UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Detección Automática de Información Sensible en Redes Sociales

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Víctor Botti Cebriá
Tutora: Ana María García Fornes
Directora Experimental: Elena Del Val Noguera

Curso 2018-2019

Resum

La detecció d'informació sensible tenint en compte la privacitat és un tema relevant en Xarxes Socials. Moltes vegades és difícil per als usuaris administrar la privacitat associada a les seues publicacions en xarxes socials tenint en compte les posibles conseqüències. L'objectiu principal d'aquest treball és proporcionar als usuaris informació sobre la sensibilitat de la informació que compartiràn quan decidisquen publicar un missatge de text en mitjans online. Per a això es planteja el desenvolupament de una eina que permet analitzar la sensibilitat sobre la base dels diferents tipus d'informació (categories) que es detecten en el missatge (i.e.,ubicació, dades personals, salut, atacs personals, emocions etc.). Per a la detecció de les diferents categories es farà ús de libreries de reconeixement d'entitats, ontologies, diccionaris i anàlisis de sentiment. Aquesta eina s'avaluarà mitjançant un dataset elaborat a partir de missatges de la xarxa social Twitter. Finalment, s'integrarà en la xarxa social Pesedia, dirigida a la infància i adolescència, per a proporcionar informació als usuaris sobre el risc de publicar un determinat contingut i ajudar-los en la presa de decisions de la seua publicació.

Paraules clau: Privacitat, Sensibilitat de la informació, Xarxes socials, Classificació

Resumen

La detección de información sensible teniendo en cuenta la privacidad es un tema relevante en Redes Sociales. Muchas veces es difícil para los usuarios administrar la privacidad asociada a sus publicaciones en redes sociales teniendo en cuenta sus posibles consecuencias. El objetivo principal de este trabajo es proporcionar a los usuarios información acerca de la sensibilidad de la información que van a compartir cuando deciden publicar un mensaje de texto en medios online. Para ello se plantea el desarrollo de una herramienta que permita analizar la sensibilidad en base a los distintos tipos de información (categorías) que se detecten en el mensaje (i.e.,ubicación, datos personales, salud, ataques personales, emociones etc.). Para la detección de las distintas categorías se hará uso de librerías de reconocimiento de entidades, ontologías, diccionarios y análisis de sentimiento. Esta herramienta se evaluará mediante un dataset elaborado a partir de mensajes de la red social Twitter. Finalmente, se integrará en la red social Pesedia, dirigida a la infancia y adolescencia, para proporcionar información a los usuarios sobre el riesgo de publicar un determinado contenido y ayudarles en la toma de decisiones de su publicación.

Palabras clave: Privacidad, Sensibilidad de la información, Redes sociales, Clasificación

Abstract

Detecting sensitive information with privacy in mind is a relevant issue on Social Networks. It is often difficult for users to manage the privacy associated with their posts on social networks taking into account their possible consequences. The main objective of this work is to provide users information about the sensitivity of the information they will share when they decide to publish a message in online media. For this purpose, the development of a tool to measure sensibility based on the different types of information (categories) detected in the message (i.e., location, personal data, health, personal attacks, emotions, etc.) is proposed. Entity recognition libraries, ontologies, dictionaries and sentiment analysis will be used to detect the different categories. This analyzer will be evaluated with a dataset elaborated from messages of the social network Twitter. Finally, it will be integrated into the social network Pesedia, aimed for children and teenagers, to

provide information to users about the risk of publishing a certain content and help them in making decisions about its publication.

Key words: Privacy, Information sensitivity, Social networks, Classification

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

1.1 Motivación

Es innegable que las redes sociales se han convertido en un pilar de la sociedad moderna. Están presentes de manera continuada en el día a día de las personas debido principalmente al auge de las nuevas tecnologías [9, 28].

Durante este uso de las redes sociales, los usuarios hacen publicaciones o leen las de otros y pueden contar lo que quieran sin filtros [12]. Esto puede hacer que los usuarios publiquen información sobre su ubicación, su estado de salud o su ideología, lo cual puede ser usado por otras personas o entidades con fines no benévolos. [13, 26].

Existen una gran cantidad de campañas para concienciar a la gente sobre las implicaciones de compartir sus datos personales en las redes sociales [21, 32, 31]. Sin embargo, muchos usuarios siguen publicando información personal debido a que realmente no son conscientes de que lo que publican está revelando información sobre ellos y la repercusión que puede tener [17]. Algunos estudios consideran que la concienciación y la confianza no necesariamente promueven comportamientos menos arriesgados, especialmente entre los jóvenes [14]. Este resultado está línea con el número de jóvenes que reportan experiencias negativas a pesar de las iniciativas llevadas a cabo por las campañas educativas.

Como alternativa a los materiales y campañas educativas, se ha considerado que herramientas o mecanismos integrados en las redes sociales que asistan a los usuarios para tomar mejores decisiones en materia de privacidad pueden reducir la exposición a riesgos de privacidad [1, 5]. Específicamente, las intervenciones *soft-paternalism* [1] han sido consideradas como un método adecuado para influir en los comportamientos de privacidad de los usuarios sin que pierdan la libertad de elección.

En este trabajo se propone resolver el problema de compartir publicaciones de texto que contengan información sensible en medios sociales utilizando técnicas basadas en la idea de *soft-paternalism*. La herramienta que se presenta analiza el contenido de la publicación, detecta si hay presente información perteneciente a categorías potencialmente sensibles e informa al usuario para ayudarle en el proceso de toma de decisiones.

El trabajo realizado en este proyecto se enmarca dentro de dos proyectos de investigación realizados en el grupo de investigación GTI-IA [1]. El primer proyecto es *PESEDIA: Privacidad en Entornos Sociales EDucativos durante la Infancia y la Adolescencia* (TIN2014-55206-R). En este proyecto se desarrolló la plataforma de la red social Pesedia que se utiliza en este trabajo. El segundo proyecto *Agentes inteligentes para asesorar en privacidad en redes sociales* (TIN2017-89156-R) está centrado en el desarrollo de agentes que proporcio-

[1] http://gti-ia.upv.es

nen información al usuario sobre los riesgos de privacidad que potencialmente pueden surgir cuando realizan determinadas acciones en la red social Pesedia. La herramienta que se desarrolla en este proyecto se ha integrado en la red social y podrá ser utilizada por los agentes de la red para ayudar asistir a los usuarios en la toma de decisiones en la red.

