraer este tipo de información de textos pueden ser a través de transformadores, empleando técnicas PLN con bases de datos estructurados que contengan terminologías clínicas. Estas bases de datos estructuradas se diseñan para categorizar y clasificar términos médicos e información clínica en tablas estandarizadas con un código único para cada concepto médico. Existen varias bases de datos como por ejemplo *International Classification of Diseases* (ICD), *Systematized Nomenclature of Medicine – Clinical Terms* (SNOMED CT) y *Unified Medical Language System* (UMLS). De todas estas bases de datos, UMLS posee como gran ventaja que proporciona códigos estándar para miles de conceptos biomédicos e incluye los vocabularios de ICD y SNOMED.

A pesar de los buenos resultados cuando se trata de literatura médica u hojas de información de fármacos, el PLN requiere un preprocesamiento previo de los textos de las redes sociales, ya sea por la presencia de metadatos como los *hashtags* en tweets, abreviaturas no convencionales, jerga, *emojis*, faltas de signos de puntuación, errores ortográficos o gramaticales[19]. Además de estos problemas habituales de procesamiento de textos, existen otros relacionados con el ruido y desequilibrio. Mediante el análisis de las características semánticas y lingüísticas de los textos ayuda a clasificar y detectar automáticamente los efectos adversos medicamentosos[20].

Por último, no es menos importante que el hecho de extraer información de estos medios podría representar un desafío a nivel técnico, político o de la protección de datos. A pesar de ello, la hipótesis de este trabajo es que las redes sociales pueden suponer un importante apoyo en las tareas de Farmacovigilancia para detectar efectos adversos desconocidos y, por lo tanto, incrementar la seguridad de los fármacos comercializados. Aunque existen una gran cantidad de estudios empleando como idioma de trabajo el inglés, no existe una gran cantidad de estudios que extraigan información del español.

De esta forma, este trabajo podría contribuir a aumentar el número de estudios que utilizan el idioma español para establecer el estado del arte en este campo y poder comparar los resultados obtenidos con otras técnicas incluidas en la bibliografía.

### 1.3. Objetivos

Este estudio tiene como objetivo primordial el diseño e implementación de un sistema para la detección de efectos adversos a medicamentos en textos biomédicos. En este estudio y, debido al auge de las redes sociales, se ha seleccionado como textos biomédicos los mensajes disponibles en una red social como Twitter.



Debido a que el estudio se divide en varias etapas consecutivas, cada una de éstas posee objetivos específicos como indicamos a continuación:

- Creación de un *corpus* formado por los fármacos y terminología médica en el idioma español que después se empleará en la etapa de desarrollo y evaluación de extracción y reconocimiento de entidades.
- Construcción de un diccionario para fármacos y efectos adversos basado en los recursos empleados en el punto anterior.
- Recolección de datos de la red social Twitter.
- Implementación de un sistema para la detección de fármacos y efectos adversos a través de un enfoque basado en diccionario.
- Evaluación del sistema con el *corpus* previamente creado.
- Análisis de errores para diferenciar las principales causas de falsos positivos y falsos negativos en el sistema desarrollado.
- Comparación con otros sistemas según la bibliografía consultada.

#### 1.4 Estructura del trabajo

El trabajo se encuentra estructurado como sigue:

- Capítulo 1 “Introducción”: Breve introducción al proyecto, motivaciones para su realización, continuando con los objetivos que se pretenden cubrir con el proyecto, así como la estructura del trabajo aquí expuesto.

- Capítulo 2 “Antecedentes”: Descripción de trabajos realizados en el ámbito del análisis de las redes sociales para detectar reacciones adversas a medicamentos. Además, se presenta el análisis de las fuentes de términos en español tanto para fármacos como de efectos adversos, junto con los métodos empleados en la extracción de entidades.

## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

- Capítulo 3 “Planteamiento del problema”: Descripción de la base teórica sobre la que se construye la propuesta de solución de este trabajo, junto con las herramientas utilizadas para su implementación.

- Capítulo 4 “Descripción del sistema desarrollado”: Propuesta de la solución al problema, descripción de recursos utilizados, alternativas valoradas en cada paso y las etapas o tareas realizadas.

- Capítulo 5 “Metodología de desarrollo y diseño”: Se describe los métodos empleados a lo largo de este trabajo, así como su diseño.

- Capítulo 6 “Resultados”: Exposición de los resultados y comparación con los resultados obtenidos de otros estudios incluidos en la bibliografía.

- Capítulo 7 “Conclusiones”: Discusión de los resultados y resumen de las conclusiones, así como potenciales líneas de desarrollo.

- Capítulo 8 “Presupuesto y cálculo de costes”: Planificación y presupuesto total del proyecto, incluyendo gastos directos e indirectos.

## 2. Antecedentes

### 2.1 Extracción de la información aplicada al dominio biomédico

El ámbito principal de este proyecto es el PLN. Mediante el PLN se investiga la manera de comunicar las máquinas con las personas mediante el uso de lenguas naturales, como el español y cuyo principal objetivo es automatizar este proceso de traducción. Como parte del PLN, este proyecto se enfoca en la extracción de información que consiste en extraer automáticamente información estructurada a partir de documentos no estructurados, en este caso mensajes de redes sociales. Los sistemas de extracción de entidades y más concretamente el Reconocimiento de Entidades Nombradas (Named Entity Recognition, por sus siglas en inglés NER) realiza la tarea de buscar información muy concreta en colecciones de documentos, detectar información relevante, extraerla y etiquetarla en un formato adecuado para su procesamiento automático.

Una Entidad Nombrada es una palabra o conjunto de palabras que contienen un nombre como puede ser la marca comercial de un medicamento o un nombre de una persona. Estas entidades suelen tener una longitud de unos pocos caracteres y se encuentran en los textos no estructurados.

Las entidades que abarca este proyecto pertenecen al dominio biomédico, aunque existen ciertas especificidades dentro de este dominio que añaden complejidad a la detección de efectos adversos a medicamentos:

- Los nombres comerciales y los nombres de los principios activos suelen diferir.
- Los nombres comerciales no son constantes en el tiempo, sino que pueden variar.
- El empleo extenso de abreviaturas, junto con modificadores léxicos como el plural o la incorporación de palabras compuestas que reflejen el sistema de liberación o un incremento de dosis, por ejemplo: “retard”, “forte” o “flas”.

Por todo ello, este proyecto aplicará técnicas de NER para reconocer fármacos y efectos adversos de mensajes de usuarios en las redes sociales.

## 2.2 Revisión de las fuentes de información biomédicas

Todas las fuentes de información empleadas para la creación de este proyecto se encuentran en la web de forma libre y de fácil acceso a todos los usuarios. Los términos médicos pertenecen a UMLS que es un conjunto de archivos y software que agrupa los vocabularios y estándares biomédicos y , por extensión, del campo de la salud para permitir la interoperabilidad entre sistemas informáticos. Este sistema ha sido desarrollado por el *National Institute of Health* (NIH) perteneciente al gobierno de Estados Unidos. A pesar de que existen otros vocabularios y bases de datos como *Medical Subject Headings* (MeSH) o *Logical Observation Identifier Names and Codes* (LOINC), UMLS es ya considerado como un estándar, que además es una herramienta con licencia gratuita.

El UMLS emplea los términos médicos relacionados con los síntomas siguiendo los conceptos incluidos en el vocabulario *Medical Dictionary for Regulatory Activities* (MedDRA). Este diccionario de términos médicos fue desarrollado por la *International Conference of Harmonisation* (ICH) y es propiedad de *International Federation of Pharmaceutical Manufacturers and Associations* (IFPMA). MedDRA hace especial hincapié en la entrada, recuperación, análisis y visualización de datos. Se aplica en todas las fases del desarrollo de fármacos excepto en la experimentación animal, además de los efectos sobre la salud y funcionamiento defectuoso de dispositivos médicos. Además, esta terminología tiene como ventaja que está traducida a múltiples idiomas, incluido el español (conocida como MDRSPA) . La última edición fue en Marzo 2022, si bien para este proyecto se ha empleado una versión anterior incluida en el metatesauro de UMLS (Noviembre 2021).

## Antecedentes

En cuanto a las fuentes de medicamentos suelen estar disponibles a través de las agencias sanitarias nacionales competentes. En el caso de España, se denomina Centro de Información de Medicamentos de la AEMPS (CIMA) aunque en ocasiones otros autores han optado por la disponibilidad de información en otras webs no institucionales dedicadas a la información de medicamentos. Es habitual para la extracción de información y creación de un diccionario de términos utilizar la información disponible en la red mediante técnicas de *webscraping*[21]. El empleo de *webscraping* implica un ahorro de tiempo y costes al no depender de sistemas de pago y de fácil disponibilidad de información.

Un importante obstáculo es el idioma empleado, aunque ha habido multitud de estudios y trabajos destinados a explotar estas redes sociales, hay una gran mayoría en inglés y en mucha menor medida en español[22, 23]. La primera autora que realizó un *corpus* de efectos adversos y de fármacos en español fue Segura-Bedmar et al[21], a través de la extracción de datos de un foro clínico en español. Mediante posteriores trabajos de esta misma autora, se construyó un diccionario de fármacos y sus efectos adversos utilizando la base de datos MedDRA, llegando hasta tener registrados 74865 efectos y 7593 fármacos[22].

### 2.3 Reconocimiento/Extracción de entidades a través de técnicas de Aprendizaje Automático

El aprendizaje automático ha sido parte fundamental del desarrollo del PLN en la subtarear del NER y la extracción de entidades. Es habitual el empleo de tipos de técnicas dentro del aprendizaje automático en este ámbito:

- Aprendizaje supervisado
- Aprendizaje no supervisado

A grandes rasgos, en el aprendizaje supervisado se trabaja con datos anotados, intentando encontrar una función que, dadas las variables de entrada, les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un conjunto de datos y así aprende a asignar la etiqueta de salida adecuada a un nuevo valor, prediciendo el valor

de salida. Sin embargo, en el caso del aprendizaje no supervisado no se disponen de datos anotados para el entrenamiento. Sólo se conocen los datos de entrada, pero no existen datos de salida que correspondan a una determinada entrada. A continuación, se describe con más exactitud su empleo en estudios en el ámbito de aplicación de este proyecto.

### 2.3.1 Aprendizaje supervisado

El aprendizaje supervisado entrena el algoritmo a partir de datos que han sido previamente etiquetados de manera manual. Cuando se etiqueta un dato, se le asigna una clase. A su vez, una clase representa una entidad que es la representación de un objeto o un concepto del mundo real.

De forma general cuando existe la disponibilidad de *corpus* anotados, se emplean los métodos supervisados en lugar de los no supervisados. Eso se debe a que cuanto más grande sea el conjunto de datos, mayor es la eficacia del algoritmo de aprendizaje automático. Mediante ellos se han extraído características sintácticas, semánticas y de sentimiento para clasificar los comentarios en textos biomédicos relacionados con RAM[24, 25]. Los enfoques tradicionales del empleo de métodos supervisados en el aprendizaje automático han sido el modelo oculto de Markov, campo aleatorio condicional (CRF, siglas en inglés) o de máxima entropía.

Sin embargo, no dejan de estar exentos de inconvenientes, así por ejemplo, cuando se emplea CRF bajo el aprendizaje supervisado se requieren grandes cantidades de datos de entrenamiento por lo que su utilidad en la práctica es limitada a ciertos ámbitos. Las limitaciones podrían ser un campo con la poca disponibilidad de datos anotados [26, 27], o por ejemplo la baja disponibilidad de estudios realizados en español en nuestro proyecto, entre los que destacan los trabajos realizados por Segura-Bedmar[21, 22]. Como consecuencia de ello, existe el problema que no podrían extraer todo el potencial de información de los textos y, por lo tanto, carecían de suficiente exactitud.

### 2.3.2 Aprendizaje no supervisado

En los primeros trabajos relacionados con la detección de efectos adversos de textos de redes sociales se emplearon *lexicones* desarrollados *ad-hoc* para un pequeño número de medicamentos estudiados. Estos métodos presentan ventajas ya que no necesitan un conjunto de datos entrenados por lo que es ideal para todos aquellos campos en los que no existe grandes cantidades de datos o campos muy limitados de investigación. A pesar de ello, presenta también desventajas entre las que destaca que los resultados pueden variar considerablemente si existen anomalías, por lo que habitualmente esta técnica se suele combinar con otros métodos que mejoren la eficacia.

Con el empleo de redes neuronales se mejoró la exactitud ya que cada palabra era insertado en un vector y cada mensaje es proyectado en una matriz, permitiendo emplear redes neuronales con diferentes arquitecturas y clasificadores. Otras estrategias para mejorar la exactitud incluyen el empleo de otras estrategias como redes convolucionales o el empleo de pesos para corregir el desequilibrio de los datos de origen[28]. A pesar de la mejoría, además de añadir técnicas de depuración de textos como *ngrams* o etiquetado *Part-of-Speech* (PoS), se han incorporado otro tipo de técnicas más “profundas” que tienen como objetivo comprender todo el significado de una frase más que enfocarse en un segmento del texto.

Particularmente importante representa una nueva herramienta llamada *QuickUMLS*, que es un método no supervisado que en lugar de tener parámetros que condicionan su resultado, depende de un conocimiento externo que prediga las etiquetas. Esto implica que la calidad del modelo está limitado por la calidad de la base del conocimiento, con lo que el modelo no puede predecir aspectos que la base del conocimiento no contiene. Otro aspecto que redundaría en la pérdida de la exactitud sería que el modelo puede generalizar más allá de los datos anotados. Esto es particularmente importante en el caso de la detección de RAM en textos de redes sociales, ya que las palabras no sólo podrán ser caracterizadas como efectos adversos sino que también pueden ser indicaciones u otras situaciones.

### 2.3.3 Espacios vectoriales

Con el objetivo de encontrar una buena representación matemática de las palabras, se han empleado modelos basados en espacios vectoriales en los que las palabras representan puntos en un espacio euclídeo de  $n$  dimensiones donde palabras con significado similar tendrán que estar más cerca que otras con ningún significado en común. A partir de este modelo, surge *word embeddings* (o incrustaciones de palabras) que es un vector numérico que representa un espacio dimensional reducido. El objetivo de *word embeddings* es reducir la dimensionalidad de los textos analizados, utilizar una palabra para predecir las palabras alrededor de ella o incorporar el significado semántico de la palabra.

El resultado de *word embeddings* es emplearlo en modelos de aprendizaje automático (entrenamiento o inferencia) y representar o visualizar patrones en el *corpus* utilizado para entrenarlos. Mediante la extrapolación de las técnicas empleadas en *word embeddings* se puede realizar también en frases (*sentence embeddings*) o documentos completos (*document embeddings*). De entre los modelos más empleados en *word embeddings* se puede decir que son *Word2Vec*, *FastText*, *Glove* y ,más recientemente, *BERT*.

Otra técnica relacionada con el espacio vectorial es el método TF-IDF (Term Frequency-Inverse Document Frequency) en el que su vectorización implica calcular TF-IDF para cada una de las palabras de un *corpus* y luego presentar esa información en un vector. Sin embargo, a diferencia de otras técnicas de *word embeddings* no tiene en cuenta el entorno de la palabra. En ocasiones, incluso se pueden emplear una combinación de estos modelos como *Word2Vec* con TF-IDF ya que no son técnicas mutuamente exclusivas, de forma que se puedan capturar relaciones semánticas sutiles mediante la combinación de ambos.

### 2.3.4 Enfoque seleccionado



## Antecedentes

Para el desarrollo de este proyecto, se ha decidido emplear un enfoque híbrido basado primero en el aprendizaje no supervisado ya que es ideal para el problema que se nos plantea: un conjunto de textos desestructurados junto con la ausencia de textos anotados en español para la extracción de entidades. Sin embargo, debido a la posible pérdida de exactitud que podría suceder en la etapa del NER, se procederá a la limpieza de textos y preprocesamiento para aumentar la exactitud y la adecuada extracción de entidades.

Por otro lado, una vez obtenidas las entidades se debe extraer los efectos adversos para conocer si existe una relación entre las diferentes clases de entidades. Debido a ello, es necesario generar las características sintácticas y semánticas de las entidades relacionadas en el corpus. De esta forma, para la generación de características se empleó TF-IDF solo y combinado con *Word2Vec*. Las razones de esto se podrían entender por:

- *Word2Vec* genera un vector multidimensional que intenta establecer una relación de palabras entre ellas, mientras TF-IDF no captura el significado.
- TF-IDF no requiere datos externos para el entrenamiento a diferencia de *Word2Vec*.
- TF-IDF es ideal para grandes documentos y con muchas palabras ya que emplea una menor cantidad de memoria.

En la **tabla 1** se indica el algoritmo empleado en esta etapa de clasificación:

<i>Algoritmo estadístico de aprendizaje</i>
---

Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

<b>Entrada:</b> todas las instancias con al menos un par de términos biomédicos y fármacos/medicamento $R(\text{fármaco}, \text{término})$
<b>Salida:</b> donde las instancias tienen una relación efecto adverso y fármaco/medicamento
<b>Procedimiento:</b> <ol style="list-style-type: none"><li>1. Para cada instancia <math>R(\text{fármaco}, \text{término})</math>: Características: Extracción mediante TF-IDF solo o más <i>Word2Vec</i>.</li><li>2. Separar las instancias de la relación entre los conjuntos de entrenamiento y test.</li><li>3. Entrenar un clasificador basado en el conjunto de entrenamiento</li><li>4. Utilizar el clasificador para clasificar instancias en el set de prueba en dos clases <math>R(\text{fármaco}, \text{término}) = \text{positivo}</math> y <math>R(\text{fármaco}, \text{término}) = \text{negativo}</math></li></ol>

Tabla 1. Algoritmo estadístico de aprendizaje

Como resumen de todas las etapas que se van a desarrollar en este proyecto a grandes rasgos, se puede considerar que la detección de efectos adversos a través de las redes sociales sigue las etapas descritas en la **figura 1**.

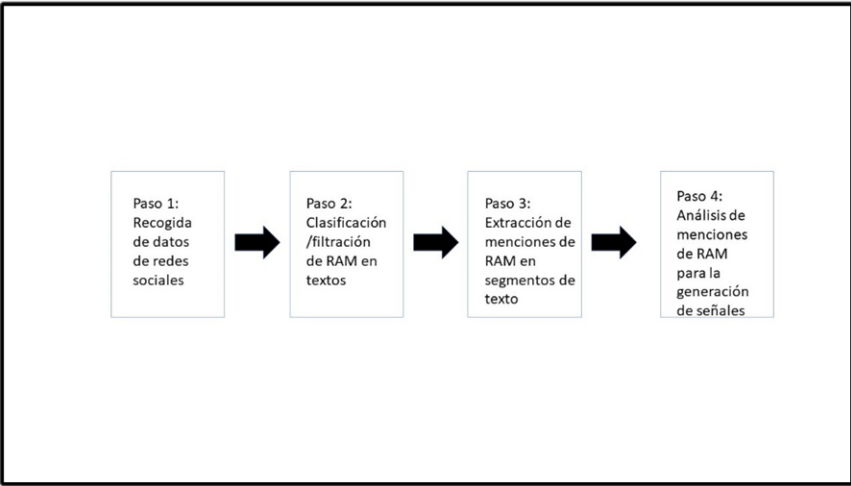


Figura 1. Pipeline de detección de efectos adversos

### 3. Planteamiento del problema

#### 3.1 Teoría de la extracción de conceptos de texto no estructurados

El problema relacionado con la extracción de conceptos de documentos no estructurados ha sido formalizado de la siguiente manera[29] :

Sea un diccionario  $S$  un conjunto de cadenas,  $C$  una colección de conceptos, de forma que un mapa asocia un concepto de la colección a uno o más cadenas en el diccionario.

$$C: C \rightarrow S$$

Dado un documento  $d$  que podemos representar como un orden secuencial de *tokens*  $\{d_1, ..., d_n\}$ , una función de similitud  $strsim$  y un umbral de similitud  $\alpha \in [0,1]$ , el algoritmo de extracción de conceptos debería ser:

$$\{(d_{ij}, c_k) \mid \exists s_h \in C(c_k) \text{ s.t. } strsim(d_{ij}, s_h) \geq \alpha\}$$

donde  $d_{ij}$  representa una secuencia de *tokens* en  $d$  y  $s_h \in S$  es una cadena representando un concepto  $c_k \in C$ .

El problema de emplear un diccionario que pueda coincidir con una determinada cadena puede formularse de la siguiente manera:

Dada una cadena objetivo  $x$ , un umbral de similitud  $\alpha$ , un diccionario  $S$  y una función de similitud  $strsim$ , se desea encontrar el subconjunto  $\mathcal{Y}_{x,\alpha} \subseteq S$ , tal que:

$$\mathcal{Y}_{x,\alpha} = \{y \in S \mid strsim(x, y) \geq \alpha\}$$

Una solución de este problema podría ser computar la similitud de cada cadena en el diccionario  $S$  hasta conseguir la cadena objetivo  $x$ . Sin embargo, el coste computacional sería muy caro ya que tendríamos una complejidad dependiente del

tamaño del diccionario  $S$ . Esto es especialmente importante cuando se pretende emplear un diccionario como UMLS que puede contener más de 6 millones de cadenas según el idioma elegido.

Esto se ha podido solventar mediante el empleo de una herramienta como QuickUMLS cuyo mecanismo se detalla a continuación:

Dado un documento  $d$  de una longitud  $n$ , un umbral de similitud  $\alpha$  y un tamaño de ventana  $w$ , QuickUMLS genera para cada token  $d_1 \in d$ , todas las posibles secuencias de token  $d_{ij} = \{d_i, \dots, d_j\}$ ,  $j \in \{i, \dots, i+w-1\}$ . Luego, mediante heurística, determina si la secuencia  $d_{ij}$  es una secuencia válida de tokens. Si así lo es, se procede al siguiente paso de identificar cadenas en  $S$  que sean similares a  $d_{ij}$ . Una vez que el subconjunto de todas las posibles cadenas coincidentes se determine, QuickUMLS selecciona el subconjunto más apropiado:

$$\mathcal{Z}_{d,\alpha} = \bigcup_{d_{ij}} (\mathcal{Y}_{d_{ij},\alpha})$$

En cuanto a la evaluación de los resultados obtenidos mediante técnicas de detección/extracción de efectos adversos, existen dos tipos de evaluaciones:

- Cualitativa: Comparando tipos de medicamentos, según la gravedad de sus efectos adversos[6].

- Cuantitativa: Se han empleado métricas como precisión, exactitud, exhaustividad o *f-score*. Estas métricas se emplearon en los primeros trabajos que empleaban aprendizaje supervisado y, posteriormente, con la inclusión del aprendizaje no supervisado se ha empleado para comparar los resultados de los dos enfoques[29].

### 3.2 Herramientas utilizadas

Python ha sido el lenguaje de programación empleado en este proyecto, en concreto la versión 3.9.7. Considerando el objetivo principal investigador del proyecto y las opciones valoradas en cuanto al lenguaje de programación, se ha optado por Python frente a Java.

## Planteamiento del problema

Aunque la discusión de qué lenguaje es mejor está fuera del ámbito de este proyecto, se ha seleccionado Python debido a su sencillez, en la obtención de resultados rápidos con diferentes estrategias y en la amplia disponibilidad de librerías relacionadas con aprendizaje automático y PLN.

Las librerías empleadas en este proyecto han sido:

- NumPy: librería especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos.
- Pandas: librería especializada en el manejo y análisis de estructuras de datos.
- SpaCy: librería que permite construir aplicaciones de PLN, proporcionando modelos preentrenados de diferentes idiomas. SpaCy es un software comercial de código abierto bajo licencia de Massachusetts Institute of Technology (MIT).
- Natural Language Tool Kit (NLTK): librería que ofrece una variedad de capacidades de manipulación de cadenas. Contiene un gran repositorio de plantillas con modelos de análisis de texto, gramáticas basadas en características y ricos recursos léxicos para construir un modelo de lenguaje completo.
- Pyinstaller: librería que transforma todos tus archivos Python en un solo paquete. Es ideal para distribuir la aplicación desarrollada ya que el usuario sólo tiene que instalar la aplicación y no requiere la instalación de más módulos.
- InnoSetup: sistema de instalación en un entorno Windows.

A lo largo de las etapas del proyecto cuando se empleen por primera vez cada librería se indicará la versión utilizada.

### 3.3 Construcción de diccionario de medicamentos

En la construcción del diccionario de medicamentos se emplearon fuentes que fuesen completas, de fácil acceso y que contuviesen los medicamentos autorizados en España. Aunque existen varias de ellas que cumplen los dos primeros requisitos, en ocasiones no contemplan los medicamentos autorizados sólo en España y añaden los autorizados en otros países de habla hispana o en Estados Unidos.

Después de este filtrado, se consideraron dos fuentes principales de información:

- CIMA[30]
- Vademecum[31]

### 3.3.1 CIMA

CIMA es una fuente de información respecto a todos los fármacos y medicamentos autorizados y comercializados en España. Esta base de datos es creada, mantenida y actualizada por parte de la AEMPS, que es un organismo autónomo que entre otras tareas se encarga de la autorización, mantenimiento o revocación de la comercialización de todos los medicamentos existentes en España. Está disponible tanto para los profesionales sanitarios y el público general a través de su página web, e incluye entre su información: nombre del medicamento, principios activos, excipientes, código nacional, número de registro de la autorización, fecha de autorización de la comercialización, laboratorio responsable, forma farmacéutica, vía de administración, forma de la presentación, estado actualizado de la comercialización (autorizado/temporalmente suspendido/revocado), códigos *Anatomical Therapeutic Chemical* (ATC) , condiciones de prescripción y dispensación, ficha técnica y prospecto del paciente.

Además de toda esta información, se encuentra disponible un recurso llamado “Base de datos completa con el Nomenclator de prescripción” en el que se encuentra toda la información disponible en forma de archivos xml, son 15 archivos en los que cada archivo contiene información de medicamentos clasificada por ATC, denominación

## Planteamiento del problema

comercial, forma farmacéutica, laboratorios, principios activos, situación del registro, etcétera.

```
<atc>
  <nroatc>3</nroatc>
  <codigoatc>A01A</codigoatc>
  <descatc>A01A - PREPARADOS ESTOMATOLÓGICOS</descatc>
</atc>
<atc>
  <nroatc>4</nroatc>
  <codigoatc>A01AA</codigoatc>
  <descatc>A01AA - Agentes para la profilaxis de las caries</descatc>
</atc>
<atc>
  <nroatc>5</nroatc>
  <codigoatc>A01AA01</codigoatc>
  <descatc>A01AA01 - Fluoruro de sodio</descatc>
</atc>
<atc>
  <nroatc>6</nroatc>
  <codigoatc>A01AA02</codigoatc>
  <descatc>A01AA02 - Monofluorofosfato de sodio</descatc>
```

Figura 2. Ejemplo de entrada del archivo *DICCIONARIO\_ATC.xml*

```
<principiosactivos>
  <nroprincipioactivo>3219</nroprincipioactivo>
  <codigoprincipioactivo>2A</codigoprincipioactivo>
  <principioactivo>RANITIDINA</principioactivo>
</principiosactivos>
<principiosactivos>
  <nroprincipioactivo>3220</nroprincipioactivo>
  <codigoprincipioactivo>2CH</codigoprincipioactivo>
  <principioactivo>RANITIDINA HIDROCLORURO</principioactivo>
</principiosactivos>
<principiosactivos>
  <nroprincipioactivo>5600</nroprincipioactivo>
  <codigoprincipioactivo>5A</codigoprincipioactivo>
  <principioactivo>RIBES NIGRUM</principioactivo>
</principiosactivos>
<principiosactivos>
  <nroprincipioactivo>6380</nroprincipioactivo>
  <codigoprincipioactivo>9A</codigoprincipioactivo>
  <principioactivo>RETINOL</principioactivo>
</principiosactivos>
<principiosactivos>
  <nroprincipioactivo>6381</nroprincipioactivo>
  <codigoprincipioactivo>9AC</codigoprincipioactivo>
  <principioactivo>RETINOL ACETATO</principioactivo>
</principiosactivos>
<principiosactivos>
  <nroprincipioactivo>8343</nroprincipioactivo>
  <codigoprincipioactivo>9B</codigoprincipioactivo>
  <principioactivo>VITAMINA A</principioactivo>
</principiosactivos>
```

Figura 3. Ejemplo de entrada del archivo *DICCIONARIO\_PRINCIPIOS\_ACTIVOS.xml*

A fecha de Marzo'2022, la base de datos contiene 41172 presentaciones comerciales con 24048 medicamentos únicos. Debido a esta enorme y exhaustiva información, este recurso ha sido el preferido para obtener el diccionario de medicamentos.

Con el objetivo de eliminar información no importante en este proyecto, se hizo un filtrado para eliminar cualquier información superflua que no contenga ni principios activos ni nombres comerciales del medicamento. Esta información, que se consideró superflua correspondía, como se puede observar en la **figura 4**, a que el nombre comercial contiene información tal como la dosis, forma farmacéutica, método de administración o presentación comercial. Así quedaron nombres de presentaciones comerciales únicas la igual que los principios activos sin tener en cuenta otras salvedades.

ALLYNAT 240 mg COMPRIMIDOS RECUBIERTOS, 100 comprimidos
ALLYNAT 240 mg COMPRIMIDOS RECUBIERTOS, 20 comprimidos
ALLYNAT 240 mg COMPRIMIDOS RECUBIERTOS, 40 comprimidos
ALMAX 1g/7,5 ml SUSPENSION ORAL , 1 frasco de 225 ml
ALMAX 500 mg COMPRIMIDOS MASTICABLES , 30 comprimidos
ALMAX FORTE 1,5 g SUSPENSION ORAL , 16 sobres
ALMOGRAN 12,5 mg COMPRIMIDOS RECUBIERTOS CON PELICULA , 3 comprimidos

*Figura 4. Ejemplo de estructura en CIMA de nombres comerciales de medicamentos*

### 3.3.2 Vademecum

El sitio web que corresponde a [www.vademecum.es](http://www.vademecum.es) es una guía de productos farmacéuticos que se publica y actualiza de forma periódica, y en la cual aparece recopilada la información otorgada por las compañías farmacéuticas. El Vademécum es un instrumento dirigido exclusivamente a los profesionales médicos y que es distribuido por la compañía editora únicamente a dichos profesionales. Presenta un buscador propio con diferentes índices ya sea por indicaciones, principio activos, marca comercial,



## Planteamiento del problema

laboratorio o por clasificación ATC. Si bien presenta un catálogo muy extenso y muy completo, en nuestro caso sólo sirvió para obtener una información complementaria que nos daba CIMA.

La información disponible en este sitio web aunque gratuita requirió el empleo de técnicas de *webscraping*. El webscraping es una técnica que se utiliza para obtener de forma automática el contenido que hay en páginas web a través de su código HTML. El uso de estas técnicas tienen como finalidad recopilar grandes cantidades de datos de diferentes páginas web cuyo uso posterior puede ser muy variado, aunque en el caso de este proyecto fue construir un diccionario de medicamentos y principios activos. :

Para ello se emplearon las siguientes herramientas:

- Selenium WebDriver: Esta herramienta permite a los desarrolladores probar y registrar las interacciones con una aplicación web y luego repetirlas las veces que se desee, de forma completamente automática. Selenium engloba un número amplio de proyectos de software libre empleados para automatización del desarrollo web, que soporta diferentes lenguajes, entre ellos el lenguaje de programación de este proyecto (Python). Mediante una API como WebDriver permite al desarrollador simular las interacciones con cualquier navegador ya sea Firefox, Chrome, Edge, Safari o Internet Explorer. Desde 2018, la API es un estándar W3C oficial.

- BeautifulSoup (bs4): Se trata de una librería que facilita extraer información de las páginas web. Extrae información de archivos HTML, XML y otros lenguajes de marcado. Esta librería ayuda a descargar el contenido que interesa de una página web y lo almacena sin necesidad de guardar también todo el contenido de HTML.

- urllib2: Se trata de un módulo Python para acceder y utilizar recursos de internet identificados por *Uniform Resource Locators* (URLs). Ofrece una interfaz muy simple, a través de la función `urlopen`. Esta función es capaz de acceder a URLs usando una variedad de protocolos diferentes.

Posteriormente, se eliminaron duplicidades de principios activos, así como aquellas presentaciones comerciales que tuviesen diferentes sufijos que corresponden a formas

de liberación como flas, retard o prolong fueron eliminadas para evitar la reiteración de marcas comerciales o falsos positivos en el posterior minado de datos de la red social.

Una vez comparados la información tanto de CIMA como la obtenida de Vademecum se obtuvieron 5636 medicamentos y principios activos que constituyeron el diccionario de fármacos de este proyecto.

### 3.4 Construcción de diccionario de términos médicos

Al igual que en la construcción del diccionario de medicamentos, se emplearon fuentes que fuesen completas, de fácil acceso y que contuviesen los términos en español. Desgraciadamente en español no existen tantas bases de datos tan exhaustivas como en medicamentos.

Aunque existen diferentes plataformas, se optó por dos fuentes de términos médicos:

- MedDRA (español)-MDRSPA[32]
- Diccionario de términos médicos de la Real Academia Nacional de Medicina[33]

#### 3.4.1 MedDRA- (español)

Como se ha comentado anteriormente, MedDRA tiene la ventaja que es multilingüe de forma que cada país puede trabajar en su idioma oficial. Cada término tiene asociado un código que es el mismo con independencia del idioma .

Es un vocabulario con una estructura jerarquizada, compuesto por cinco niveles, desde un nivel más general hasta otro más específico, llamado LLT (*Lowest Level Terms*). Estos términos se utilizan para definir hallazgos en la práctica clínica. En el nivel siguiente llamado PT (*Preferred Terms*), donde cada término es un concepto médico individual que puede ser un síntoma, signo, diagnóstico de enfermedad, indicación terapéutica, investigación, procedimiento quirúrgico o médico y, por último, historia familiar o

## Planteamiento del problema

sociomédica. La relación de PT con LLT es que cada LLT está relacionado con un PT pero cada PT está relacionado con al menos un LLT.

Subiendo en la jerarquía, se encuentran los HLT (*High Level Terms*) que están relacionados con los PT ya sea por anatomía, patología, fisiología, etiología o función. A su vez, los HLT se agrupan en HLGT (*High Level Group Terms*). Por último, estos HGLT se agrupan en SOC (*System Organ Classes*), basados en etiología, sitio de manifestación o propósito. Como ejemplo de la exhaustividad de este vocabulario se puede ver el número total de términos en la **tabla 2**. Es, por ello, que al ser utilizado como diccionario de clasificación de efectos adversos por el *International Conference of Harmonisation* (ICH) ha sido elegido como principal diccionario de términos médicos. Como principal inconveniente es que al emplear redes sociales que utilizan un lenguaje informal y no dado a términos científicos, se podrían infraestimar muchos efectos adversos indicados por los usuarios.

ID Tipo Semántico	Nombre del Tipo Semántico	Términos
T047	Enfermedad o Síndrome	18471
T033	Hallazgos	10314
T191	Proceso neoplásico	5771
T037	Lesión o intoxicación	4576
T061	Procedimiento terapéutico o preventivo	4277
T046	Función patológica	3535
T184	Signo o síntoma	2314
T059	Procedimiento de laboratorio	1800
T048	Disfunción mental o de comportamiento	1542
T034	Resultado de laboratorio o de pruebas	1418
T019	Anormalidad congénita	1210
T060	Procedimiento diagnóstico	1141

*Tabla 2. Distribución de tipos semántica de MedDRA*

### 3.4.2. Diccionario de términos médicos de la Real Academia Nacional de Medicina

La Real Academia Nacional de Medicina ha elaborado un diccionario de términos médicos. Mediante técnicas de *webscraping* de la misma forma que se realizó para la

## Planteamiento del problema

creación del diccionario médico, se obtuvo un conjunto de términos. Se han empleadas las mismas herramientas y librerías empleadas anteriormente en el diccionario de medicamentos.

Este diccionario de términos contiene 109706 términos, aunque no presenta una estructura jerarquizada ni tan exhaustiva como la que presenta MedDRA y se pensó en ser utilizado como fuente complementaria a esta última para mejorar la detección de efectos adversos. Tras su uso y debido a la alta generación de falsos positivos, se descartó como fuente de diccionario de términos médicos.

### 3.5 Minado de mensajes en Twitter

Tal y como se ha comentado en la Introducción, Twitter ha sido la red social seleccionada para la recolección de los mensajes de usuarios. Durante dos meses se procedió a hacer una búsqueda de los mensajes de los usuarios. Se emplearon dos librerías fundamentalmente:

- Tweepy: Es una de las librerías más populares para acceder a Twitter. Se accede, previa creación de cuenta de desarrollador de Twitter y obtención de credenciales. Tiene como principal desventaja un límite de recolección de 3200 tweets y sólo permite hacer recolección de tweets que tengan menos de una semana.

- Snsrape: Se trata de una herramienta para *scraping* especializada en redes sociales. Con esta herramienta es posible capturar elementos como los perfiles del usuario, hashtags, búsquedas o mensajes relevantes. Se puede emplear en muchas redes sociales como Facebook, Instagram, Telegram, Twitter,...Entre los atributos que puede rescatar de un tweet los más destacables para este proyecto fueron la localización geográfica, fecha, texto del mensaje, número de identificación del tweet, usuario. Además con respecto a Tweepy, presentaba como ventaja fundamental el poder realizar búsquedas sin importar la fecha ni el límite de una semana, además había un número ilimitado de tweets que se podían analizar. Otra ventaja sustancial es que mediante esta herramienta se podían seleccionar la eliminación de retweets ya que podían perturbar los hallazgos en las fases posteriores del proyecto. Aunque en un principio se optó por Tweepy, al final se obtuvieron una mayor cantidad y calidad

## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

mediante Snsrape, por lo que todos los mensajes obtenidos fueron mediante esta herramienta. La búsqueda se realizó utilizando el diccionario de medicamentos y fármacos y empleando como intervalo de tiempo 2 meses. Finalmente se obtuvieron 1029001 tweets (previos al preprocesamiento) que contenían en algún espacio del tweet una referencia a alguno de los medicamentos o principios activos incluidos en nuestro diccionario. En la **tabla 3** se indican las menciones de las primeros 30 marcas comerciales detectadas.