1.2 Objetivos

El objetivo principal de este trabajo es proporcionar a los usuarios información acerca de la sensibilidad, desde el punto de vista de la privacidad, de la información que van a compartir cuando deciden publicar un mensaje de texto en medios en línea. Para ello se plantea el desarrollo de una herramienta que integrará una medida de sensibilidad en base a los distintos tipos de información (categorías) que se detecten en el mensaje (i.e., ubicación, datos personales, creencias, etc.). Para la detección de las distintas categorías se hará uso de librerías de reconocimiento de entidades, ontologías, diccionarios y análisis de sentimiento. Este servicio se integrará en la red social Pesedia [4] para proporcionar información a los usuarios sobre el riesgo de publicar un determinado contenido y ayudarles en la toma de decisiones de su publicación.

Para alcanzar este objetivo se han planteado los subobjetivos que se describen a continuación:

- Analizar propuestas similares realizadas en otros trabajos.
- Extraer, limpiar y almacenar los textos obtenidos de redes sociales.
- Analizar tipos de información contenida en mensajes de texto.
- Analizar herramientas existentes para la detección de información, en caso de no encontrar herramientas desarrollarlas.
- Implementar la herramienta desarrollada como un servicio web.
- Desarrollar un plug-in para acceder a la herramienta desde la red social Pesedia.
- Informar al usuario sobre la información revelada a partir de un mensaje de texto.
- Estudiar medidas para la evaluación de un dataset.
- Estudiar técnicas para comprobar el correcto funcionamiento de la herramienta desarrollada.
- Realizar pruebas de rendimiento de la herramienta.

1.3 Metodología

Para la realización del proyecto se ha utilizado el desarrollo en cascada, o también llamado secuencial, debido a que se ordenan las etapas del desarrollo de software de forma que una etapa debe esperar a la finalización de la etapa anterior.

Las etapas que se han seguido han sido las siguientes:

1. **Análisis de requisitos**. En esta fase se analiza qué es lo que necesitan los usuarios finales para así determinar que objetivos se deben cumplir.

2. **Diseño del programa**. En esta fase se realizan los algoritmos necesarios además de investigar qué herramientas pueden ser útiles en la siguiente etapa.
3. **Codificación**. En esta fase se implementa el código de la aplicación haciendo uso de librerías y otros componentes.
4. **Pruebas**. Una vez ya se encuentra la herramienta programada, se le realizan pruebas para comprobar su correcto funcionamiento. En caso de que se produzcan errores se corrigen.
5. **Verificación del programa**. En esta última fase es donde el usuario final ejecuta el sistema y se comprueba que cumpla con sus necesidades.

1.4 Estructura

Teniendo en cuenta la motivación, los objetivos y la metodología presentados de este trabajo, el resto del documento se organiza como se muestra a continuación:

- El capítulo 2 hace un recorrido sobre distintos trabajos de investigación relacionados con el tema de la privacidad y la detección de la sensibilidad en redes sociales.
- El capítulo 3 explica el análisis del problema y se realiza la especificación de requisitos junto con un análisis del marco legal y ético.
- El capítulo 4 presenta el diseño de la solución propuesta y explica cómo se ha desarrollado la herramienta.
- El capítulo 5 describe cómo realizar la implantación de la herramienta.
- El capítulo 6 describe las pruebas que se han realizado sobre la herramienta para evaluar su rendimiento.
- El capítulo 7 contiene las conclusiones. En este punto se habla sobre su relación con las asignaturas del grado y sobre trabajos futuros que se pueden hacer para ampliar la herramienta.

CAPÍTULO 2
Estado del arte

Los usuarios muchas veces no son conscientes de la sensibilidad de la información que comparten en las redes sociales ni de su repercusión [15]. Un usuario de una red social podría compartir un mensaje de texto con otro usuario en el que confía. Sin embargo el usuario inicial no sabe con seguridad hasta dónde puede llegar la información que ha compartido. Además, es posible que el usuario inicial tampoco se dé cuenta de la cantidad de información que puede estar contenida de manera implícita y/o explícita en el mensaje que ha compartido [27, 33]. En estos casos, es importante comprender los potenciales riesgos a los que un usuario se puede ver expuesto a la hora de compartir información personal en la red. Por esta razón, el uso de indicadores de riesgo y sensibilidad pueden ayudar a un usuario a decidir si está dispuesto a compartir determinados mensajes.

La sensibilidad de la información puede ser considerada como la pérdida potencial asociada con la divulgación de información. Esta definición permite que la información sensible sea percibida como más arriesgada y más incómoda de divulgar [20]. Generalmente, por definición, los datos personales son más sensibles que los datos. Los datos se utilizan a menudo para referirse a información cuantificada y almacenada digitalmente. En este trabajo, utilizamos datos e información como un mismo concepto. En lo que respecta a los datos personales, se trata de información que, directa o indirectamente, puede vincularse a un individuo y que puede identificarlo específicamente. Desde el punto de vista legal, los países han sido forzados a regular las actividades que implican la recogida, el almacenamiento y manipulación de la información personal. Una de estas regulaciones en la Unión Europea es la RGPD (Reglamento General de Protección de Datos). Según esta regulación, los datos que se consideran más sensibles son: raza, opiniones políticas, opiniones religiosas, otras creencias, salud o sexo [11].

En el área de las redes sociales se han propuesto varios mecanismos y medidas para tratar de ayudar a los usuarios en la toma de decisiones a la hora de compartir información sensible.

Alemany et al. [4] presentan una medida de riesgo de privacidad que tiene en cuenta el alcance potencial (i.e., audiencia) de una publicación que realiza el usuario teniendo en cuenta no sólo la política de privacidad que el usuario asocia a la publicación (i.e., público, amigos, privado, etc.) si no también la posición en la estructura de la red social y los flujos de información. Aunque la medida que se propone puede orientar al usuario sobre el riesgo potencial de publicar, esta medida no tiene en cuenta la sensibilidad del contenido que se va a compartir.

Talukder et al. [30] presentan una herramienta llamada Privometer para proteger la privacidad en Facebook (ver Figura 2.1). En esta red social podía existir software malicioso, que haciendo uso de la API de Facebook obtenía datos de los usuarios a través de

las cuentas de amigos y conocidos. Para intentar solucionar este problema se desarrolló Privometer, una herramienta que mide la cantidad de información sensible que se está revelando en el perfil de usuario, crea listas donde se puede ver qué personas agregadas tienen más posibilidad de revelar información y sugiere cambios en el perfil de usuario para aumentar la privacidad y disminuir lo máximo posible la cantidad de información sensible que pueda ser utilizada por otros. Esta herramienta solamente servía para medir la información revelada en los perfiles de los usuarios, no en los mensajes que publicaba.

Figura 2.1: Privometer en Facebook.

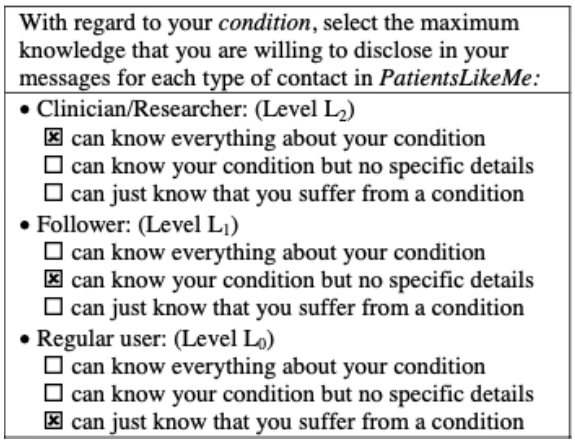
Pensa et al. [22] proponen un marco de referencia para mantener la información sensible de los usuarios bajo control en las redes sociales. Este marco tiene principalmente dos funciones. La primera consiste en el cálculo de una medida de privacidad. Esta medida dependerá de la información que esté revelando cada usuario, y dependiendo del valor que tenga se le advertirá al usuario dónde se está revelando más información. La segunda función consiste en ayudar a los usuarios a configurar la privacidad de sus perfiles en las redes sociales haciendo que sea lo más sencillo posible y limitando el número de operaciones que se tienen que hacer manualmente. El problema que no resuelve este trabajo es que no se proponen formas de detectar si un usuario revela información al comunicarse con otros y no solamente en el perfil. Además no se desarrolló ninguna herramienta, solamente se propusieron de forma teórica formas de implementar una herramienta para ayudar a los usuarios.

Mao et al. [16] presentan un análisis sobre la revelación de información sensible en redes sociales, en concreto sobre Twitter. Lo que se hace en el trabajo es entender qué tipo de información revelan los usuarios y posteriormente categorizarla según lo que se revele. En concreto, se detectan tres tipos de información: planes sobre las vacaciones, información sobre el estado de salud, y mensajes publicados bajos los efectos del alcohol o las drogas. A continuación, habiendo hecho este análisis, se desarrolla una herramienta para detectar automáticamente estos tres tipos de información en tweets que se estén publicando a tiempo real. Con este trabajo se quiere lograr poder advertir a estos usuarios sobre lo que están publicando, y de esta manera poder usar estos clasificadores como un mecanismo de defensa para la revelación de información sensible en redes sociales. En este artículo si que se han llegado a desarrollar clasificadores para detectar información sensible, pero solamente han identificado tres categorías, lo cual no incluye todo el tipo de información que un usuario puede revelar en las redes sociales. Además este estudio se ha desarrollado haciendo uso de tweets en Ingles, por lo que los clasificadores solamente

funcionan en este idioma y no se ha llegado a implementar ninguna forma de informar a los usuarios cuando vayan a publicar información sensible.

Caliskan et al. [8] realizan un estudio para identificar si un texto contiene información sensible haciendo uso de modelado de temas, reconocimiento de nombres, ontologías de privacidad, análisis de sentimiento y normalización de texto. Para ello, han utilizado 500.000 tweets de 100.000 usuarios en los que se incluyen seguidores y relaciones. Usando estos tweets, han clasificado los tipos de información que revelan los usuarios en distintas categorías: gente, deportes, ficción, diversión, emociones, localización, discusiones, insultos, noticias, tiempo, detalles personales, religión y familia. Además una vez acabado el estudio, comprueban qué usuarios son los que revelan más información y descubrieron que esto no depende del número de seguidores que tenga un usuario, sino de sus relaciones, cuanta más información revelen sus conocidos más información revelará el usuario. Este artículo presenta una interesante clasificación de la información revelada en distintas categorías, por esta razón más adelante las usaremos para desarrollar el proyecto. Pero estos clasificadores presentados en el artículo funcionan con textos en inglés, y no se han implementado en ninguna red social, por lo que no se utiliza para ayudar a los usuarios a la toma de decisiones.

Sanchez et al. [25] estiman la sensibilidad de la información que el usuario comparte a partir de su percepción del riesgo y de la relación de confianza que tiene con los usuarios con los que van a compartir la información. Para determinar la percepción de riesgo y las relaciones de confianza, los usuarios tienen que responder a una serie de preguntas predefinidas sobre temas sensibles para cada tipo de contacto. Específicamente, el sistema presenta una serie de preguntas relacionadas con diferentes temas sensibles (por ejemplo, la religión, raza, sexualidad, historia clínica, etc.) y, para cada uno de ellos, el usuario tiene que decidir con qué tipo de usuario (seguidor, investigador, etc) compartiría esta información y lo aplican a la red social *PatientsLikeMe*. Podemos observar un ejemplo de este tipo de preguntas en la Figura 2.2. Este trabajo, aunque és implementado en una red social y ayuda a los usuarios con su privacidad, solamente funciona en inglés y está centrado en la privacidad de los perfiles de usuario y no en sus publicaciones.