MARCA COMERCIAL	APARICIONES
IBUPROFENO	34055
RIVOTRIL	29427
ASPIRINA	24122
CLONAZEPAM	20170
OMEPRAZOL	15535
HIDROXICLOROQUINA	8536
AZITROMICINA	8369
DEXAMETASONA	6799
TRAMADOL	6596
DIAZEPAM	6538
LORATADINA	6289
FOLICO	6204
MORFINA	6019
ENEAS	5758
NAPROXENO	5668
VALIUM	5159
BUSCAPINA	4902
EVISTA	4445
AILYN	4354
DALSY	4310
BCG	4043
PROZAC	4022
ALPRAZOLAM	3926
DICLOFENACO	3919
SERTRALINA	3907
AAS	3878
MISOPROSTOL	3739
MIDAZOLAM	3695
PARACETAMOL	3649
NOLOTIL	3565

*Tabla 3. Clasificación de marcas comerciales según apariciones en la red social*

De acuerdo a las menciones según la marca comercial, hacemos una distribución según ATC como refleja la **figura 5**:

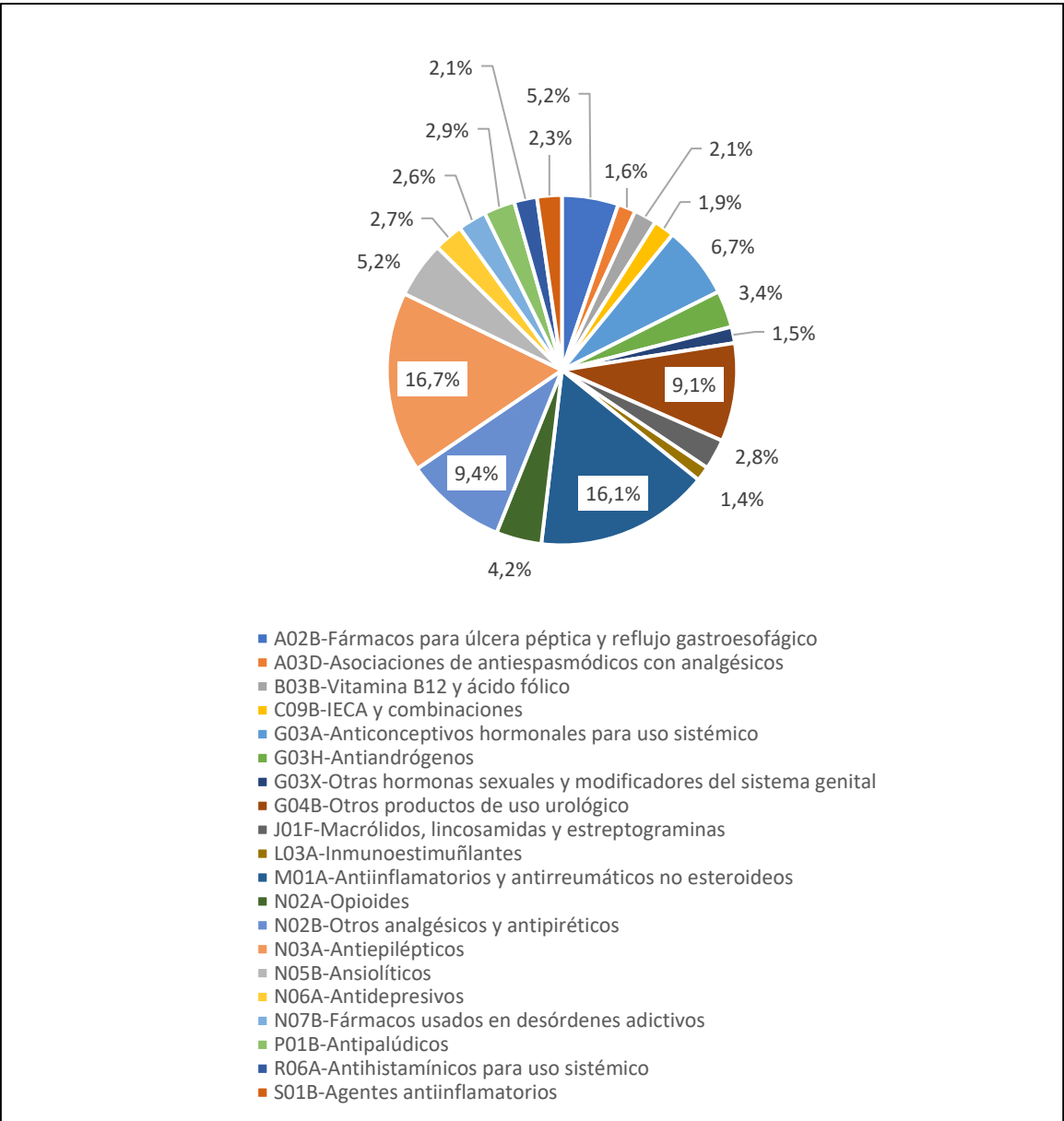


Figura 5. Distribución de menciones de marcas comerciales agrupadas por ATC

En la **tabla 3** se indican las menciones según el principio activo, así como en la **figura 6** se encuentran la distribución de menciones agrupadas según la clasificación ATC, donde prácticamente son similares a las aparecidas por marca comercial.

Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

PRINCIPIO ACTIVO	APARICIONES
IVERMECTINA	38668
IBUPROFENO	34055
CLONAZEPAM	20170
OMEPRAZOL	15535
HIDROXICLOROQUINA	8536
AZITROMICINA	8369
REMDESIVIR	7873
DEXAMETASONA	6799
TRAMADOL	6596
DIAZEPAM	6538
LORATADINA	6289
FOLICO	6204
MORFINA	6019
NAPROXENO	5668
BCG	4043
ALPRAZOLAM	3926
DICLOFENACO	3919
SERTRALINA	3907
MISOPROSTOL	3739
MIDAZOLAM	3695
PARACETAMOL	3649
PREDNISONA	3418
METFORMINA	3242
FENTANILO	3135
SALBUTAMOL	3058
FLUOXETINA	3029
QUETIAPINA	2992
KETOROLACO	2915
ATRACURIO	2788
MINOXIDIL	2705

*Tabla 4. Clasificación de principios activos según apariciones en la red social*



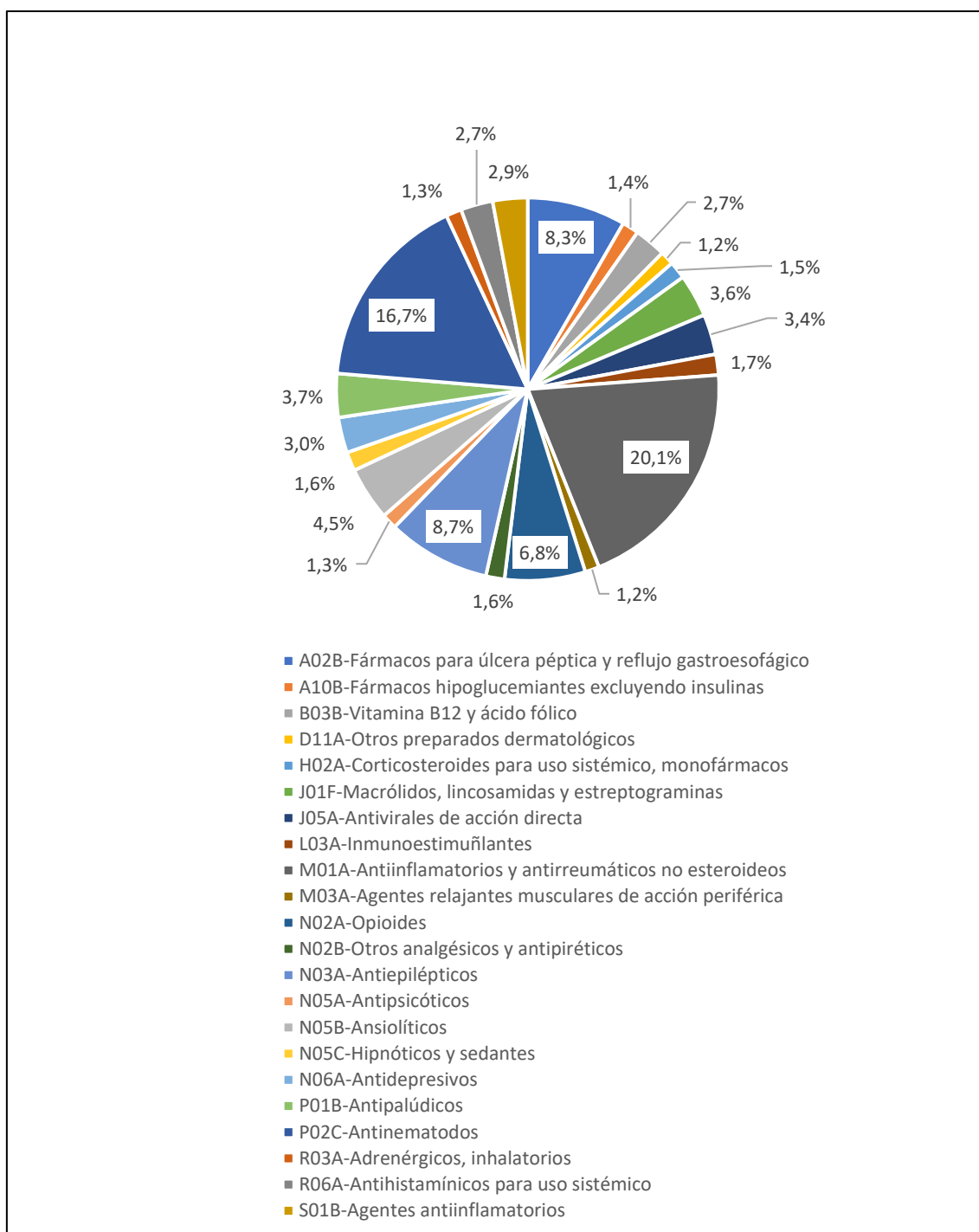


Figura 6. Distribución de menciones de principios activos agrupadas por ATC

A continuación, en la **figura 7** se exponen mensajes extraídos de la red social (se han borrado los nombres de usuarios):

1372451868870332420," Trastornos de pánico, con problemas de adaptabilidad al cambio, imagínate el año que llevo. Además la sanidad y la sociedad, con 5 años me mandaron tranxilium pediátrico para poder dormir los días que tenía un examen. A los 5 años."

1358706841866424322,"Por el Tranxilium y la Quetiapina me ha salido un sarpullido en la espalda y lo estoy odiando muchísimo. Que asco de efectos secundarios, pero claro, como para no tenerlos si tomo casi la cantidad máxima diaria recomendada"

1375016053575716867," Y la doxilamina, tan deseada y necesitada en este último año, un antihistamínico con indicación para el insomnio debido a su efecto secundario "

1372100799900303361,"Me habían recetado antes Clonazepam, pero tampoco funciona. Ya en crisis de ansiedad la opción era Diazepam vía intravenosa, una dosis alta que me mandaba a dormir."

*Figura 7. Ejemplos de mensajes extraídos de la red social Twitter*

## 4. Descripción del sistema desarrollado

### 4.1 Esquema de la solución

El sistema se inicia con los textos de los tweets que sufren una etapa de limpieza para intentar eliminar el ruido que afecta a las fases posteriores perjudicando la etapa final de extracción de entidades. Tras finalizar cada una de las fases, el resultado final será la extracción de entidades de forma que sea posible detectar la relación entre efectos adversos y medicamentos. En la **figura 8** se muestra el pipeline correspondiente al proyecto desarrollado:

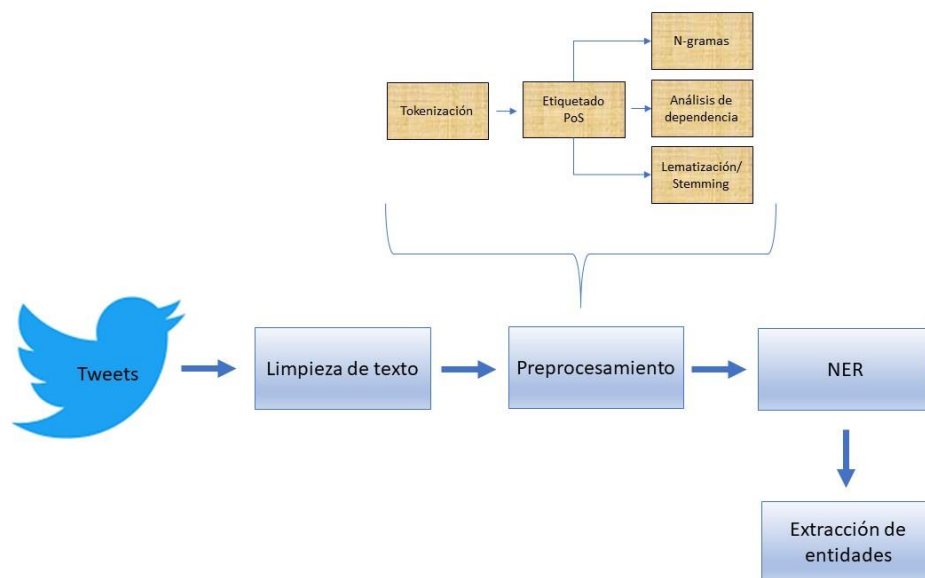


Figura 8. Pipeline PLN para detección de efectos adversos

### 4.2 Limpieza de texto

Una vez obtenidos los mensajes, es necesaria la limpieza de los mensajes mediante una etapa de preprocesamiento de los textos obtenidos, para evitar los posibles errores de codificación.

Esta primera etapa se hizo con métodos generales (un ejemplo se puede encontrar en la **figura 9**), incluidas en funciones del archivo **preproc.py**, que son utilizadas para el procesamiento del archivo mediante **preproc\_tweets.py**:

- Pasar texto a minúsculas (función minúsculas)
- Eliminación de datos de usuario o de respuestas a un usuario en concreto (función eliminaArroba).
- Eliminación de comillas (función eliminaComillas)
- Eliminación de direcciones url (función eliminaWeb).
- Eliminación de hashtags (función eliminaHashtag).
- Sustitución de exclamaciones y de interrogaciones con más de un carácter (función sustituyeMultiExclamaciones y sustituyeMultiInterrogacion)
- Eliminación de asteriscos (función eliminarAsteriscos)
- Sustitución de *emojis* (función sustituyeEmoji).
- Eliminación de espacios en blanco (función eliminarWhiteSpace)
- Eliminación de guiones (función eliminarGuiones)
- Eliminación de repetición de 3 o más caracteres dentro de una cadena.  
(función eliminarRepeticiones)
- Eliminación de otros signos de puntuación, como – o [] (función encontrarCorchetes, eliminarCorchetes, eliminarParentesis)

Mediante estos métodos por ejemplo se consiguió que los mensajes que mencionaban *rivotril* pasaran de 297332 *tokens* a 267161 *tokens*.

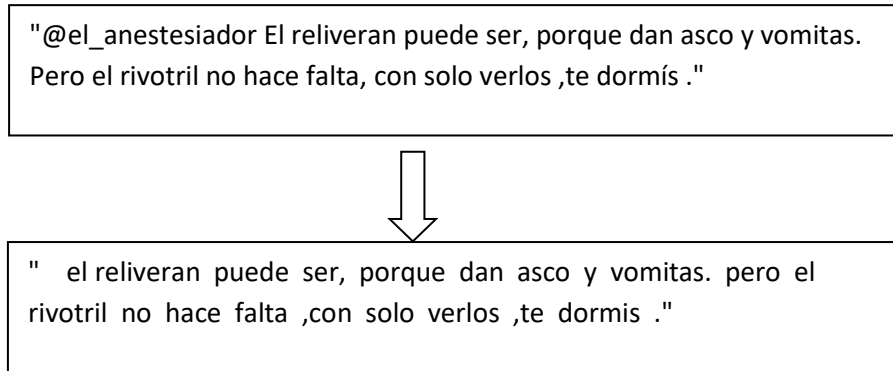


Figura 9. Ejemplo de preprocesamiento de tweets con eliminación del usuario

Posteriormente, se valora si los tweets siguen conteniendo “impurezas” en esta etapa que pueden consistir en espacios en blanco duplicados como pueden ser signos de puntuación inadecuados, direcciones web o *hashtags*... Por ejemplo, en el análisis de la limpieza de los textos que contienen “rivotril” se obtuvo una limpieza final de aproximadamente de 0.0260%. Habitualmente, se considera que un texto es apto para las siguientes fases del procesamiento cuando está a un nivel inferior de impureza del 5%.

Uno de los grandes problemas en esta etapa son los *emojis*, no existe una óptima solución al problema que representa, ya que si se eliminan se puede perder información y, por lo tanto, exactitud en el proceso final y en la extracción de información. Por otro lado, la posibilidad de transformar esos símbolos en unicode es posible y de ahí extraer su significado de acuerdo a las tablas normalizadas[34].

Considerando las limitaciones de analizar todos los *emojis* de forma exhaustiva, se consideró que aquellos *emojis* que afectasen al “sentimiento” de un mensaje, deberían ser traducidos a su correspondiente código y su significado de acuerdo a las tablas normalizadas. Aquellos *emojis* que sólo reflejan lugares, género o figuras geométricas fueran eliminados al no aportar ningún tipo de información emocional o de síntomas y signos clínicos.

En la **tabla 5** se observan ejemplos de procesamiento de *emojis*:



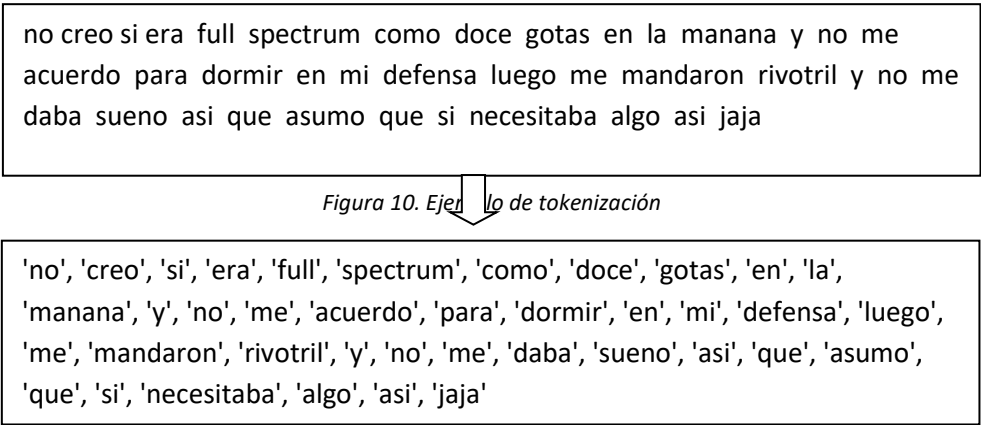
Emoji	Unicode	Significado	Resultado
	2196 FE0F	“up-left arrow”	(eliminado)
	1F620..1F625	“angry face”	enfadado

Tabla 5. Ejemplos de preprocesamiento de emojis

4.3 Preprocesamiento

4.3.1 Tokenización

La tokenización consiste en segmentar el texto en tokens, en este caso, palabras y signos de puntuación. Un ejemplo de tokenización se puede ver en la **figura 10**:



Tras el preprocesamiento o limpieza de los datos, la tokenización corresponde a la primera fase para la detección y es fundamental para posteriores fases como el etiquetado Part-of-Speech (*PoS tagging*). En esta fase y en próximas fases del procesamiento se ha empleado la librería SpaCy. Esta librería soporta la tokenización y el entrenamiento para más de 60 lenguajes, incluyendo el español. Trabaja con modelos

## Descripción del sistema desarrollado

neuronales para el etiquetado, *parsing*, reconocimiento de entidades y clasificación de textos, además de otros sistemas de entrenamiento y gestión del flujo de trabajo. La versión empleada en este proyecto es 3.2.4.

SpaCy como librería presenta una *pipeline* como se indica en la **figura 11**, de forma resumida se pueden considerar:

1. *Tokenizer*(tokenizador): divide el texto entrante en tokens, estos tokens pueden ser palabras, signos de puntuación o cualquier espacio en blanco.
2. *Tagger*(etiquetado gramatical): aplica un modelo estadístico al texto de entrada. Analiza la función morfológica de cada token que compone el texto original.
3. *Dependency parsing*(análisis sintáctico): Descubre las dependencias sintácticas de la frase.

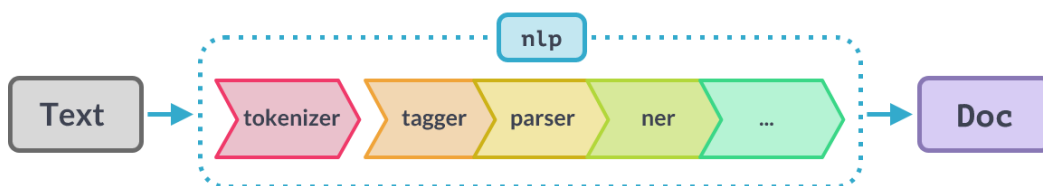


Figura 11. Componentes de la librería SpaCy (Fuente: SpaCy.io)

Sin embargo, uno de los problemas que ya ha sido referido es la falta de disponibilidad de modelos en español, eso hace que la tokenización mediante algún modelo en español no tenga disponible un *tokenizer*. En concreto, en español existen dos modelos ampliamente utilizados como son “es\_core\_news\_sm” y “es\_core\_news\_md”, la diferencia entre ambos radica en que “sm” se refiere a modelos reducidos, más rápidos pero menos precisos. La referencia “md” son modelos medios entrenados con mayor datos y mayor precisión. Por todo ello, se decide emplear como modelo “es\_core\_news\_md” version 3.2.0[35].