With regard to your *condition*, select the maximum knowledge that you are willing to disclose in your messages for each type of contact in *PatientsLikeMe:*

- Clinician/Researcher: (Level L_2)
  - ☒ can know everything about your condition
  - ☐ can know your condition but no specific details
  - ☐ can just know that you suffer from a condition
- Follower: (Level L_1)
  - ☐ can know everything about your condition
  - ☒ can know your condition but no specific details
  - ☐ can just know that you suffer from a condition
- Regular user: (Level L_0)
  - ☐ can know everything about your condition
  - ☐ can know your condition but no specific details
  - ☒ can just know that you suffer from a condition

Figura 2.2: Pregunta utilizada por Sánchez et al.

Los trabajos anteriores presentan propuestas para afrontar el problema de la pérdida de privacidad a la hora de compartir información en las redes sociales. La mayoría de ellos comprueban los ajustes de privacidad de los usuarios para obtener métricas y puntuaciones. Sin embargo, esto no ayuda a concienciar o advertir a un usuario a no publicar información sensible, debido a que estas herramientas lo que hacen es ayudar al usuario a modificar su configuración de privacidad. Por ejemplo, permiten hacer que las publicaciones sólo sean visibles para amigos. Sin embargo, en las redes sociales se agrega a gente

como "amigos" con la que realmente no se tiene contacto y puede que no debamos compartir cierto tipo de información con ellos. Por esta razón, algunas de las herramientas y estudios propuestos anteriormente no están completos, ya que no realizan un análisis de las publicaciones o los textos que el usuario pretende publicar. El estudio que sí se centra en las publicaciones de los usuarios es el realizado por Caliskan et al. [8], pero no realizan ninguna herramienta para ayudar directamente a los usuarios, sino para encontrar qué tipos de información sensible son publicados en la red.

Nuestra propuesta se diferencia de las anteriores en los siguientes aspectos: (i) la medida de sensibilidad de la información tiene en cuenta qué tipos de información publica el usuario haciendo uso de las categorías presentadas por Caliskan et al. [8] y se ha integrado en una red social ; (ii) el módulo de clasificación trabaja con mensajes en castellano a diferencia de muchos de los trabajos previos que se centran en mensajes en inglés; (iii) es capaz de identificar términos específicos dentro de un mensaje que pueden causar riesgo de privacidad de manera que los usuarios de las redes sociales puedan tener una visión más clara de las amenazas a la privacidad de su publicación; (iv) se generan mensajes informativos para que el usuario pueda tomar decisiones informadas. Se puede ver de forma más clara las diferencias con el resto de trabajos realizados en la Tabla 2.1.

	Medida de sensibilidad	**Mensajes informativos**	**Integración en red social**	**Idioma**
Alemany et al.	✗	✓	✓	-
Talukder et al.	✓	✓	✓	Inglés
Pensa et al.	✓	✓	✗	Inglés
Mao et al.	✗	✓	✗	Inglés
Chow et al.	✗	✗	✗	Inglés
Liu et al.	✓	✗	✗	Inglés
Caliskan	✗	✗	✗	Inglés
Sanchez et al.	✓	✗	✓	Inglés
Herramienta desarrollada	✓	✓	✓	Español

Tabla 2.1: Comparación de la herramienta desarrollada con otros trabajos.

CAPÍTULO 3
Análisis del problema

Actualmente los usuarios publican información personal sin darse cuenta de la importancia de esta y posteriormente se arrepienten de haberla publicado [27, 33]. Haciendo uso de ciertas herramientas, se puede concienciar a los usuarios sobre la privacidad en las redes sociales y hacer que sean conscientes de qué tipo de información es conveniente publicar y cual no.

Por esta razón en este proyecto se plantea el desarrollo de una herramienta que realiza un análisis del texto de un mensaje que un usuario quiera publicar y advierte al usuario en el caso que revele información potencialmente sensible. Una vez el usuario sabe esto, puede decidir si quiere publicar el mensaje o prefiere modificarlo.

En este capítulo se describen los puntos más importantes que forman parte de la especificación de requisitos de la herramienta que se presenta en el TFG. La especificación de requisitos nos permite realizar una descripción completa del comportamiento de la herramienta software. Para realizar la siguiente especificación de requisitos del proyecto se ha seguido el estándar IEEE830 [18]. También se realizará un análisis del marco legal y ético de la herramienta.

3.1 Especificación de requisitos

3.1.1. Introducción

Propósito

En las siguientes secciones se definirán las especificaciones funcionales, no funcionales y del sistema para la implementación de una herramienta que permitirá analizar una cadena de texto y proporcionar información acerca de la sensibilidad. Esta herramienta será integrada en la red social Pesedia por medio de un plug-in y será utilizada por los usuarios de la red social. También podrá ser utilizada como un servicio web independiente.

Ámbito del sistema

La herramienta desarrollada en este trabajo consiste en un módulo que detecta la presencia de determinadas categorías de información en un texto, y en base a esas categorías informa si el texto contiene información sensible o no. El módulo se implementará como un servicio web. Además, el servicio podrá ser llamado desde la red social Pesedia. Para

ello, también se desarrollará un plug-in que permita la comunicación entre el servicio web y la red social.

El módulo hará una evaluación de una cadena de texto que quiera publicar un usuario, devolviendo a continuación un mensaje de advertencia. De esta manera el usuario podrá modificar la cadena de texto que iba a publicar si ve que se esta revelando información sensible.

En el caso de que el módulo se utilice en una red social, éste no bloqueará el envío de mensajes aun cuando se detecte que se está revelando información sensible. En ningún caso se guardará información sobre los mensajes que han enviado los usuarios al módulo para evaluar.

El objetivo de la herramienta es poder ayudar a los usuarios a identificar que están revelando información, la cual puede hacer que se vean comprometidos y causarles problemas.

Definiciones, acrónimos y abreviaturas

- **Pesedia**: es una red social desarrollada por el grupo de investigación GTI-IA [1], la cual se va a utilizar para probar la herramienta desarrollada.
- **Plug-in**: un plug-in es un complemento para añadir funcionalidad a una aplicación.
- **Librería**: es una aplicación que está pensada para ser usada por otras aplicaciones, esta añade funcionalidad y es utilizada para tareas especificas.

Visión general del documento

Para el correcto desarrollo de este trabajo son necesarios una serie de requisitos. En la sección 3.1.2 se va a realizar una descripción de la funcionalidad y los requisitos de la herramienta a desarrollar, los casos de uso del sistema y las características de los usuarios que van a usar la herramienta. A continuación, en la sección 3.1.3 se hará una descripción más detallada de los requisitos que debe cumplir la herramienta desarrollada y de los requisitos de rendimiento.

3.1.2. Descripción general

Perspectiva del producto

La herramienta a desarrollar se va a implementar como un servicio web. Esta herramienta está desarrollada en Python, por lo que para hacerla funcionar se necesitará un sistema con Python instalado y que cumpla todas las restricciones de librerías externas.

Funciones del producto

La herramienta desarrollada debe ser capaz de detectar la presencia de determinadas categorías de información en un texto. Para poder utilizar el servicio web dentro de la red social Pesedia se necesita desarrollar un plug-in capaz de enviar peticiones desde Pesedia al servicio web y de recibir un mensaje del servicio web para que se muestre en Pesedia. Para lograrlo, se deben cumplir las siguientes funcionalidades:

[1] http://gti-ia.upv.es/

- Recibir peticiones HTTP de tipo POST con un body de tipo JSON (plug-in y servicio web).
- Procesado para simplificar el texto a clasificar (servicio web).
- Identificación de la categoría ubicación (servicio web).
- Identificación de la categoría salud (servicio web).
- Identificación de la categoría drogas/alcohol (servicio web).
- Identificación de la categoría emociones (servicio web).
- Identificación de la categoría ataques personales (servicio web).
- Identificación de la categoría detalles personales (servicio web).
- Identificación de la categoría neutro (servicio web).
- Enviar una petición HTTP de tipo POST con un body de tipo JSON (plug-in y servicio web).
- Mostrar el mensaje recibido para dar información al usuario sobre la información revelada (plug-in).
- Detectar que un usuario ha parado de escribir (plug-in).

Los casos de uso para la herramienta desarrollada son los siguientes:

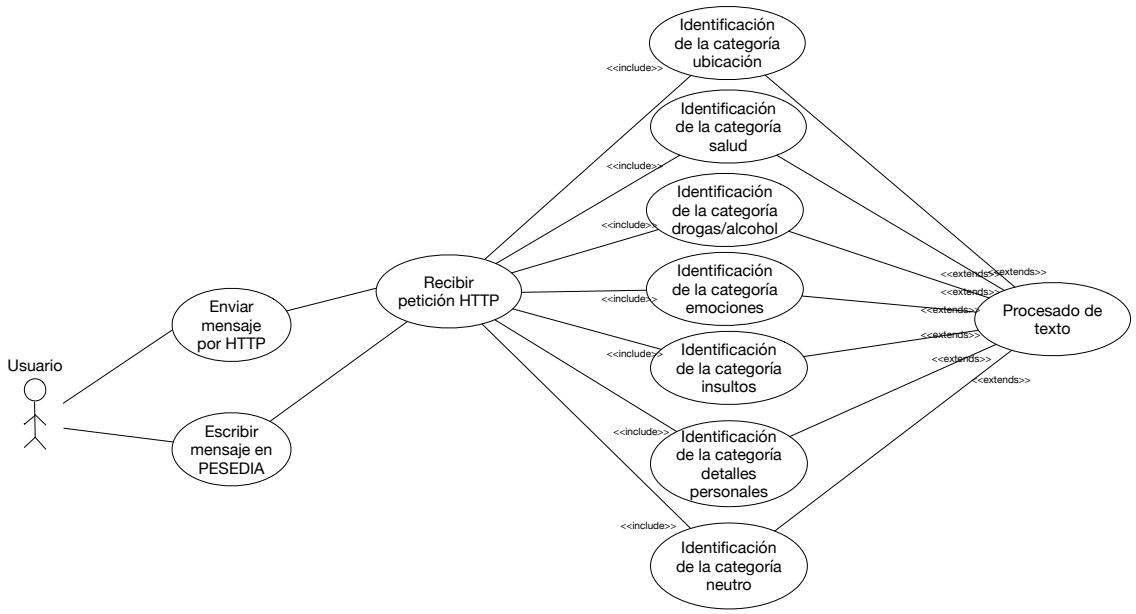


Figura 3.1: Casos de uso de la herramienta.

Características de los usuarios

Existen dos usuarios de la aplicación: un usuario de una red social, en nuestro caso Pesedia, y un usuario externo a la red social.

El usuario de la red social Pesedia (aunque también podría ser incluido en otras plataformas de redes sociales) tiene una cuenta dentro de la red social y podrá hacer uso de

la herramienta por medio de la interfaz gráfica de la red social cuando vaya a realizar una publicación.

El usuario externo también podrá hacer uso de la herramienta para saber si un texto contiene información asociada a determinadas categorías sensibles sin necesidad de estar dentro de la red social Pesedia. En este caso, no tendrá disponible una interfaz gráfica para interactuar con la herramienta.

Restricciones

La herramienta tiene varias dependencias debido a que se ha desarrollado en Python y se ha hecho uso de librerías externas. Para el correcto funcionamiento de la misma se deben de cumplir las siguientes dependencias:

- Python 3.5 o superior.
- Se deben tener instaladas las siguientes librerías:
  - **Senti-Py v1.0.0**: se trata de un analizador de sentimiento en español, esta se encuentra publicada en Github y puede ser utilizada de forma gratuita.
  - **SPACY v2.0.1**: es una librería de codigo abierto para procesamiento avanzado de lenguaje natural.
  - **PANDAS**: es una librería utilizada para la manipulación y análisis de datos en Python.
  - **UNIDECODE v1.0.**: librería utilizada para poder pasar texto en Unicode a ASCII.
  - **KERAS v2.2.4**: es una librería de redes neuronales,
  - **TENSORFLOW v1.13.1**: es una librería de codigo abierto para aprendizaje automático desarrollada por Google.
  - **NLTK v3.4**: es un conjunto de librerías para el procesamiento del lenguaje natural.