### 4.3.2 Lematización/Stemming

Otra etapa dentro del procesamiento, representa los procesos de lematización y su alternativa llamada *stemming*. Con el objetivo de entender estos dos conceptos se podría explicar con que los textos gramaticalmente utilizan diferentes formas de una

palabra, así por ejemplo, tenemos palabras como sanar, sana y sanando. Además, existen familias de palabras relacionadas con significados similares, siguiendo con el ejemplo anterior, podríamos considerar sanidad o sanitario. Por ello, se podría considerar como un proceso útil la búsqueda de una palabra que te devolviese otra palabra con una raíz similar. Por ello, ambos procesos como la lematización o el *stemming* tienen como objetivo reducir una palabra a una forma básica común que la relaciona con otras palabras, es decir:

sanar, sana, sanando  $\Rightarrow$  san

En un texto tras aplicar la búsqueda de esa raíz podría ser de la forma:

La herida esta en proceso de sanar  $\Rightarrow$  La herida esta en proceso de san

Sin embargo, estos dos procesos se diferencian en el proceso. *Stemming* se refiere a un proceso heurístico que habitualmente recorta los finales de las palabras para lograr su objetivo, que resulta en la mayor parte de los casos de la eliminación de los afijos. La lematización, en cambio, utiliza habitualmente un vocabulario y un análisis morfológico de las palabras, que tiene como resultado eliminar los finales de palabra para obtener una base llamada *lema* (en inglés, *lemma*). Habitualmente, estos procesos se realizan mediante extensiones o componentes de un *pipeline*, ya sean comerciales o de código abierto.

Ambos procesos tienen efectos beneficiosos como la reducción del número de palabras que tu sistema requiere procesar, reduciendo la dimensión y, por tanto, la complejidad del texto a procesar. Sin embargo, no dejan de tener inconvenientes ya que, en el caso del *stemming*, se podría reducir la precisión (aumentando los falsos positivos), y podría retornar datos relevantes del texto junto con irrelevantes que no tengan nada que ver con los objetivos de tu búsqueda. En el caso de la lematización, suele ser un proceso más preciso que el *stemming* ya que tiene en cuenta el significado de la palabra, empleando otros procesos como etiquetado PoS para mejorar la exactitud.



## Descripción del sistema desarrollado

A pesar de ello, el problema de la lematización es que las bibliotecas más empleadas no disponen de módulos para otros idiomas diferentes al inglés, así el paquete NLTK, ampliamente utilizado en PLN, emplea *WorldNet Lemmatizer* que sólo está disponible en inglés por lo que es habitual que si se quiere emplear este tipo de procesos en español, se emplea *Snowball* que realiza *stemming* que dispone de módulos en español y en otros idiomas.

En nuestro proyecto, se ha empleado *Snowball*, aunque se ha implementado como una opción voluntaria en el reconocimiento de entidades. Esto se debe a que su selección ha implicado una pérdida de reconocimiento de entidades, por lo que en el sistema desarrollado no se recomienda. En la **figura 12** se encuentra un ejemplo del reconocimiento de entidades empleando esta opción:

pais tratamient prevent ivermectin ibuprofen aspirin inutil paracetamol medicamento variant control tratamient ivermectin  
ibuprofen aspirin darl inutil paracetamol medicamento antiinflamatori cag trazabil inutil serviri seri tratamient ivermectin  
ibuprofen humild aspirin sirv inutil paracetamol medicamento desobedec dios todopodor histori juzg veng diarre consult  
traí fiebr ceftriaxon dex eritromicin naproxen paracetamol medicamento ibuprofen ivermectin cov seman ayno ibuprofen  
hras aspirin diari vitamin azitromicin ivermectin depend pes dosis tratamient envi mam bestial locatel merced gent loc busc

Figura 12. Ejemplo de reconocimiento de entidades empleando stemming

Como conclusión, se puede decir que el proceso de *stemming* es más sencillo que la lematización ya que este último requiere de un conocimiento de la lingüística para crear diccionarios que, posteriormente, permitirán al algoritmo buscar la forma apropiada de la palabra

### 4.3.3 Stopwords

En cada idioma existen palabras que contienen un menor significado que podrían ser eliminadas. Estas palabras se conocen como *stopwords* que engloban a los determinantes, verbos auxiliares y pronombres. Su eliminación implica un menor consumo de memoria y cálculos más rápidos. Sin embargo, existen autores que

consideran que estas palabras contienen un significado del dominio y que la reducción de dimensionalidad del texto (y complejidad) no compensa la pérdida de significado).

En este proyecto se ha empleado el módulo disponible en español del paquete NLTK, aunque no ha tenido un efecto apreciable en los resultados finales de este sistema.

#### 4.4 Reconocimiento de entidades (NER)

Tras la etapa de limpieza, y una vez realizada la tokenización y, si se desea, se procede al NER. Con el objetivo de aumentar la homogeneidad de los tweets y evitar una pérdida de rendimiento con una lista amplia de medicamentos, se seleccionaron sólo los tweets que involucraban a los 10 medicamentos/principios activos más frecuentes como reflejaban los datos del apartado 3.5 *Minado de mensajes en Twitter*. Los medicamentos/principios activos seleccionados fueron: Ibuprofeno, Rivotril, Aspirina, Clonazepam, Omeprazol, Hidroxicloroquina, Azitromicina, Ivermectina, Remdesivir y Dexametasona.

En el reconocimiento de entidades se emplearon las propiedades de la librería *Spacy* que proporciona un etiquetado sintáctico y funcional, añadiéndole elementos en su *pipeline*. Debido a la disponibilidad de un diccionario conteniendo los principios activos y medicamentos disponibles en España (obtenidos de fases anteriores), se procedió a la creación de una regla de entidades (*entity ruler*, en inglés) que englobasen estos términos. Mediante esta técnica incluida en la librería *Spacy* el usuario establece una serie de instrucciones para encontrar y etiquetar entidades. Tras realizar esta actividad, se pueden añadir al flujo de trabajo como un nuevo componente. No solo realiza el etiquetado, sino que el sistema desarrollado guarda las entidades en un diccionario para ser utilizado en otras fases, pudiendo elegir la etapa en la que se desea incorporar, aunque por defecto es la última etapa y así es en este proyecto.

Además de los medicamentos/principios activos también se añadió una nueva entidad basada en patrones como son la dosis. Esta entidad se programó mediante un patrón como indica la **figura 13**:

```
patterns_dosis = [{"SHAPE": "dd"}, {"ORTH": "mg"}], [{"SHAPE": "dd"}, {"IS_SPACE": True}, {"ORTH": "mg"}], [{"SHAPE": "ddd"}, {"ORTH": "mg"}], [{"SHAPE": "ddd"}, {"IS_SPACE": True}, {"ORTH": "mg"}], [{"SHAPE": "d"}, {"ORTH": "mg"}], [{"SHAPE": "d"}, {"IS_SPACE": True}, {"ORTH": "mg"}]]
```

Figura 13. Definición de la entidad "dosis"

Con respecto a las entidades relacionadas con síntomas clínicos, se ha empleado la herramienta conocida como QuickUMLS. Como se ha comentado anteriormente en el apartado 2.3 *Reconocimiento/Extracción de entidades*, QuickUMLS es una herramienta que emplea un algoritmo no supervisado para extraer entidades de textos médicos mediante la comparación de cadenas. Esta herramienta emplea MetamorphoSys que contiene todos los archivos UMLS. Está disponible en un amplio abanico de idiomas, por lo que ha sido ampliamente utilizado en el ámbito de la detección de entidades biomédicas en textos.

QuickUMLS puede ser instanciado como objeto en el que se puede elegir diferentes configuraciones:

- criterios de *overlapping*: en el caso de que varios conceptos puedan ser reconocidas como diferentes entidades, se elige la forma de actuación. Se puede seleccionar *score* que elige de entre todas las entidades aquella con mayor cercanía a la palabra (similitud, *similarity* en inglés) o mediante *length* que elige aquella entidad con mayor número de caracteres que coincidan.

- umbral : mínimo valor de similitud (*similarity* en inglés) entre las cadenas. Habitualmente se elige un valor de 0,6-0,7. Un valor más bajo, implicará el incremento de falsos positivos. Un valor más alto, implicará un menor número de entidades detectadas. En nuestro sistema se ha empleado 0.6, ya que se ha comprobado la creación de un gran número de falsos positivos cuando es inferior a éste, así como muchos conceptos eran considerados como múltiples entidades. . El hecho de emplear un parámetro de similitud medio-bajo como es 0,6 se debe a la prioridad de detectar cuantos más términos biomédicos mejor, incluso cuando esto suponga la creación de falsos positivos en un número controlable.

· nombre de la similitud: por defecto, se emplea Jaccard como medida de la similitud definida matemáticamente por la siguiente **figura 14**:

$$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2}$$

*Figura 14. Expresión matemática de la similitud mediante Jaccard*

Una similitud de Jaccard se encuentra en el rango de 0 a 1. Si los dos documentos o palabras tienen un valor de 1, significa que son iguales. En cambio, si es de valor 0, indica que no son nada iguales.

· ventana: número máximo de tokens a considerar para el *matching*, en este proyecto se empleó el número por defecto que fue 5.

Las entidades reconocidas por QuickUMLS fueron etiquetadas como UMLS. Debido a la limpieza previa del texto antes de realizar el NER, algunas de los métodos no fueron empleados, como la reducción del texto a minúsculas.

Tras esta etapa los tweets fueron extraídos en base a tres características en el que se indicaban la entidad dosis reconocida (una o varias de ellas), entidad medicamento y entidad UMLS, pudiendo detectarse múltiples ocurrencias de cada una de ellas. Debido al hecho que nuestro objetivo era detectar efectos adversos de medicamentos, se extrajeron todos aquellos mensajes que hubiesen detectado algún síntoma clínico.

#### 4.5 Clasificación de efectos adversos

Tras el paso anterior, se etiquetaron los datos mediante revisión automática comparando el medicamento/principio activo con sus potenciales efectos adversos descritos en la ficha técnica. En el caso que hubiese duda, se hizo una revisión manual.

## Descripción del sistema desarrollado

Con el objetivo de evaluar la capacidad que tiene el sistema de predecir los efectos adversos, se emplearon diferentes algoritmos que se evalúan en el siguiente subapartado, aunque finalmente se decidió por emplear regresión logística por obtener los mejores resultados en cuanto a precisión.

### 4.5.1 Extracción de características (*feature extraction*)

Con el objetivo de analizar los tweets en los que se había considerado que aparecían efectos adversos, se procesaron los textos mediante diferentes métodos de *word embeddings*. Mediante esta técnica las palabras son convertidas en vectores que, posteriormente, son procesados mediante el algoritmo seleccionado. Los métodos empleados han sido:

- CountVectorizer: método que proporciona una forma fácil y sencilla de *tokenizar* los textos y construir un vocabulario de palabras conocidas, pero también codificar nuevos documentos utilizando el vocabulario creado. Además, tiene la capacidad de ignorar las puntuaciones y convertir las mayúsculas en minúsculas cuando aparezcan en el texto. Los parámetros más importantes incluidos en este método son:

- stop\_words: debido a que esta técnica sólo cuenta las ocurrencias de cada palabra, existen palabras muy comunes como 'de' o 'y' que serán características muy importantes del texto pero que aporten poco o nada de significado del texto. El modelo podría mejorarse si no se tienen en cuenta estas palabras.

- ngram: permite mejorar la potencia predictiva en algunos casos ya que clasifica las palabras no como tokens únicos, sino que podrían ser formadas por 2- o 3-gramas, por ejemplo, en un texto con "tengo infección abdominal", si seleccionamos 2-gramas, tendría "tengo infección" y "infección abdominal".

- min\_df, max\_df: frecuencias mínimas y máximas del de palabras/n-gramas para ser utilizadas como características.

· TF-IDF: método alternativo que consiste en medir la frecuencia del término (TF) y la frecuencia inversa del término (IDF), mediante la fórmula siguiente:

$TF(t) = (N.^{\circ} \text{ de veces que el término } t \text{ en un documento}) / (N.^{\circ} \text{ total de términos en el documento})$

$IDF(t) = \log(N.^{\circ} \text{ total de documentos} / N.^{\circ} \text{ de documentos con el término } t \text{ en ellos}).$

El valor IDF mide lo importante que es un término, es decir, destaca aquellas palabras que aparecen en muy pocos documentos en todo el *corpus*. Finalmente, para obtener el valor de TF-IDF se realiza la fórmula de la **figura 15**:

$$tfidf(t, d, D) = tf(t, d) * idf(d, D)$$

Figura 15. Fórmula TF-IDF

Una puntuación alta de TF-IDF se obtiene mediante un término que tiene una frecuencia alta en un documento y una frecuencia de documento baja en el *corpus*. Para una palabra que aparece en casi todos los documentos, el valor de IDF se acerca a 0, lo que hace que TF-IDF también se acerque a 0. El valor de TF-IDF es alto cuando los valores de IDF y TF son altos, es decir, la palabra es rara en todo el documento pero frecuente en un documento.

· *Word2Vec*: es un método que crea vectores de palabras que son representaciones numéricas de características de palabras. La eficacia de *Word2Vec* viene de su capacidad de agrupar vectores de palabras similares. *Word2Vec* emplea dos arquitecturas diferentes: *CBOW* (bolsa de palabras continua) y *skip-gram*. La arquitectura CBOW es muy similar a una red neuronal prealimentada (feed forward). Mediante este modelo se intenta predecir una palabra objetivo de una lista de palabras

del contexto. El modelo *skip-gram* es una red neuronal simple en el que dado un *corpus*, el modelo analiza las palabras de cada sentencia y trata de usar cada palabra para predecir que palabras serán vecinas. Al contrario que TF-IDF, en *Word2Vec* emplea aprendizaje no supervisado. Aunque se han comparado otros métodos como *FastText* con *Word2Vec*, el empleo de *FastText* fue descartado debido a las características de este modelo y el proyecto. *FastText* posiciona las palabras de un documento en un espacio vectorial basado en los n-gramas de las letras que las conforman y en *skip-grams* de palabras para detectar el contexto en que se utilizan. Los resultados entre las dos han sido similares aunque el hecho de que el entrenamiento en *FastText* requiere altos requisitos de memoria cuando se trata de *corpus* de tamaño cada vez más grandes, ha hecho que se ha ya empleado *Word2Vec*.

### 4.5.2 Algoritmos considerados

- Regresión logística: Técnica de clasificación que emplea una función logística para modelar la variable dependiente. La variable dependiente es binaria, por ejemplo, es o no es un efecto adverso. Matemáticamente, el planteamiento del problema se puede formular así, donde  $x$  representan los atributos del problema (en este caso, una cadena de texto), de forma que sus valores podrían corresponder a cuantas veces aparece una palabra en un texto mientras que la predicción  $y$  representa la probabilidad de que aparezca un efecto adverso como se observa en la **figura 16**:

$$y = \sigma(z) = \sigma(WX) = \sigma\left(\sum (w_i x_i)\right) = \sigma\left(\sum (w_0 x_0 + w_1 x_1 + \dots + w_n x_n)\right)$$

Figura 16. Expresión matemática de la regresión logística

La función empleada llamada función logística o sigmoidea, se obtiene mediante la fórmula, como aparece en la **figura 17**:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Figura 17. Función logística

Las principales ventajas de la regresión logística se basan en que es fácil de implementar, interpretar y fácil de entrenar, así como la rapidez en la clasificación de registros desconocidos mientras que los inconvenientes que tiene su uso se basa en que si el número de observaciones es menor que el número de características podría

conducir al *overfitting*, así como que asume la linealidad entre la variable dependiente y la variable independiente.

· Support Vector Machine (SVM): Es un modelo lineal para problemas de clasificación y de regresión. Puede resolver problemas lineales y no lineales, ya que el algoritmo que usa se basa en la creación de una línea o un hiperplano que separa los datos en clases. Cuando los datos se pueden separar mediante una línea recta se habla de SVM lineal mientras que cuando no se puede realizar linealmente se recurren a las funciones kernel que transforman espacios no lineales en lineales, pudiendo los datos clasificarse. Las ventajas que presenta SVM en tareas de clasificación son la efectividad en espacios altamente dimensionales, incluso donde el número de dimensiones es mayor que el número de muestras, junto con que se pueden emplear diferentes funciones *kernel* que pueden ser diseñadas específicamente para el problema a tratar. Por otra parte, el rendimiento cuando se emplean es malo debido al empleo de su validación cruzada, así como la posibilidad de *overfitting* en caso de que el número de características sea superior al número de muestras. Otra importante desventaja en el proyecto que nos ocupa es que para grandes *datasets*, el tiempo de entrenamiento puede ser muy elevado, e incluso elegir la función kernel correcta podría ser computacionalmente intensa.

· Naive Bayes Multinomial: es un clasificador relativamente sencillo de implementar. Uno de los aspectos desfavorables de estos clasificadores es que asume que los términos que aparecen en un documento son todos independientes entre sí lo que puede no ser



## Descripción del sistema desarrollado

totalmente cierto debido a la estructura del lenguaje. La forma multinomial considera el número de apariciones del término para evaluar la contribución de la probabilidad condicional dada la clase con lo que el modelado de cada documento se ajusta mejor a la clase a la que pertenece. Las ventajas que presenta este método es que se pueden integrar múltiples variables en los cálculos para clasificar datos, es fácil de integrar con características conocidas en un conjunto de datos. Por otra parte, presenta como desventajas que los predictores se consideran independientes entre sí, que puede no ser cierto y generar un modelo que no se ajuste correctamente. El esfuerzo computacional aumenta a medida que aumentan las características de la muestra.

- k-NN: el algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. De esta forma, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecer. Este grupo será, por tanto, el de mayor frecuencia con menores distancias. Se denomina también un algoritmo de aprendizaje vago, ya que en la fase de entrenamiento sólo almacena el *dataset* y cuando llegan nuevos datos, clasifica los datos en la categoría que es más similar a los nuevos datos. Este algoritmo presenta alta precisión, pero es computacionalmente costoso ya que almacena todos los datos y utiliza una gran cantidad de memoria.

- Bosque aleatorio (CRF): conjunto de árboles de decisión combinados en el que ninguno de ellos ve todos los datos de entrenamiento. Esto hace que cada árbol se entrene con distintas muestras de datos para un mismo problema. De esta forma, al combinar sus resultados, unos errores se compensan con otros y tenemos una predicción que generaliza mejor. Las principales ventajas de este algoritmo es que es muy fácil de usar con buenos resultados de predicción ya que son fáciles de entrenar, sin embargo, existe una alta tendencia al *overfitting* y pueden ser ineficientes cuando se utilizan un número alto de árboles.

Junto con todos estos algoritmos, se podría necesitar herramientas que permitan evitar los errores que se producen debido a un conjunto de datos desequilibrados, es decir, en este proyecto se puede llegar a detectar un número elevado de efectos adversos o viceversa, que podrían llegar a tener una proporción del 70-80% que podrían tener

consecuencias en la exactitud de la predicción y ,por lo tanto, en el resto de las métricas que evalúen al sistema desarrollado. En aprendizaje automático se distinguen entre

parámetros e hiperparámetros. En el caso de los primeros, son las variables utilizadas por el algoritmo para predecir los resultados en base a los datos de entrada. Estas variables forman parte del entrenamiento del modelo, como por ejemplo, el coeficiente de variables independiente que se emplea en Regresión Logística. En cambio, los hiperparámetros son variables que el usuario especifica mientras construye el modelo de aprendizaje automático. Así, los hiperparámetros se emplean para evaluar los parámetros óptimos del modelo. Sus valores se deciden por el usuario que construye el modelo.