3.1.3. Requisitos específicos

Requisitos de rendimiento

La herramienta será utilizada en talleres de aprendizaje realizados en la UPV. Por cada sesión habrá aproximadamente un total de 150 usuarios, por lo que se espera que la herramienta consiga procesar unas 180 peticiones en un tiempo inferior a un minuto.

Requisitos funcionales

En las siguientes tablas (Tablas 3.1 a 3.6) se describe el comportamiento y funcionalidades de la herramienta software del proyecto cuando se cumplen ciertas condiciones.

Identificador	RF01
Nombre	Recepción de peticiones HTTP de tipo POST
Descripción	Recibe peticiones de tipo POST y extrae su contenido, el cual debe tener el formato JSON.
Entradas	Petición HTTP
Salidas	Texto

Tabla 3.1: Requisito funcional RF01

Identificador	RF02
Nombre	Procesado de texto
Descripción	Recibe un texto y elimina caracteres especiales, stopwords, urls...
Entradas	Texto
Salidas	Texto

Tabla 3.2: Requisito funcional RF02

Identificador	RF03
Nombre	Detección categoría ubicación.
Descripción	Recibe un texto e identifica si se revela información del tipo ubicación, en caso de ser así se pone a '1' en la posición correspondiente a esta categoría en el vector de categorías.
Entradas	Texto y un vector.
Salidas	Texto y un vector.

Tabla 3.3: Requisito funcional RF03

Identificador	RF04
Nombre	Detección categoría salud.
Descripción	Recibe un texto e identifica si se revela información del tipo salud, en caso de ser así se pone a '1' en la posición correspondiente a esta categoría en el vector de categorías.
Entradas	Texto y un vector.
Salidas	Texto y un vector.

Tabla 3.4: Requisito funcional RF04

Identificador	RF05
Nombre	Detección categoría drogas/alcohol.
Descripción	Recibe un texto e identifica si se revela información del tipo drogas/alcohol, en caso de ser así se pone a '1' en la posición correspondiente a esta categoría en el vector de categorías.
Entradas	Texto y un vector.
Salidas	Texto y un vector.

Tabla 3.5: Requisito funcional RF05

Identificador	RF06
Nombre	Detección categoría emociones.
Descripción	Recibe un texto e identifica si se revela información del tipo emociones, en caso de ser así se pone a '1' en la posición correspondiente a esta categoría en el vector de categorías.
Entradas	Texto y un vector.
Salidas	Texto y un vector.

Tabla 3.6: Requisito funcional RF06

Identificador	RF07
Nombre	Detección categoría ataques personales.
Descripción	Recibe un texto e identifica si se revela información del tipo ataques personales, en caso de ser así se pone a '1' en la posición correspondiente a esta categoría en el vector de categorías.
Entradas	Texto y un vector.
Salidas	Texto y un vector.

Tabla 3.7: Requisito funcional RF07

Identificador	RF08
Nombre	Detección categoría detalles personales.
Descripción	Recibe un texto e identifica si se revela información del tipo detalles personales, en caso de ser así se pone a '1' en la posición correspondiente a esta categoría en el vector de categorías.
Entradas	Texto y un vector.
Salidas	Texto y un vector.

Tabla 3.8: Requisito funcional RF08

Identificador	RF09
Nombre	Detección categoría neutro.
Descripción	Recibe un texto y un vector, en caso de que el vector no indique que se han identificado información sensible se marcará que el texto es neutro.
Entradas	Texto y un vector.
Salidas	Texto y un vector.

Tabla 3.9: Requisito funcional RF09

Identificador	RF10
Nombre	Enviar petición HTTP.
Descripción	Esto se hará desde la herramienta desarrollada y el plug-in, consiste en el envío de una petición HTTP en la que se envía información sobre el mensaje escrito por el usuario.
Entradas	Vector.
Salidas	Texto.

Tabla 3.10: Requisito funcional RF10

Identificador	RF11
Nombre	Mostrar el mensaje recibido para dar información al usuario sobre la información revelada.
Descripción	El plug-in mostrará el mensaje recibido por parte de la herramienta desarrollada, en la cual se le informa al usuario sobre la información que está revelando en el mensaje.
Entradas	Petición HTTP.
Salidas	Texto.

Tabla 3.11: Requisito funcional RF11

Identificador	RF12
Nombre	Detectar que un usuario ha parado de escribir.
Descripción	El plug-in esperará 2 segundos después de que el usuario haya parado de escribir, en caso de que no se reanude la escritura se considerará que el usuario ha parado de escribir.
Entradas	Texto.
Salidas	Texto.

Tabla 3.12: Requisito funcional RF12

3.2 Análisis del marco legal y ético

En este proyecto se va a trabajar con mensajes publicados en redes sociales. En concreto, se van a utilizar mensajes obtenidos de Twitter y de Pesedia.

Los mensajes de Twitter se utilizarán para la evaluación de la herramienta. Estos mensajes no revelan información sobre los usuarios que los publicaron. Debido a que no se ha guardado información que identifique a los usuarios sino solamente el texto que fue publicado, están anonimizados. Esto hace imposible, o prácticamente imposible, encontrar quién lo publicó.

Esta herramienta no almacenará ningún dato de los usuarios ni del contenido ni de la sensibilidad de los mensajes procesados. La herramienta solamente recibe una petición, la analiza y le muestra al usuario si ha detectado información sensible. Si en trabajos futuros quisiéramos guardar información para generar estadísticas, se informaría a los usuarios de esto, teniendo derecho a no aceptar las condiciones del servicio, o solicitar el borrado de sus mensajes.

Teniendo en cuenta los datos con los que se trabaja en el proyecto, podemos decir que se cumple la normativa RGPD [6].

CAPÍTULO 4
Diseño y desarrollo de la solución

4.1 Arquitectura del sistema

La arquitectura del sistema consta de la herramienta desarrollada y el plug-in, junto con la arquitectura de Pesedia [5]. La arquitectura del sistema desarrollado se muestra en la Figura 4.1.