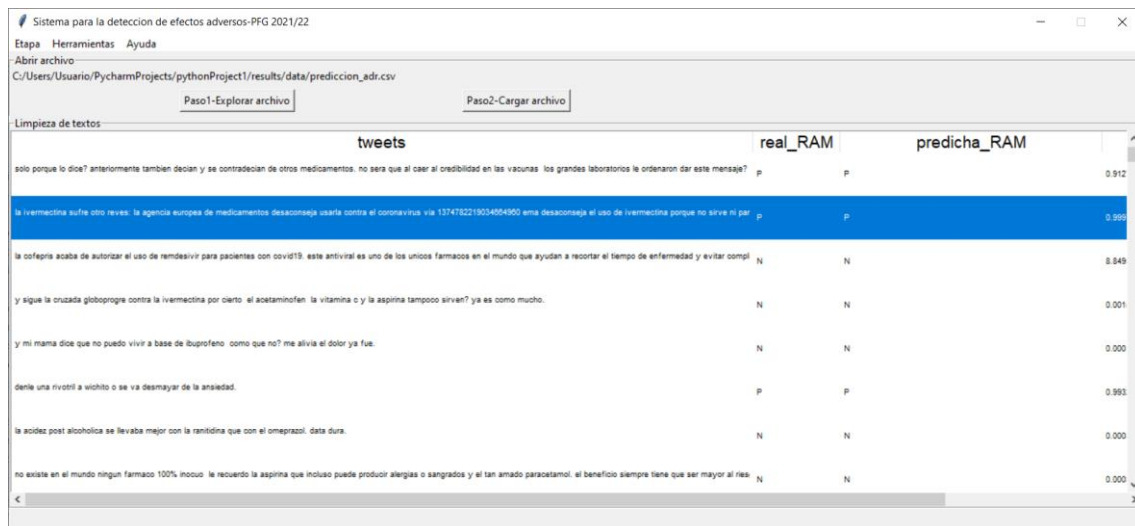
#### 4.6. Desarrollo de una aplicación basada en el sistema

El sistema se desarrolla para ser utilizada mediante aplicación de escritorio con el objetivo de que el usuario pudiese emplear una interfaz que sea fácil de instalar, utilizar y muy intuitiva. Para ello, se diseñaron diferentes etapas en las que el usuario cargaba un texto y obtenían unos resultados por pantalla y en forma de archivos. Finalmente, en la última etapa obtiene y puede visualizar el clasificador con el texto del mensaje, la presencia o ausencia de efecto adversos contenida en el texto, la presencia o ausencia de efecto adverso predicha y la probabilidad de la predicción. Esta visualización puede almacenarse como un archivo disponible en formato .csv.

Debido al empleo de una aplicación externa como QuickUMLS, la aplicación requiere de ella para una de las etapas, concretamente el reconocimiento de entidades. El resultado dependerá del empleo de los diccionarios de términos médicos (incluido en QuickUMLS), si bien es estándar la utilización de los términos UMLS para la detección de términos biomédicos en textos.

Un ejemplo del resultado final se observa en la **figura 18**:

## Descripción del sistema desarrollado



Sistema para la detección de efectos adversos-PFG 2021/22

Etapas Herramientas Ayuda

Abrir archivo

C:/Users/Usuario/PycharmProjects/pythonProject1/results/data/prediccion\_adr.csv

Paso1-Explorar archivo Paso2-Cargar archivo

Limpieza de textos

tweets	real_RAM	predicha_RAM	
solo porque lo dice? anteriormente tambien decian y se contradiaban de otros medicamentos, no sera que al caer al credibilidad en las vacunas los grandes laboratorios le ordenaron dar este mensaje?	P	P	0.912
la ivermectina sufre otro revés: la agencia europea de medicamentos desaconseja usarla contra el coronavirus via 1374782213034954900 ema desaconseja el uso de ivermectina porque no sirve ni par	P	P	0.993
la cortepra acaba de autorizar el uso de remdesivir para pacientes con covid19. este antiviral es uno de los unicos farmacos en el mundo que ayudan a recortar el tiempo de enfermedad y evitar compl	N	N	8.849
y sigue la cruzada globoprogre contra la ivermectina por cierto el acetaminofen la vitamina c y la aspirina tampoco sirven? ya es como mucho.	N	N	0.001
y mi mama dice que no puedo vivir a base de ibuprofeno como que no? me alivia el dolor ya fue.	N	N	0.000
denle una rivotril a wicito o se va desmayar de la ansiedad.	P	P	0.993
la adicex post alcohólica se levaba mejor con la ranitidina que con el omeprazol. data dura.	N	N	0.000
no existe en el mundo ningún farmaco 100% inocuo. le recuerdo la aspirina que incluso puede producir alergias o sangrados y el tan amado paracetamol. el beneficio siempre tiene que ser mayor al ries	N	N	0.000

Figura 18. Pantalla con los resultados finales del sistema



## 5. Metodología de desarrollo y diseño

### 5.1 Metodología

El desarrollo del sistema ha sido condicionado por la finalidad investigadora de este proyecto, intentando obtener un ejemplo reflejo de los objetivos propuestos. Es por ello que , entre todas las metodologías de desarrollo de software existentes, la más adaptada a este proyecto es la metodología ágil, permitiendo una alta flexibilidad en el desarrollo del propio software. Por un lado, permite adaptar el software a las necesidades que van surgiendo y , por otro lado, en cada ciclo de desarrollo se van agregando nuevas funcionalidades a la aplicación final, permitiendo añadir pequeñas funcionalidades en lugar de grandes cambios.

Con vistas a obtener un prototipo rápido se ha preferido desarrollar una aplicación de escritorio, fácilmente instalable y con capacidad para trabajar sobre los recursos del usuario tras su instalación. A su vez, se ha preferido el lenguaje Python sobre otros lenguajes de programación tal y como se ha señalado en el apartado 3.2 *Herramientas utilizadas*. Si bien no deja de tener inconvenientes que se señalan en el apartado 7. *Conclusiones*, permite la creación de *scripts* fáciles de implementar con la ayuda de las librerías disponibles para este lenguaje relacionados con el campo del Aprendizaje Automático y PLN. A pesar de ello, se ha intentado mantener una alta cohesión y bajo acoplamiento en el diseño del programa, reduciendo la interacción entre diferentes módulos y permitiendo que cada módulo codificado estuviese compuesto por elemento altamente relacionados.

### 5.2 Planificación

En la planificación del trabajo no se ha tenido en cuenta el tiempo dedicado a la obtención del minado de mensajes en redes sociales ni tampoco la implementación diccionarios externos o la elaboración de diccionarios internos.

## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

El hecho de no incluirlos en la planificación se debe dos motivos:

- No es objetivo de este estudio evaluar distintas alternativas de diccionarios en el empleo de detección de RAM en redes sociales ni tampoco un sistema de minado de mensajes en redes sociales.

- El propio objeto de este proyecto es la obtención de un ejemplo sencillo de implementación, con diccionarios ya creados y con datos disponibles por parte del usuario.

La planificación del trabajo se puede observar en la **figura 19** mediante el diagrama de Gantt:

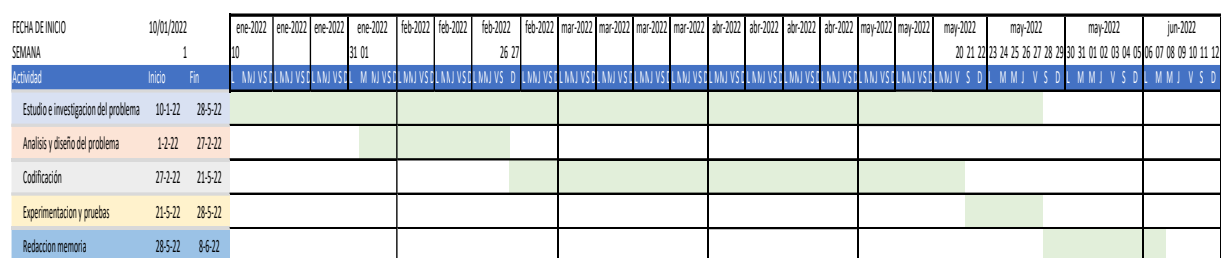


Figura 19. Diagrama de Gantt del proyecto

Como se ha comentado anteriormente, la metodología ágil permite la incorporación de nuevas funcionalidades a medida que se avanza en el desarrollo del proyecto. En este caso, se puede observar en el diagrama de Gantt como el estudio y la investigación del problema ha continuado durante todo el tiempo dedicado, solapándose con otras etapas. Sin tener en cuenta el minado de textos en redes sociales ni la obtención de diccionarios, el trabajo ha llevado prácticamente 6 meses.

La distribución de los esfuerzos se indica en la **tabla 6**:

Estudio e investigación (50%)	Análisis y diseño (10%)	Codificación (30%)	Experimentación y pruebas (5%)	Redacción de memoria (5%)	Total
1 persona * 3 meses = 3 pm	1 persona * 0,75 mes = 0,75 pm	1 persona * 1,75 meses = 1,75 pm	1 persona * 0,25 mes = 0,25 pm	1 persona * 0,25 mes = 0,25 pm	6 pm
3 meses	0,75 mes	1,75 meses	0,25 meses	0,25 meses	6 meses

*Tabla 6. Distribución del esfuerzo*

### 5.3 Requisitos del sistema

#### 5.3.1 Requisitos funcionales

Los requisitos funcionales del sistema incluyen:

- El sistema aceptará como datos de entrada archivos en formato .csv obtenidos del minado de la red social Twitter a través de la interfaz gráfica de usuario (GUI) de la aplicación.

- El sistema mostrará el resultado de cada etapa: limpieza, preprocesamiento, NER y clasificación. A su vez, en cada etapa se podrá obtener un archivo independiente para proseguir con el resto de etapas de forma continua o dilatada en el tiempo. La etapa de reconocimiento de entidades también tendrá disponible el acceso a un archivo HTML que muestra gráficamente todas aquellas entidades reconocidas por el sistema, además de la clase de entidad. El resultado final de la última etapa de clasificación serán unos gráficos que representan el número total de RAM presentes en el documento, la

matriz de confusión y la curva ROC (Receiver Operating Characteristic o Característica Operativa del Receptor) y un archivo en formato .csv que muestre el resultado de la etapa de clasificación.

- La GUI validará que en cualquier momento se cumplan los requisitos de entrada de archivos, además debido al empleo de un sistema de extracción de entidades como QuickUMLS, el sistema comprobará y exigirá la entrada del directorio donde se encuentra este sistema durante la etapa de NER. En caso de que no se cumplan, enviará un mensaje de advertencia al usuario.

- Debido a las limitaciones en cuanto a rendimiento de la capacidad de reconocer entidades en el caso de que los archivos de texto superen los 900000 caracteres, el sistema propondrá al usuario dividir el texto en estudio en partes iguales. Finalmente, tras realizar el NER, el usuario podrá unir los resultados obtenidos para hacer un archivo final y someterlo a la etapa de clasificación.

- En cuanto a los requisitos de hardware o sistema operativo, la aplicación debe poder utilizarse en un sistema Windows. Es aconsejable disponer de una memoria RAM superior a 8 gb por motivos de rendimiento. Los archivos resultantes de la aplicación no requieren ningún software especial , ya que son textos planos o en formato .csv o gráficos en formato .png o página web en formato HTML, éste último se puede abrir según el navegador por defecto del usuario.

#### 5.3.2 Requisitos no funcionales

- El tiempo de aprendizaje de la aplicación debe ser menor a una hora.
- El sistema cuenta con un manual de usuario y un video explicativo.
- La aplicación genera mensajes de error y advertencias para orientar al usuario en el uso y navegación de la aplicación.

#### 5.4 Casos de uso



## Metodología de desarrollo y diseño

Los casos de uso incluidos en esta memoria se refieren al empleo de la aplicación por parte del usuario. Se describe el principal caso de uso detallado mediante un flujo básico y los flujos alternativos, de acuerdo a la **tabla 7**:

Caso de uso	Obtener un clasificador de efectos adversos en textos de la red social Twitter.
Actor	usuario
Flujo básico	El usuario posee un texto con mensajes de la red social Twitter y entra en el sistema para detectar RAM
Flujo alternativo 1	El usuario posee un archivo obtenido de etapas previas a la clasificación y desea clasificarlo
Flujo alternativo 2	El usuario presenta un archivo con mayor número de caracteres permitido

*Tabla 7. Caso de uso*

A continuación, se describe el flujo básico en la **tabla 8**:

Flujo básico	Obtener un clasificador de efectos adversos en textos de la red social Twitter.
Descripción	El usuario dispone de un archivo resultante del minado de tweets. Se trata del principal escenario de éxito
1	El usuario ejecuta la aplicación
2	El usuario selecciona la etapa inicial de Limpieza
3	El usuario selecciona el archivo inicial y procede a la limpieza del texto
4	El sistema muestra el resultado de la limpieza, informa de la creación de un nuevo archivo con los mensajes limpios para su posterior uso en el reconocimiento de entidades.
5	El usuario selecciona reconocimiento de entidades
6	El usuario selecciona el directorio donde se encuentra el extractor de entidades QuickUMLS y el archivo limpio obtenida en la etapa de limpieza

## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

7	El sistema muestra el resultado de las entidades reconocidas. El usuario puede obtener el archivo con las entidades reconocidas y su clase, así como una página HTML para visualizarlo gráficamente.
8	El usuario selecciona la clasificación de los textos
9	El usuario selecciona el texto de la etapa de reconocimiento y el sistema procede a la clasificación de textos. El sistema crea un archivo final con el mensaje de texto y el resultado de la clasificación, así como gráficos resultantes de la clasificación en formato .png.
10	El usuario puede repetir el proceso con más textos volviendo a cada una de las etapas en función de la disponibilidad y formato de sus textos.

*Tabla 8. Flujo básico del caso de uso*

A continuación, se describe el flujo alternativo 1 en la **tabla 9**:

Flujo alternativo 1	Carga de archivo
2	El usuario dispone de un archivo resultante de una etapa posterior de la ejecución del programa y no necesita limpiarlo.
2A1	El usuario selecciona la etapa de reconocimiento de entidades para procesarlo.
2B1	El usuario selecciona la etapa de clasificación para procesarlo.

*Tabla 9. Flujo alternativo 1*

A continuación, se describe el flujo alternativo 2 en la **tabla 10**:

Flujo alternativo 2	Carga de archivo
6	El usuario presenta un archivo con mayor número de caracteres permitido por el reconocimiento de entidades. El sistema informa de ello.
6A1	El usuario selecciona un archivo de menor número de caracteres

## Metodología de desarrollo y diseño

6B1	El usuario selecciona Herramientas para dividir el texto
6B2	El usuario selecciona el archivo a dividir
6B3	El sistema muestra el resultado de la división e informa de la identificación del texto generado
6B4	El usuario vuelve a la etapa de reconocimiento de entidades

*Tabla 10. Flujo alternativo 2*



## 6. Evaluación del sistema

### 6.1 Métricas para la extracción de información

Con el objetivo de evaluar los resultados obtenidos, es muy importante seleccionar métricas de evaluación claras, reproducibles y fácilmente comprensibles. Antes de presentar las métricas seleccionadas en este proyecto, se debe definir una estructura ampliamente utilizada en el ámbito de estudio como es la matriz de confusión (también llamada tabla de contingencia). Esta matriz se encuentra dividida en cuatro categorías:

- Verdaderos Positivos (VP): ejemplos correctamente etiquetados como positivos
- Falsos Positivos (FP): ejemplos negativos incorrectamente etiquetados como positivos.
- Verdaderos Negativos (VN): ejemplos negativos correctamente etiquetados como negativos.
- Falsos Negativos (FN): ejemplos positivos incorrectamente etiquetados como negativos.

A continuación, se presenta la matriz de confusión en la **figura 20** :

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

*Figura 20. Matriz de confusión*

La matriz de confusión presenta dos columnas que indican el número de predicciones de cada clase y dos filas que representan las observaciones de cada clase. De tal forma, que cada elemento vendrá determinado por una columna que se predice y una fila que se observa.

A continuación, se definen las métricas empleando los valores de los elementos de la matriz de confusión:

·Sensibilidad (o exhaustividad/*recall*): Mide la fracción de ejemplos positivos que son correctamente etiquetados.

$$\text{Sensibilidad} = \text{recall} = \frac{VP}{VP+FN}$$

·Precisión: Mide la proporción de ejemplos predichos que son realmente positivos:

$$\text{Precisión} = \frac{VP}{VP+FP}$$

·valor F: es un parámetro definido por las dos medidas anteriores, ya que es la media armónica entre los valores de *recall* y de precisión, donde el parámetro  $\beta$  indica el peso relativo de precisión respecto al valor de *recall*. En caso de que  $\beta=1$ , se está dando la misma ponderación a precisión que a *recall*, obteniendo el valor del parámetro F1. La fórmula, donde P representa a precisión y R a *recall*, es la siguiente:

$$F(\beta) = \frac{(1+\beta^2)PR}{\beta^2P+R}$$

Un sistema ideal de extracción de información es aquél en el que no hubiera ni falsos positivos o falsos negativos. Sin embargo, esto es difícilmente conseguible en sistemas reales por lo que se observa que la precisión y el *recall* son parámetros antagónicos, cuando uno aumenta, el otro se reduce y viceversa.

La selección de estos parámetros se ha basado en la literatura disponible ya que la precisión, el *recall* y f1 son ampliamente utilizados para comparar los sistemas. Adicionalmente, con el objetivo de evaluar diferentes algoritmos planteados para el proyecto, se ha empleado el término exactitud definida como :

$$\text{Exactitud} = (VP + VN) / (VP+FP+VN+FN)$$

### 6.2 Resultados

#### 6.2.1 *Corpus* de prueba

El *corpus* de prueba consistió en la extracción de tweets que mencionaban cualquier principio activo o medicamento comercializado en España en el momento del minado. El número total de tweets correspondió a 1016753, que fueron filtrados a los diez medicamentos/principios activos más frecuentes, quedando finalmente en un *corpus* conteniendo 216105 tweets (ejemplo de archivo: “./data/output.csv”)

#### 6.2.2 Limpieza

El texto conteniendo nuestras referencias a los medicamentos y principios activos contenidos se somete a un proceso de limpieza, de forma que en nuestro ejemplo, el archivo presenta utilizado anteriormente presentaba 5437940 caracteres y acaba con 4957280 caracteres. Finalmente, se realiza un análisis de impurezas por si ha quedado algún carácter que deberíamos haber eliminado o no ha sido reconocido, en este caso, 0,009%. Finalmente, genera dos archivos e indica el directorio donde se encuentran, el archivo de texto se utilizará para las fases posteriores. El resultado final del proceso de limpieza aparece reflejado en la **figura 21**:

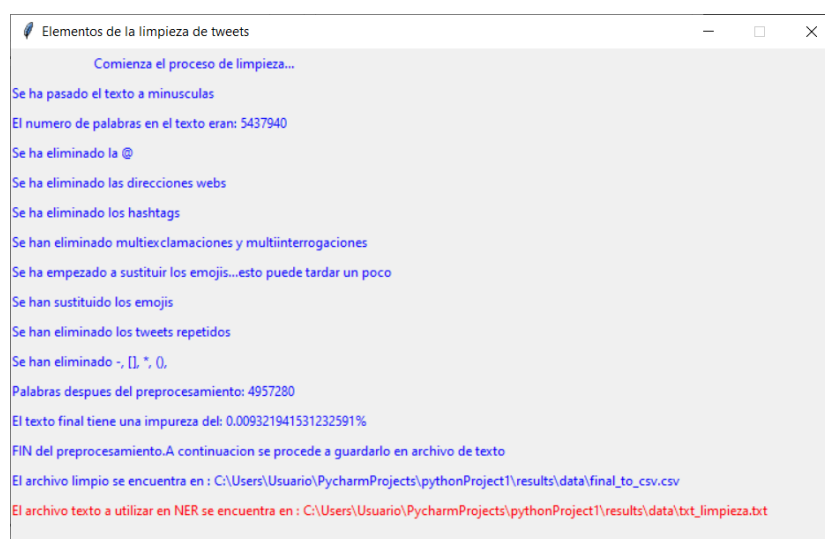


Figura 21. Resultados de la limpieza del corpus

6.2.2 Reconocimiento de entidades

Una vez obtenido el fichero de limpieza de pasos anteriores se procede a la aplicación de los diccionarios y la herramienta QuickUMLS para el reconocimiento de entidades. Los resultados se guardan en un archivo HTML llamado `NER_displacy.html` en la carpeta `data`, que se muestra en la **figura 22**:

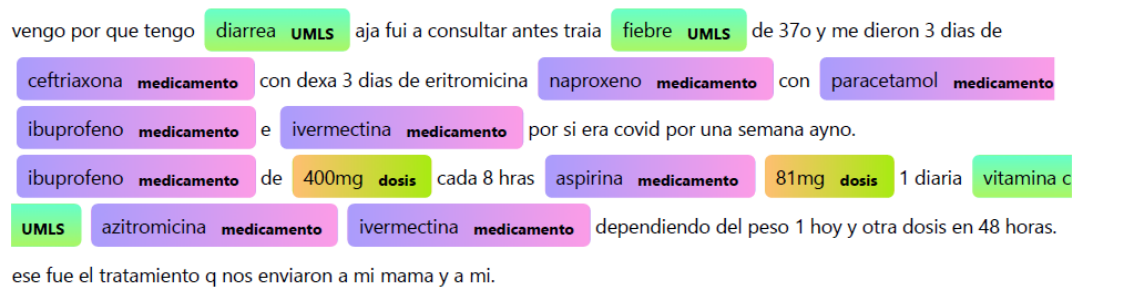


Figura 22. Extracto de un texto tras NER

A la vez que se genera un archivo conteniendo cuatro columnas que representan el texto original del tweet( llamada `text`), la entidad “`dosis`” (si no hay indicación se indica como `no`), la entidad “`medicamento`” y la entidad “`UMLS`”, como indica la **figura 23**:

texto	dosis	medicamento	UMLS
a un amigo le dieron diagnostico de inicio de infeccion de amigdalas y el tratamiento era ivermectina azitromicina e ibuprofeno. de verdad si no saben no inventen payasadas. ponganse a leer.	no	'azitromicina', 'ibuprofeno', 'ivermectina'	'infeccion'

Figura 23. Resultados del reconocimienro de entidades

Existe una opción de *stemming* en el reconocimiento de entidades, pero en el proyecto no dio buenos resultados, reduciendo el número de entidades de término médicos incluidas en la etapa final.



### 6.2.3 Clasificación de textos

Finalmente con el archivo obtenido en la fase anterior se procede a la clasificación de textos. El número de tweets que han reconocido las entidades biomédicos y medicamentos/principios activos han sido 13098, esto corresponde al 6,06% de los tweets originales filtrados por los 10 medicamentos/principios activos más numerosos. Se realiza una detección automática en función de los fármacos diana y los términos. Cuando se clasifican los efectos adversos se obtiene la siguiente clasificación de los textos en función de si contiene o no efectos adversos, como aparece reflejada en la **figura 24**:

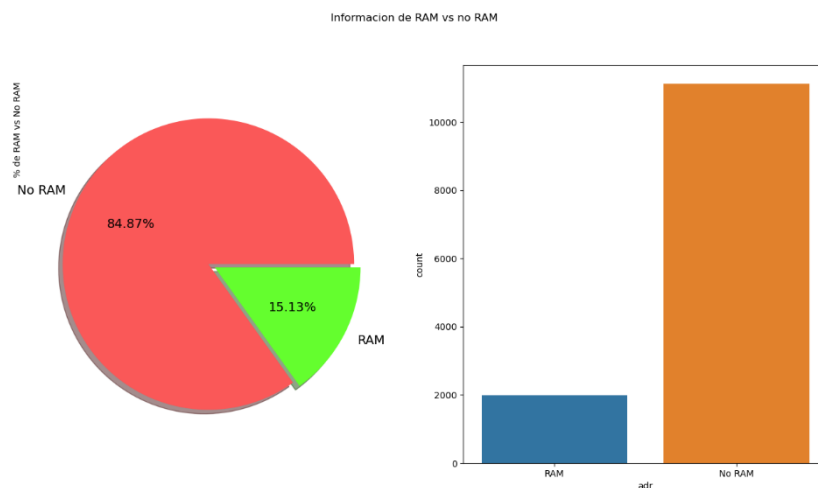


Figura 24. Número de efectos adversos detectados en los tweets.

Posteriormente, se emplean los algoritmos indicados en el apartado 4.5.2 que son:

- Naive Bayes Multinomial
- Regresión logística
- k-NN
- Linear SVM
- CRF

En un primer momento de la experimentación, se optó por experimentar con todos ellos para seleccionar aquel algoritmo con mejores prestaciones para posteriormente elegir uno de ellos. Cuando se emplean todos los algoritmos para generar un conjunto de entrenamiento y, posteriormente, al conjunto test, se obtiene los siguientes valores de exactitud y el tiempo necesario para entrenar a un conjunto indicados en la **tabla 11**:

<b>Algoritmo</b>	<b>Exactitud sobre el conjunto prueba</b>	<b>Tiempo para entrenamiento</b>
<b>Regresión Logística</b>	<b>0.95598</b>	1,12 s
Linear SVM	<b>0.958015</b>	7,86 s
k-NN	0.907125	<b>0,01 s</b>
Naive Bayes Multinomial	0.922901	<b>0,02 s</b>
CRF	0.940712	1,07 s

*Tabla 11. Resultados de la precisión de los algoritmos*

Como se puede observar en la tabla, a pesar de los buenos resultados que se obtienen con SVM lineal, tiene un gran problema de rendimiento como se indicó en la presentación de las características de los algoritmos. Naive Bayes y k-NN obtuvieron un gran rendimiento pero se pierde exactitud.

Midiendo la exactitud en el conjunto de entrenamiento, el tiempo de entrenamiento y la exactitud en el conjunto de test, se obtienen los resultados de la **tabla 12**:

<b>Algoritmo</b>	<b>Exactitud en el conjunto de entrenamiento</b>	<b>Tiempo de entrenamiento</b>	<b>Exactitud en el conjunto test</b>
Regresión Logística	<b>0.999346</b>	1.185284 s	<b>0.955980</b>
Linear SVM	0.989747	7.905696 s	<b>0.958015</b>

k-NN	0.925502	<b>0.005614 s</b>	0.907125
Naive Bayes Multinomial	0.996401	0.014771 s	0.877354
CRF	0.997273	1.093027 s	0.934860

Tabla 12. Resultados de los algoritmos

A pesar de sus buenos resultados, el mal rendimiento de SVM la descarta ya que el tiempo empleado es excesivo con respecto al resto de algoritmos con niveles similares de exactitud.

Una vez realizados estos análisis, parece probable emplear Regresión Logística en términos de rendimiento y exactitud, aunque nos queda establecer la posibilidad de *overfitting* en los algoritmos implicados. El *overfitting* se podría deducir de aquellos casos en los que existe una diferencia ostensible entre la exactitud del conjunto de entrenamiento o del test. Este error consiste en que el algoritmo empleado tiene en cuenta el ruido aleatorio más que el propio patrón. La mejor forma de evitarlo es emplear la validación cruzada, así se dividió el conjunto de datos en tres partes: 2/3 se empleó para el entrenamiento y 1/3 para el conjunto test y se hará el proceso de testing 3 veces. Los resultados de la validación cruzada se muestran en la **tabla 13**:

Algoritmo	Validación cruzada
Regresión Logística	<b>0.953098</b>
CRF	0.921357
k-NN	0.902705
Naive Bayes Multinomial	0.865620

Tabla 13. Resultados de la validación cruzada

Los resultados de esta validación cruzada corroboran el empleo de Regresión Logística como algoritmo elegido para la clasificación de los textos. Una vez seleccionado el algoritmo se ha empleado otro método que permite corregir los errores que se podrían producir por la presencia de un claro desequilibrio entre la presencia de efectos adversos y su ausencia, en nuestro caso, hay un 15,13% de efectos adversos detectados mientras que 84,87% de ausencia de efectos adversos. Para ello se emplea una herramienta disponible llamada GridSearchCV. Mediante esta herramienta se emplean todos los hiperparámetros especificados y evalúa cada una de las combinaciones de los hiperparámetros hasta que selecciona la que obtiene una mejor evaluación. Además, se realiza una validación cruzada que anteriormente se ha observado su utilidad.

En el caso que nos ocupa, se seleccionó como argumentos en el método GridSearchCV, el empleo de Regresión Logística como estimador y una validación, como en pasos anteriores, igual a 3. A su vez, dentro de los parámetros, se indica el valor desequilibrado de los datos con respecto a la presencia o ausencia de efectos adversos. En este caso, los valores de indicaron que la ausencia de efectos adversos existía en el 80% de los y su presencia en en el 20% restante.

Cuando se emplea los resultados de la clasificación muestra que el modelo presenta una precisión del 91%, *recall* del 80% y  $f1:85,21$ . Finalmente, para obtener una visualización de la relación precisión y *recall*, se emplea la curva ROC que permite la representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. En el proyecto el valor de ROC fue de 0.9783476623805226.

Empleando regresión logística, la matriz de confusión que se obtuvo fue la indicada en la **figura 25**:

## Evaluación del sistema

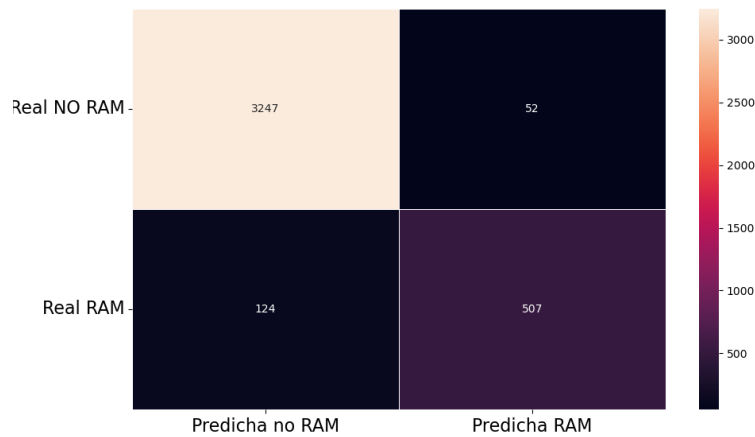


Figura 25. Matriz de confusión obtenida en la regresión logística

A continuación, se presenta la curva ROC en la **figura 26**:

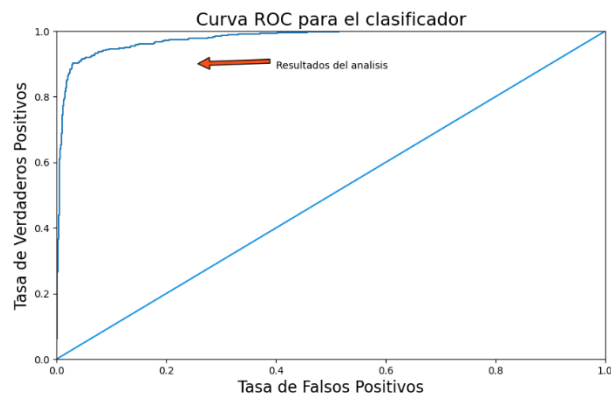


Figura 26. Curva ROC resultado de la Regresión Logística

La Regresión Logística se aplicó utilizando vectores obtenidos mediante TF-IDF, cuando se planteó el empleo de Word2Vec combinado con TF-IDF los resultados obtenidos fueron peores que cuando se empleó TF-IDF. En el caso de Word2Vec se empleó un tamaño de vector de 50, con el parámetro de ignorar palabras que tengan una frecuencia menor de 5 y con una distancia máxima entre la actual palabra y la predicha de 5. Una posible explicación es que Word2Vec podría haber hecho *overfitting* con

respecto a TF-IDF, creando falsos modelos e innecesarios. En la **tabla 14** se comparan los datos de empleando TF-IDF o TF-IDF combinada con Word2Vec:

	precisión	recall	F1
TF-IDF	91	80	85
TF-IDF+Word2Vec	76	82	78

*Tabla 14. Comparativa de resultados TF-IDF vs TF-IDF+Word2Vec*

Por último, los resultados se almacenan en un archivo .csv con el texto original, el valor de RAM real, el valor de RAM predicha y la probabilidad de la predicción.

## 7. Conclusiones

### 7.1 Conclusiones

En este proyecto se ha pretendido establecer un sistema de detección de efectos adversos de medicamentos en el que a partir de mensajes de usuarios de la red social Twitter pudiera detectarse la relación entre medicamentos y efectos adversos.

Como resultado de este trabajo, el sistema desarrollado ha tenido unos buenos resultados. El valor de F1 fue del 0,85, con una precisión y recall del 0,91 y 0,80, respectivamente. Estos valores se asemejan a otros obtenidos y consultados en la bibliografía. Aunque la detección de términos médicos se ha empleado un algoritmo no supervisado mediante el sistema QuickUMLS, la clasificación se ha realizado mediante un algoritmo supervisado. Aunque se han comparado varios algoritmos, el mejor resultado ha estado presente con Regresión Logística, con resultados óptimos en cuanto a métricas y rendimiento.

Estos buenos resultados se basan en varios aspectos. Por un lado, la limpieza de textos evita los múltiples errores sintácticos que se producen al escribir mensajes en redes sociales, eliminación de tweets duplicados o simplemente publicidad. Por otro lado, la disponibilidad de diccionarios tanto de términos médicos como de medicamentos es fundamental para realizar esta tarea. Se debe hacer hincapié respecto a los diccionarios de términos médicos ya que en nuestro estudio se ha empleado el incluido en el Metathesaurus del NIH por lo que el usuario cuando escribe en una red social podrá tener infinitas entidades para referirse a un determinado efecto adverso que no esté incluida en esa base de datos. Debido a esto tener un diccionario de términos médicos que agrupen todas estas posibilidades lo hacen en cierta forma irreal.

A pesar de ellos buenos resultados, el sistema puede tener un mejor rendimiento. Gran parte de los tweets obtenidos tuvieron que ser desechados ante la imposibilidad de detectar efectos adversos. El resultado final del sistema sólo detectó efectos adversos en el 15% de los tweets en los que se reconoció una entidad de términos médicos, siendo

habitual estos resultados ya que existen autores que obtuvieron que uno de cada 9 tweets[6].

## 7.2 Posibles mejoras y líneas futuras de desarrollo

El principal objetivo de futuras mejoras debería ser aquellos aspectos en los que peor se desenvuelve el sistema creado. A nivel de arquitectura del sistema, se debería poder contar con un entorno integrado en el que tanto el diccionario de medicamentos como de términos médicos fuesen parte del sistema y no depender de fuentes externas como pueden ser el CIMA en el caso de medicamentos o de MedDRA para los efectos adversos. La idea se basa en que, a mayor nivel de integración, mayor rendimiento se obtenga.

A su vez, el problema en este tipo de textos se basa en el lenguaje coloquial o expresiones que podrían ser incorporados mediante la creación de un lexicon que las agrupa para favorecer la obtención de mejores resultados en la detección de efectos adversos.

Otra mejora podría ser la introducción de mejoras en la detección de las variaciones léxicas. La utilización de librerías destinados a su utilización en textos de idioma inglés y luego adaptadas al español o directamente no adaptadas, representa un gran obstáculo en la mejora de cualquier sistema. Por ejemplo, la lematización representa un arma poderosa para mejorar la detección de términos, sin embargo, en una gran librería empleada en PLN como NLTK no está presente en español, teniendo que recurrir a otros recursos. Eso implica que, por ejemplo, una palabra como *ansiedad* pudiera reconocer variaciones léxicas relacionadas como *ansioso*, *me da ansias*, *ansiolítico* y , de esta forma, ya no dar falsos negativos en la detección.

Por otro lado, aunque los resultados son óptimos en cuanto a las métricas para el algoritmo empleado, se hace necesario el empleo de otro tipo de algoritmos para mejorar el rendimiento. En textos de tamaño grande mayor de 900000 caracteres, existe una imposibilidad de reconocimiento de entidades para equipos más modestos en



## Conclusiones

recursos. La rapidez en la detección adversos puede llegar a ser un factor limitante cuando se intentan clasificar textos que contienen más de 10 mil tweets.

Otras posibilidades serían la inclusión de más entidades, como pueden ser dosis o frecuencia o incluso duración de tratamiento que podrían complementar al sistema y abordar la posibilidad de vincular todas estas entidades, a pesar de que aumentaría la complejidad y las necesidades computacionales del sistema.

Por último, el sistema se ha implementado como una aplicación de escritorio que puede realizar perfectamente su tarea como sistema, aunque podría ser una limitación en el caso de querer implementarla en un entorno más dinámico o con más recursos humanos. Otro tipo de mejoras incluirían la usabilidad, la accesibilidad o la integración del minado de redes sociales en la aplicación. De esta forma, el sistema puede hacer en casi tiempo real, la obtención de textos en la primera etapa y la detección de efectos adversos en la etapa final.



## 8. Presupuesto y cálculo de costes

### 8.1 Descripción del proyecto

Autor: Roi Arias Rico

Departamento: Lenguajes y Sistemas Informáticos

Título: Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español.

Duración: Inicio febrero 2021 y acabó en junio 2022.

### 8.2 Cálculo de costes

#### · Costes de personal

El proyecto fue realizado por una única persona. En la **tabla 15** se indican las etapas y el coste de horas:

Etapas	Coste/hora	Horas totales	Coste total
Análisis y diseño	€60	216	€12960
Codificación		656	€39360
Experimentación y pruebas		80	€4800
Documentación		80	€4800
TOTAL		1032	€61920

*Tabla 15. Costes de personal*

· Costes de equipamiento

Para el desarrollo de este proyecto fue empleado un único ordenador, tanto para la codificación y las pruebas pertinentes, de acuerdo a la **tabla 16**:

Concepto	Unidad	Coste
Ordenador portátil	1	€ 1000
Coste total		€ 1000

*Tabla 16. Costes de equipamiento*

· Costes de software

El coste del sistema operativo se incluye en los costes de hardware. En cuanto al software utilizado propiamente para la programación de este proyecto ha sido mediante el empleo de software libre en la **tabla 17**

Concepto	Unidad	Coste
PyCharm Community version	1	€ 0
Unified Medical Language System (UMLS)	1	€ 0
Anaconda3 2021-Distribution	1	€ 0
Coste total		€ 0

*Tabla 17. Costes de software*

## Presupuesto y cálculo de costes

### · Otros costes

El coste de material fungible como papel, tinta de impresora, costes de encuadernación y otras que no han sido considerados previamente se indica en la **tabla 18**:

Concepto	Unidad	Coste
Papel	1	€ 5
Tinta de impresora	1	€ 25
Costes de encuadernación	1	€ 3
Otros gastos	1	€ 8
Coste total		€ 41

*Tabla 18. Costes de material fungible y otros gastos*

### 8.3 Presupuesto

El presupuesto se indica en la **tabla 19**, de acuerdo a todos los costes anteriormente señalados:

Concepto	Coste
Costes de personal	€61920
Costes de equipamiento	€ 1000
Costes de software	€ 0
Otros costes	€ 41
Coste total	€ 62961

*Tabla 19. Presupuesto del proyecto*

No se ha tenido en cuenta ni los costes indirectos ni los impuestos resultantes de la actividad al ser un proyecto de investigación.