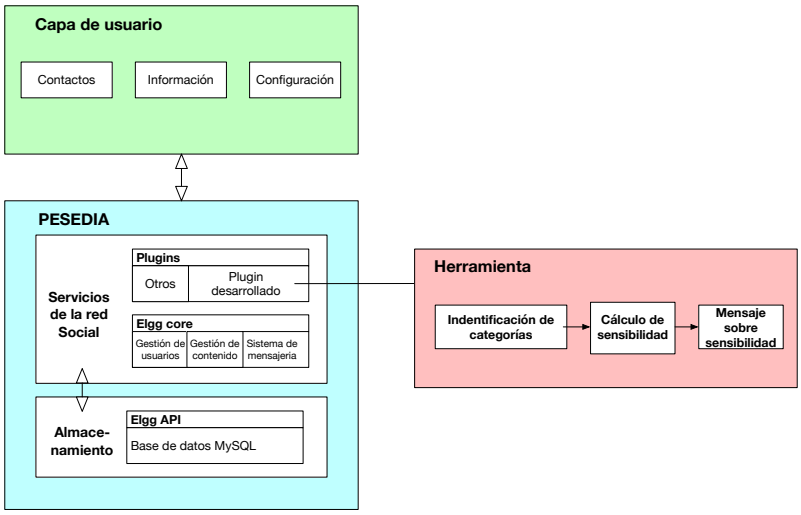


Figura 4.1: Visión general del sistema.

En la arquitectura se identifican tres módulos:

- **La capa de usuario** que permite la interacción entre usuario y la red social Pesedia.
- **La red social Pesedia**, formada por los servicios de red la red social y por la capa de persistencia.
- **La herramienta de detección de información sensible** desarrollada a lo largo de este proyecto.

La capa de usuario es la encargada de mostrar la información asociada a cada usuario, esta capa está dividida en tres apartados: contactos (amigos y relaciones del usuario), información (publicaciones, perfil, etc) y configuración de la cuenta.

La arquitectura de Pesedia está formada por dos componentes principales: los servicios de la red social y el sistema de almacenamiento de la misma, donde se proporciona

una capa de persistencia de toda la información generada en la red social. También incluye un apartado de plug-ins, donde se almacenan los plug-ins que se instalan en el sistema. En este lugar es donde quedará instalado el plug-in desarrollado en el proyecto que asegura una correcta comunicación entre Pesedia y la herramienta. Mediante este plug-in, cuando un usuario va a publicar información en Pesedia, la herramienta recibe información sobre esta publicación y detecta si se trata de información sensible, informando al usuario del nivel de sensibilidad del texto y de las categorías reveladas.

La herramienta desarrollada permite identificar, cuando se va a publicar un mensaje en Pesedia, si el usuario está revelando información sobre su ubicación, salud, drogas/alcohol, emociones, ataques personales y/o detalles personales. Para ello recibe un mensaje de texto con la información que el usuario va a publicar en la red social, y utiliza diversos métodos, cada uno de los cuales identifica si se está revelando información de alguna de las categorías mencionadas. En base a la revelación de estas categorías de información se realiza una estimación de la sensibilidad, y, a partir de esta estimación, se genera un mensaje que informa al usuario del nivel de sensibilidad de la información que está revelando. Este aviso puede llegar directamente al usuario en caso de llamar al servicio web desarrollado o a través de la interfaz de la red social Pesedia si está conectado a la red social. Este aviso llega al usuario antes de que se realice la publicación del mensaje, es decir, antes de que revele información sensible, lo que le permite decidir si continua y revela esta información o decide desistir o modificar el mensaje. En la Figura 4.2 se muestra la arquitectura de la herramienta desarrollada:

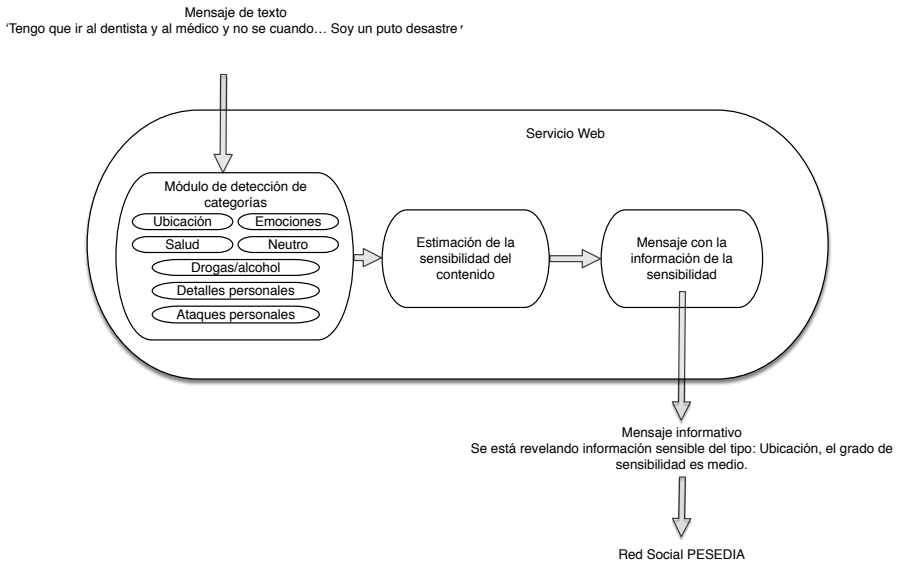


Figura 4.2: Diseño detallado de la herramienta.

El módulo de detección de categorías esta formado a su vez por otros módulos. Cada uno de estos módulos tiene la función de detectar una categoría, cada uno usando distintos métodos. El módulo encargado de detectar la Ubicación, analiza el texto haciendo uso de librerías encargadas de detectar localizaciones, como ciudades, montañas o ríos. El módulo de Salud, Drogas/alcohol y ataques personales hacen uso de diccionarios en los que se incluyen términos relacionados con la detección de estas categorías. El módulo de emociones usa librerías que analizan el texto encontrando emociones negativas o positivas. Por último el módulo de detalles personales hace uso de técnicas para encontrar información como tarjetas de crédito o relaciones de familia o amigos.