## Bibliografía

- [1] AEMPS. "Buenas Prácticas de Farmacovigilancia del Sistema Español de Farmacovigilancia de medicamentos de uso humano."  
[https://www.aemps.gob.es/vigilancia/medicamentosUsoHumano/docs/BPFV-SEFV\\_octubre-2008.pdf](https://www.aemps.gob.es/vigilancia/medicamentosUsoHumano/docs/BPFV-SEFV_octubre-2008.pdf) (accessed 2022/02/12, 2022).
- [2] C. A. Bond and C. L. Raehl, "Adverse drug reactions in United States hospitals," *Pharmacotherapy*, vol. 26, no. 5, pp. 601-8, May 2006, doi: 10.1592/phco.26.5.601.
- [3] M. McClellan, "Drug safety reform at the FDA--pendulum swing or systematic improvement?," *N Engl J Med*, vol. 356, no. 17, pp. 1700-2, Apr 26 2007, doi: 10.1056/NEJMp078057.
- [4] M. D. Rawlins, "Pharmacovigilance: paradise lost, regained or postponed? The William Withering Lecture 1994," *J R Coll Physicians Lond*, vol. 29, no. 1, pp. 41-9, Jan-Feb 1995. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/7738878>.
- [5] L. Hazell and S. A. Shakir, "Under-reporting of adverse drug reactions : a systematic review," *Drug Saf*, vol. 29, no. 5, pp. 385-96, 2006, doi: 10.2165/00002018-200629050-00003.
- [6] A. Sarker *et al.*, "Utilizing social media data for pharmacovigilance: A review," *J Biomed Inform*, vol. 54, pp. 202-12, Apr 2015, doi: 10.1016/j.jbi.2015.02.004.
- [7] R. Harpaz, W. DuMouchel, N. H. Shah, D. Madigan, P. Ryan, and C. Friedman, "Novel data-mining methodologies for adverse drug event discovery and analysis," *Clin Pharmacol Ther*, vol. 91, no. 6, pp. 1010-21, Jun 2012, doi: 10.1038/clpt.2012.50.
- [8] A. Patki, A. Sarker, P. Pimpalkhute, A. Nikfarjam, and R. Ginn, "Mining adverse drug reaction signals from social media: going beyond extraction," in *Proceedings of BioLinkSig*, 2014.
- [9] S. Yeleswarapu, A. Rao, T. Joseph, V. G. Saipradeep, and R. Srinivasan, "A pipeline to extract drug-adverse event pairs from multiple data sources," *BMC Med Inform Decis Mak*, vol. 14, p. 13, Feb 24 2014, doi: 10.1186/1472-6947-14-13.
- [10] A. Bravo, H. Saggion, and P. Accuosto. "Plan de impulso de las Tecnologías del Lenguaje." <https://plantl.mineco.gob.es/sanidad/Paginas/sanidad.aspx> (accessed 2022/02/13).
- [11] M. Pirmohamed *et al.*, "Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients," *BMJ*, vol. 329, no. 7456, pp. 15-9, Jul 3 2004, doi: 10.1136/bmj.329.7456.15.
- [12] A. Nikfarjam *et al.*, "Early Detection of Adverse Drug Reactions in Social Health Networks: A Natural Language Processing Pipeline for Signal Detection," *JMIR Public Health Surveill*, vol. 5, no. 2, p. e11264, Jun 3 2019, doi: 10.2196/11264.
- [13] A. Benton *et al.*, "Identifying potential adverse effects using the web: a new approach to medical hypothesis generation," (in eng), *Journal of biomedical informatics*, vol. 44, no. 6, pp. 989-996, 2011, doi: 10.1016/j.jbi.2011.07.005.
- [14] R. Xu and Q. Wang, "Large-scale extraction of accurate drug-disease treatment pairs from biomedical literature for drug repurposing," *BMC Bioinformatics*, vol. 14, p. 181, Jun 6 2013, doi: 10.1186/1471-2105-14-181.
- [15] H. Gurulingappa, A. Mateen-Rajput, and L. Toldo, "Extraction of potential adverse drug events from medical case reports," *J Biomed Semantics*, vol. 3, no. 1, p. 15, Dec 20 2012, doi: 10.1186/2041-1480-3-15.

- [16] S. Sohn, J. P. Kocher, C. G. Chute, and G. K. Savova, "Drug side effect extraction from clinical narratives of psychiatry and psychology patients," *J Am Med Inform Assoc*, vol. 18 Suppl 1, pp. i144-9, Dec 2011, doi: 10.1136/amiajnl-2011-000351.
- [17] S. Tuarob, C. S. Tucker, M. Salathe, and N. Ram, "An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages," *J Biomed Inform*, vol. 49, pp. 255-68, Jun 2014, doi: 10.1016/j.jbi.2014.03.005.
- [18] R. Ginn, P. Pimpalkhute, A. Nikfarjam, and A. Patki, "Mining Twitter for adverse drug reaction mentions: a corpus and classification benchmark.," in *Proceedings of the fourth workshop on building and evaluating resources for health and biomedical text processing.*, 2014.
- [19] I. R. Edwards and M. Lindquist, "Social media and networks in pharmacovigilance: boon or bane?," *Drug Saf*, vol. 34, no. 4, pp. 267-71, Apr 1 2011, doi: 10.2165/11590720-000000000-00000.
- [20] C. Tao, M. Filannino, and O. Uzuner, "Prescription extraction using CRFs and word embeddings," *J Biomed Inform*, vol. 72, pp. 60-66, Aug 2017, doi: 10.1016/j.jbi.2017.07.002.
- [21] I. Segura-Bedmar, P. Martinez, R. Revert, and J. Moreno-Schneider, "Exploring Spanish health social media for detecting drug effects," *BMC Med Inform Decis Mak*, vol. 15 Suppl 2, p. S6, 2015, doi: 10.1186/1472-6947-15-S2-S6.
- [22] I. Segura-Bedmar, R. Revert, and P. Martinez, "Detecting drugs and adverse events from spanish social media streams," *Proceedings of the 5th International Louhi Workshop on Health Document Text Mining and Information Analysis*, 2014.
- [23] A. Yates and N. Goharian, "ADRTTrace: detecting expected and unexpected adverse drug reactions from user reviews on social media sites," *Proceedings of the 35th European conference on advances in information retrieval*, pp. 816-9, 2013.
- [24] M. Yang, M. Kiang, and W. Shang, "Filtering big data from social media--Building an early warning system for adverse drug reactions," *J Biomed Inform*, vol. 54, pp. 230-40, Apr 2015, doi: 10.1016/j.jbi.2015.01.011.
- [25] A. Sarker and G. Gonzalez, "Portable automatic text classification for adverse drug reaction detection via multi-corpus training," *J Biomed Inform*, vol. 53, pp. 196-207, Feb 2015, doi: 10.1016/j.jbi.2014.11.002.
- [26] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *J Biomed Inform*, vol. 45, no. 5, pp. 885-92, Oct 2012, doi: 10.1016/j.jbi.2012.04.008.
- [27] S. Karimi, A. Metke-Jimenez, M. Kemp, and C. Wang, "Cadec: A corpus of adverse drug event annotations," *J Biomed Inform*, vol. 55, pp. 73-81, Jun 2015, doi: 10.1016/j.jbi.2015.03.010.
- [28] T. Munkhdalai, O. E. Namsrai, and K. Ryu, "Self-training in significance space of support vectors for imbalanced biomedical event data," *BMC Bioinformatics*, vol. 16 Suppl 7, p. S6, 2015, doi: 10.1186/1471-2105-16-S7-S6.
- [29] Soldaini L. and N. Goharian, "A Fast, Unsupervised Approach for Medical Concept Extraction.," presented at the MedIR Workshop SIGIR, 2016.
- [30] CIMA. "Centro de Informacion de Medicamentos de la AEMPS." AEMPS. <https://cima.aemps.es/cima/publico/nomenclator.html> (accessed 2022-01-31, 2022).
- [31] "www.vademecum.es." (accessed 2022-01-31, 2022).
- [32] NIH. "MedDRA (español)-Unified Medical Language System(UMLS)." NIH. <https://cima.aemps.es/cima/publico/nomenclator.html> (accessed 2022-01-31, 2022).



## Bibliografía

- [33] "Diccionario de términos médicos." Real Academia Nacional de Medicina. <https://dtme.ranm.es/buscador.aspx> (accessed 2022-01-31, 2022).
- [34] "Emoji Sequence Data for UTS #51 Version: 14.0." Unicode. <https://unicode.org/Public/emoji/14.0/emoji-sequences.txt> (accessed 2022-04-25, 2022).
- [35] "es\_core\_news\_md v.3.2.0." SpaCy. [https://github.com/explosion/spacy-models/releases/tag/es\\_core\\_news\\_md-3.2.0](https://github.com/explosion/spacy-models/releases/tag/es_core_news_md-3.2.0) (accessed 2022-04-25, 2022).



## Listado de siglas, abreviaturas y acrónimos

ABPI: Association of the British Pharmaceutical Industry

AEMPS: Agencia Española del Medicamento y Productos Sanitarios

API: Application Programming Interface

ATC (Clasificación): Anatomical Therapeutic Chemical

CIMA: Centro de Información de Medicamentos de la AEMPS

CRF: Conditional Random Forest

FDA: Food and Drug Administration

GUI: Interfaz Gráfica de Usuario

ICD: International Classification of Diseases

ICH: International Conference of Harmonisation

IFPMA: International Federation of Pharmaceutical Manufacturers and Associations

LOINC: Logical Observation Identifier Names and Codes

MedDRA: Medical Dictionary for Regulatory Activities

MeSH: Medical Subject Headings

MIT : Massachusetts Institute of Technology

NER: Named Entity Recognition (Reconocimiento de Entidades Nombradas)

NIH: National Institute of Health

NLTK: Natural Language Tool Kit

PLN: Procesamiento de Lenguaje Natural

PoS: Part of Speech

RAM: Reacción adversa al Medicamento

ROC: Receiver Operating Characteristic (Característica Operativa del Receptor)

SNOMED: Systematized Nomenclature of Medicine

SNOMED CT: Systematized Nomenclature of Medicine – Clinical Terms

Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

SVM: Support Vector Machine

TF-IDF: Term Frequency-Inverse Document Frequency

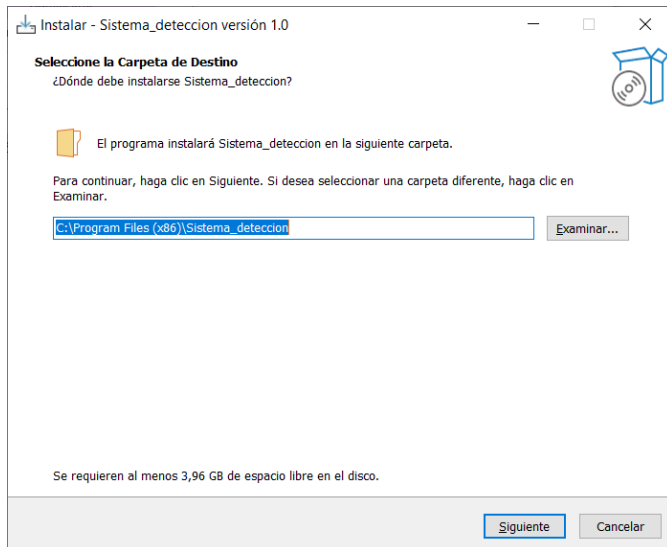
UMLS: Unified Medical Language System

URL: Uniform Resource Locators

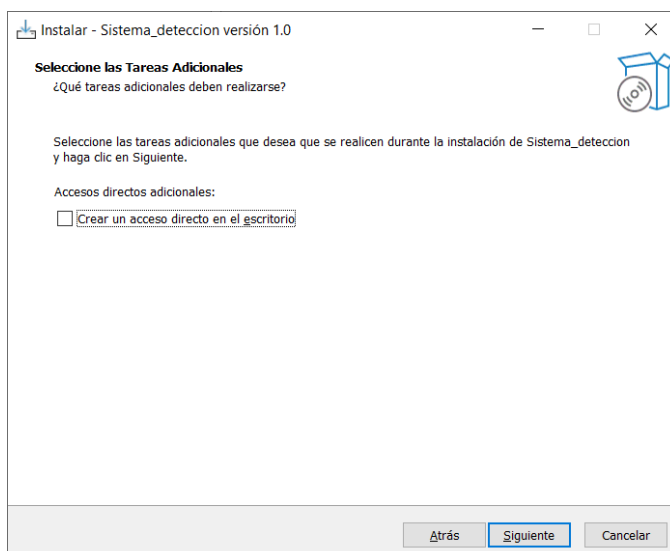
# Anexo

## Anexo A Guía de instalación

Se ejecuta el instalador llamado “SISTEMA.exe” y aparece la siguiente ventana:

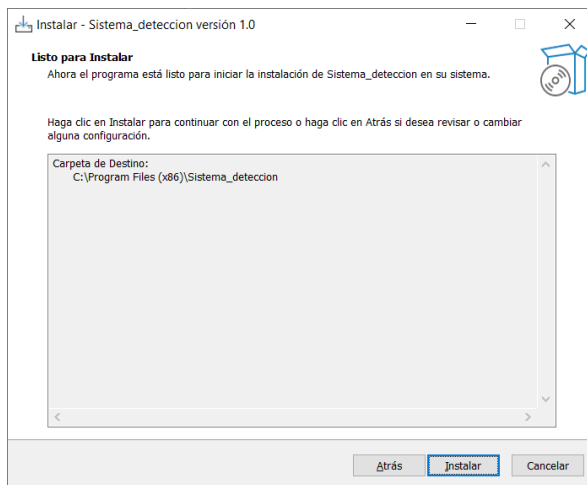


Se presiona “Siguiente” de la ventana de instalación y aparece la siguiente ventana:

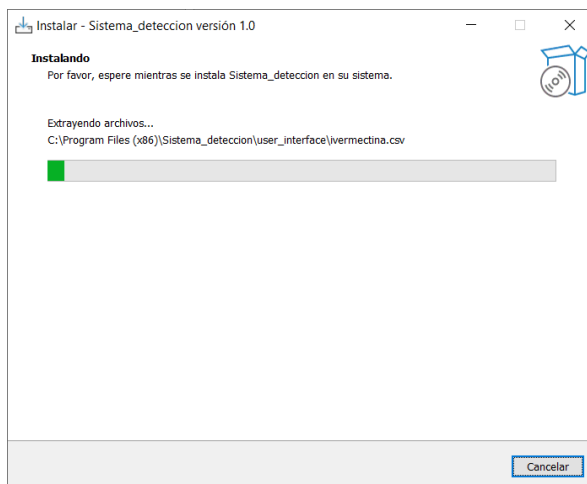


## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

Tras pulsar “Siguiente” , aparece la siguiente ventana:



Se presiona “Instalar” y comienza el proceso de instalación:



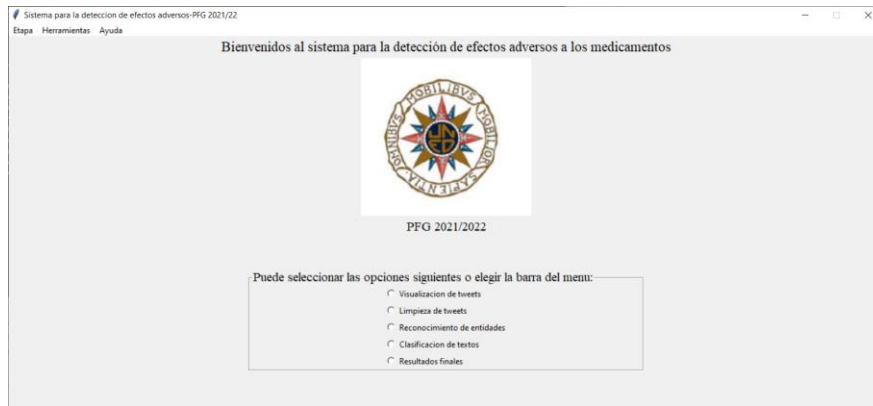
Tras finalizar la instalación, se navega por el directorio de Windows hasta que se llega a su carpeta y se ejecuta el archivo “user\_interface.exe”.

La desinstalación se realiza como cualquier programa instalable de Windows, mediante Panel de control y ,posteriormente, seleccionando el programa.

## Anexo

### Anexo B Manual de usuario

Una vez instalado, el programa se ejecuta con permisos de **Administrador** mediante el archivo “user\_interface.exe”. Tras su ejecución aparece la ventana de bienvenida:

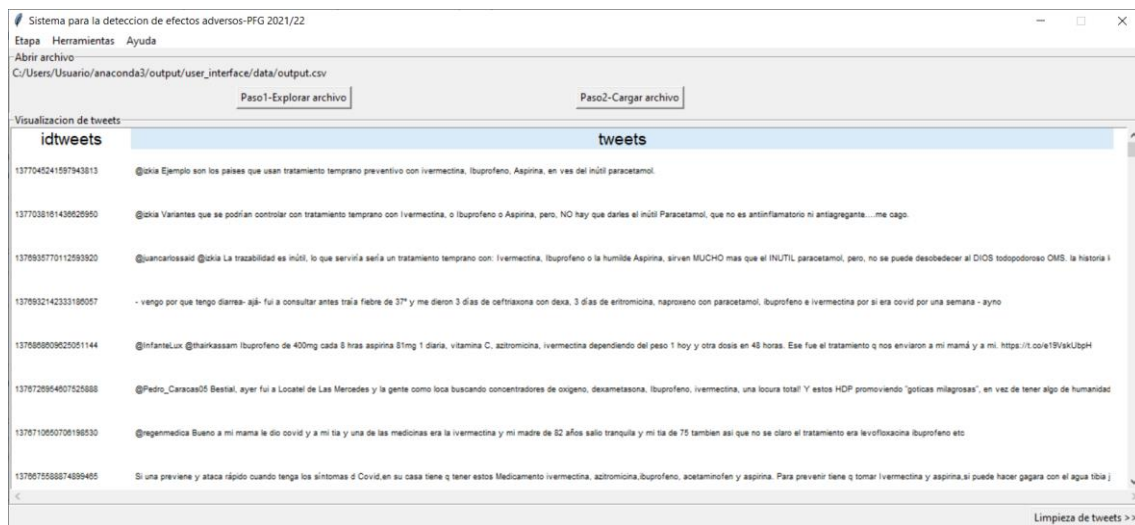


Cada una de las ventanas funciona de forma autónoma y, a la vez, conectada de forma secuencial según las necesidades del usuario, es decir, el usuario podría trabajar de forma continua o guardar el archivo resultante de cada etapa y procesarlo cuando lo necesitase. Las fases se pueden acceder pulsando en la ventana de inicio o a través de las barras de herramientas.

A continuación, se explica con detalle cada una de las ventanas:

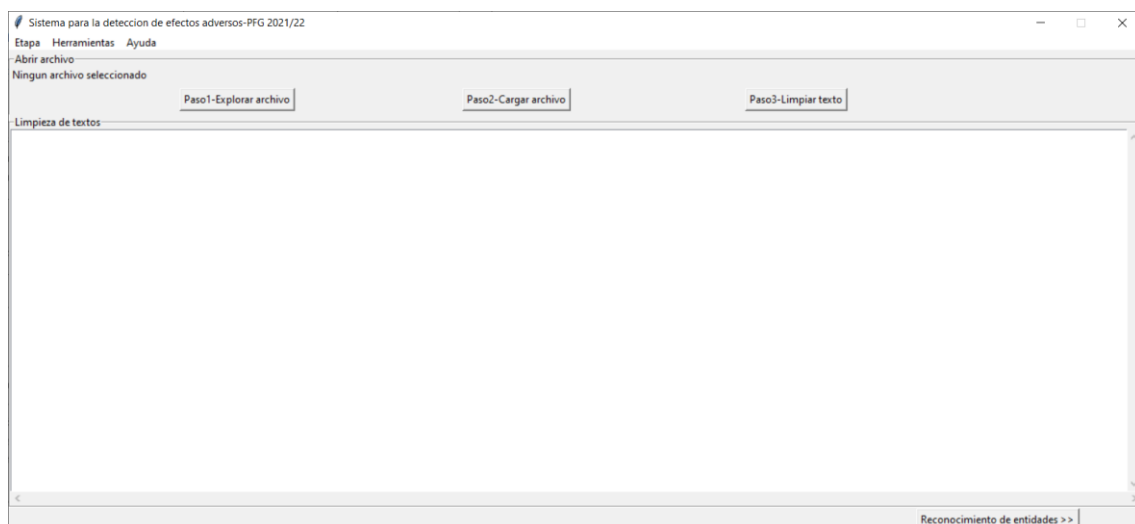
1.- Visualización de tweets: únicamente realiza la visualización de los tweets obtenidos del minado. Se selecciona el archivo mediante “Paso1-Explorar archivo” y , posteriormente, se selecciona su visualización mediante “Paso2-Cargar archivo”. En el proyecto si se explora hasta la carpeta “data” se encuentra el archivo disponible del minado empleado en este proyecto llamado “./data/output.csv”. El resultado de visualizar este archivo es:

## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español



En esta etapa, podemos presionar el botón de “Limpieza de tweets” para ir a la fase siguiente.

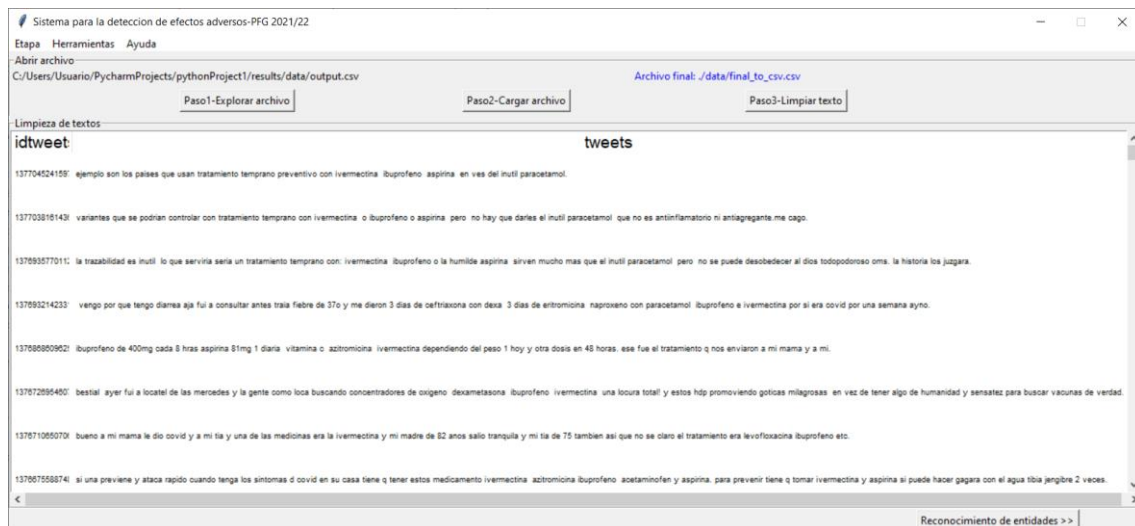
2.-Limpieza de textos: En esta etapa se procede de la misma forma que en la etapa anterior, es decir, se pulsa botón que indica “Paso 1-Explorar archivo”, luego, el botón que indica “Paso 2-Cargar archivo” y , por último, se presiona el “Paso 3-Limpiar archivo”.



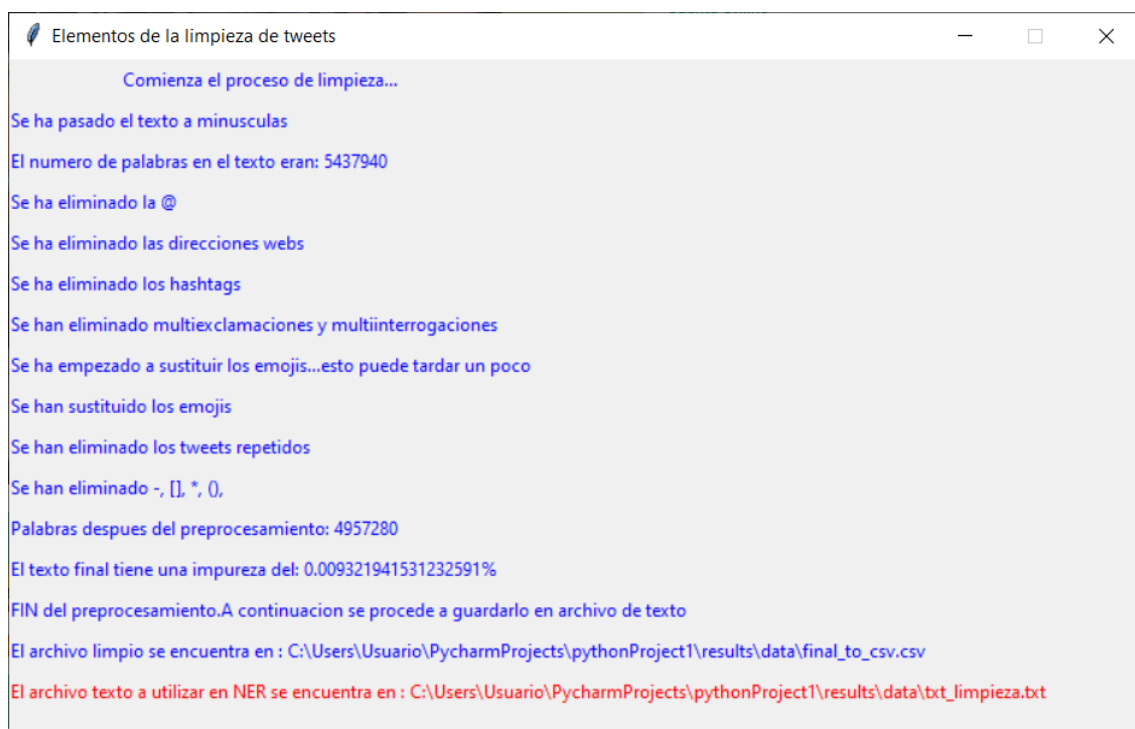
Los resultados de la siguiente visualización, tras cargar los datos y proceder a la limpieza del archivo utilizado en la etapa anterior y disponible en la carpeta de la instalación (“./data/output.csv”), se observan en la siguiente figura:



## Anexo



Aparece en la parte superior en azul el archivo final con terminación .csv, además se generará un archivo .txt (ambos contenidos en la carpeta data). Además, una vez terminada la etapa de limpieza se indicará en una ventana emergente los procesos de limpieza realizados:



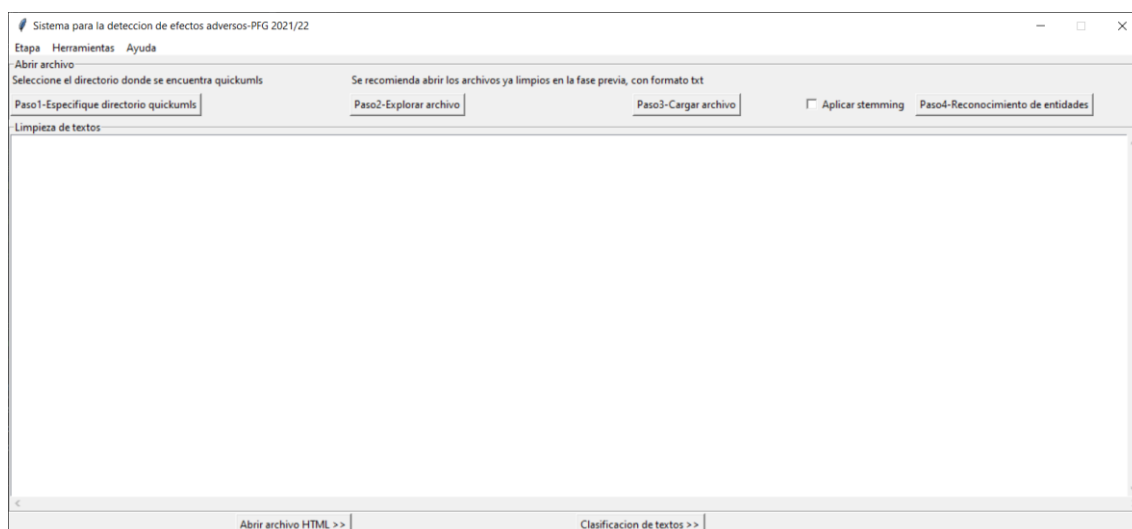
A continuación, podemos acceder a la etapa de “Reconocimiento de entidades si así se desea, presionando el botón correspondiente “Reconocimiento de entidades”.

3.-NER: en esta etapa se emplea la herramienta QuickUMLS, por lo que es necesario que instale esta herramienta para su uso en esta etapa. El archivo de

## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

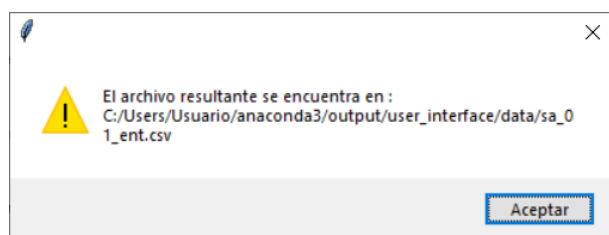
instalación incorpora una carpeta denominada “quick” donde se encuentran los archivos necesarios para ejecutar. No obstante, si el usuario así lo decidiese se podría utilizar otro directorio que el usuario disponga.

Debido al uso de aplicaciones externas y el empleo de grandes cantidades de texto que obligarían a un uso extensivo de memoria, se han limitados los archivos a 900000 caracteres. En el caso de que el archivo lo supere, se ha ideado unas herramientas que dividen el archivo en partes iguales y que , luego, los resultados de NER pueden ser concatenado para la etapa final. Los pasos a seguir difieren un poco que en etapas anteriores, lo primero es seleccionar donde se encuentra nuestros datos QuickUMLS que es el “Paso 1- Especifique directorio quickumls” (generará un error en el caso de que no haya sido cargado correctamente cuando se proceda al paso 4. Posteriormente, se selecciona el archivo que se desea reconocer sus entidades “Paso2-Explorar archivo” y ,una vez seleccionado, se presiona el “Paso3-Cargar archivo”



Tras este paso, se observa como se ha cargado en la pantalla nuestro archivo, en el caso de este manual se ha llamado “./data/txt\_limpieza.txt”, debido a que este archivo tiene más de 900000 caracteres, se procede a su división en partes iguales, empleando el módulo de “Herramientas” (ver paso en el apéndice dedicado a “Herramientas”). Además, tenemos la opción de aplicar *stemming* aunque no siempre es aconsejable (en el caso del trabajo original el *stemming* generó más problemas que beneficio, por lo que no se utilizó). Se genera como paso intermedio un archivo HTML, que se encuentra en

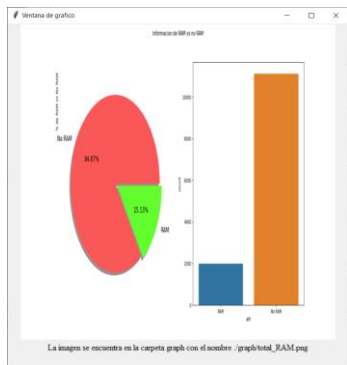
la carpeta “data” y que se llama “NER\_displacy” (ruta: “./data/NER\_displacy.html”). Como advertencia al usuario, esto es un proceso lento, más lento a medida que aumenta el número de entidades a reconocer y el tamaño del texto. Como resultado de esta etapa, se genera un archivo con un nombre acabado en “\*\_ent.csv”. En el ejemplo del modelo, se emplearon 18 archivos que tienen el formato “sa\_”+número+”.txt” que tendrá como resultado de esta etapa un archivo llamado “sa\_”+número+”\_ent.csv”, disponibles en la carpeta data



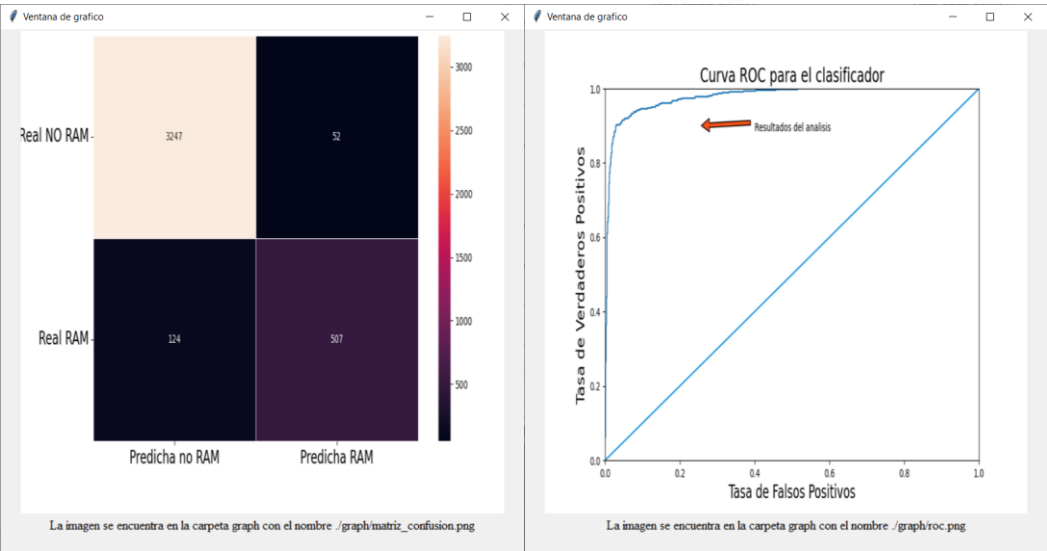
En esta etapa NER, también podemos concatenar estos archivos resultantes para el último paso, para ello se presiona el botón “Concatenar resultados NER”, el resultado final de esta etapa será un archivo final en la ruta “./data/ext\_full.csv”. Otros archivos generados consisten en la lista de medicamentos encontrados (en la ruta “./data/med\_data.json”), lista de entidades biomédicas encontradas (en la ruta “./data/lista\_umls.txt”), así como se guardarán los componentes (pipe) de la pipeline utilizada en las carpetas nlp\_drug (para fármacos) y nlp\_terminos (para los términos biomédicos).

4.-Clasificación de textos: Se repiten los pasos, es decir, “Paso1- Explorar archivo”, posteriormente “Paso 2-Cargar archivo” y , finalmente, “Paso3-Clasificación”. Esta es la etapa más lenta ya que se emplean algoritmos y clasificadores que enlentecen el proceso. Empleando el archivo resultante de la etapa anterior “ext\_full.csv”, se obtienen varias gráficas, la primer a el número de efectos adversos detectados:

Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

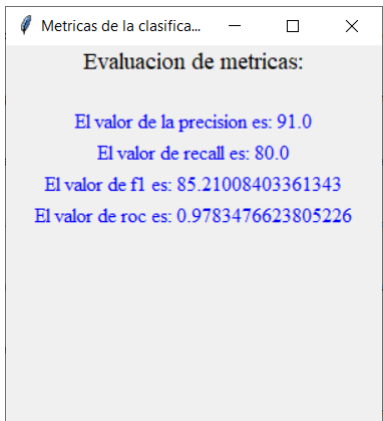


Otra gráfica obtenida sería la matriz de confusión y la curva ROC del algoritmo utilizado con esos textos:



Estas gráficas se guardarán en la carpeta llamada graph.

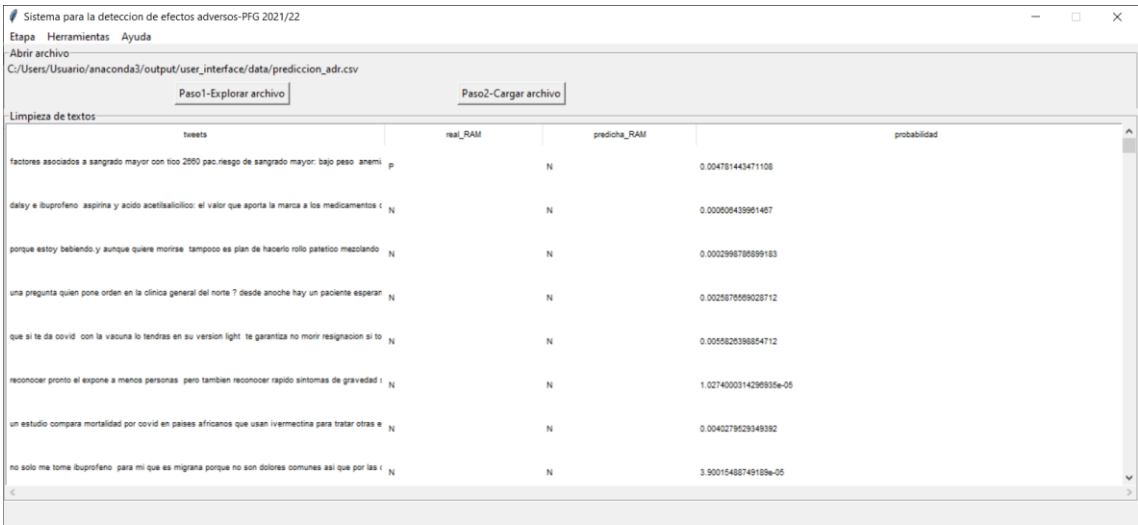
En esta etapa, también se pueden visualizar las métricas obtenidas pulsando al botón “Opcional-Ver métricas” y se obtiene una ventana:



## Anexo

Finalmente en esta etapa se puede utilizar se puede obtener el resultado final mediante un archivo en la ruta “./data/prediccion\_adr.csv” que contiene el mensaje original, el efecto adverso conocido, el efecto adverso predicho y la probabilidad de predicción. Este archivo se podrá visualizar en la siguiente etapa. Además hay un archivo intermedio denominado “./data/ext\_full\_adr.csv” que contiene el mensaje original y las entidades dosis, medicamentos y UMLS.

5.- Resultados finales: Mismo proceso de etapas anteriores, se procede a “Paso 1-Explorar archivo” y , luego, “Paso 2-Cargar archivo” y se visualiza en el cuadro:



tweets	real_RAM	predicha_RAM	probabilidad
factores asociados a sangrado mayor con tico 2690 pac./riesgo de sangrado mayor: bajo peso anemi	p	N	0.004781443471108
delay e ibuprofeno aspirina y acido acetilsalicilico: el valor que aporta la marca a los medicamentos t	N	N	0.00808439961487
porque estoy bebiendo y aunque quiere morirse tampoco es plan de hacerlo rollo patetico mezclando	N	N	0.002398788899183
una pregunta quien pone orden en la clinica general del norte ? desde anoche hay un paciente esperan	N	N	0.0025870589028712
que si te da covid con la vacuna lo tendras en su version light te garantiza no morir resignacion si lo	N	N	0.0055826388854712
reconocer pronto el expone a menos personas pero tambien reconocer rapido sintomas de gravedad i	N	N	1.0274000314296935e-06
un estudio compara mortalidad por covid en paises africanos que usan ivermectina para tratar otras e	N	N	0.0040279528348382
no solo me tome ibuprofeno para mi que es migraña porque no son dolores comunes así que por las i	N	N	3.90015488749189e-05

Aunque no forma parte del proceso, hay una ventana de “Herramientas” que presenta las siguientes utilidades:

- Dividir archivo: Se emplea cuando los archivos de texto son demasiado grandes para el NER. Divide el archivo en partes iguales que pueden ser procesados.
- *Webscrapping* para diccionario de medicamentos y principios activos: Se ha empleado en este trabajo a partir de la página web vademecum.org, que mantiene una base de datos actualizada de medicamentos comercializados en España.
- Minado de twitter: Pulsando el botón “Cargar código de scrapping de twitter” se accede al código fuente empleado para el minado de los tweets. El hecho de no incluirlo implementado se debe a que se necesitan las claves que Twitter proporciona de forma individual a cada usuario por lo que se omite en base a la confidencialidad.

## Sistema para la detección de efectos adversos a medicamentos en textos biomédicos en español

Como nota de rendimiento del programa, el programa pierde rendimiento a medida que aumenta el número de caracteres del textos. El procesamiento en un entorno Windows 10 y 32 GB de memoria RAM de 13098 tweets en la última etapa de clasificación llevan unos 360-400 segundos.