4.2 Tecnología utilizada

A continuación se describen las tecnologías y herramientas utilizadas para la realización del proyecto:

- **Python** es un lenguaje de programación interpretado y es multiparadigma. Soporta orientación a objetos, programación imperativa y pogramación funcional.
- **Sqlite** es un sistema de gestión de bases de datos relacional contenida en una pequeña biblioteca escrita en C.
- **JSON** es un formato de texto sencillo para el intercambio de datos.
- **Twitter API** es un conjunto de funciones que ofrece Twitter para poder acceder a los tweets publicados en la red social.
- **Ontología** es una definición formal de tipos, propiedades y relaciones entre entidades dentro de un dominio concreto.
- **JavaScript** es un lenguaje de programación interpretado orientado a objetos.
- **Apache 2** es un servidor web HTTP.
- **MySQL** es un sistema de gestión de bases de datos relacional.
- **Elgg** es una herramienta utilizada principalmente para el desarrollo de webs de redes sociales.
- **PyCharm**[1] es un entorno de desarrollo especifico para Python y está desarrollado por la empresa JetBrains.
- **DB Browser for SQLite**[2] es una interfaz gráfica para poder trabajar con bases de datos de SQLite. La interfaz permite de manera sencilla hacer consultas, recuperar datos y ver como están estructuradas las bases de datos.
- **Anaconda**[3] es una herramienta utilizada para crear entornos virtuales, sobre los cuales se puede trabajar de forma independiente. El uso de entornos virtuales permite aislar recursos como librerías y el entorno de ejecución del sistema principal o de otros entornos virtuales.
- **Protégé**[4] es un editor de ontologías con el cual se pueden hacer consultas de una forma muy parecida a las SQL. Con este editor se puede ver el contenido de las ontologías y como están estructuradas.
- **Jmeter**[5] es una herramienta en Java desarrollada por Apache y es utilizada para poder hacer pruebas de estrés sobre aplicaciones web.

[1] https://www.jetbrains.com/pycharm/
[2] https://sqlitebrowser.org
[3] https://www.anaconda.com/
[4] https://protege.stanford.edu/
[5] https://jmeter.apache.org/

4.3 Desarrollo de la solución propuesta

A continuación se va a explicar cómo se ha implementado la herramienta propuesta. Primero se explicará cómo se ha desarrollado el componente principal de la herramienta que es el módulo de extracción de categorías. A continuación, se explicará la encapsulación que se ha realizado del módulo en un servicio web. Finalmente, se explicará como se ha llevado a cabo la integración del servicio web en la red social Pesedia.

4.3.1. Módulo de extracción de categorías

El módulo de extracción de categorías es el responsable de analizar un texto y detectar la presencia de información de determinadas categorías. La elección de estas categorías se ha realizado en base a estudios y trabajos previos que destacan determinados tipos de información como sensibles [8]. Para ello, el módulo realiza un preprocesado del texto, extrae qué categorías están presentes y genera un mensaje para informar al usuario (ver Figura 4.3). A continuación se describen con más detalle cada uno de los pasos.

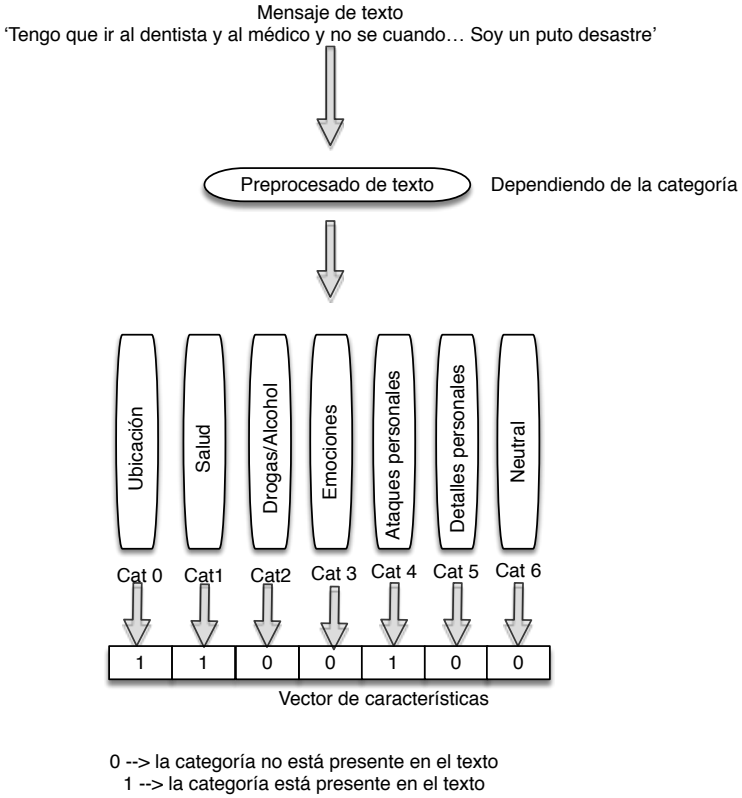


Figura 4.3: Módulo de extracción de categorías.

Preprocesado del texto

El preprocesado de texto consiste en simplificar los textos recibidos para facilitar el análisis del mismo. Este proceso se ha llevado a cabo haciendo uso de expresiones regulares, y consiste en eliminar del texto las siguientes cadenas:

- Caracteres especiales: "@", "#", "-", "_", "$"'.
- URLs: "HTTP" y "HTTPS".

- Emoticonos con formato Unidecode: U+1F603, U+1F629.
- Signos de puntuación: acentos, diéresis, puntos y comas.

El procesado de los textos no se hace de la misma manera en todas las categorías, debido a que en algunas nos interesa todo el contenido del texto sin filtrar. Por ejemplo para analizar si un texto pertenece a la categoría ”Detalles personales” nos interesan las direcciones de correo y por ello no eliminaríamos el carácter ”@”, mientras que en otras categorías es necesario dejar el texto lo más simple posible debido a que queremos extraer las palabras.

Extracción de categorías

Los textos recibidos por el módulo de extracción de categorías serán procesados para poder detectar la presencia o no de las siguientes categorías:

- Categoría 0: Ubicación
- Categoría 1: Salud
- Categoría 2: Drogas/Alcohol
- Categoría 3: Emociones
- Categoría 4: Ataques personales
- Categoría 5: Detalles personales
- Categoría 6: Neutro

Se ha realizado un análisis de las herramientas ya existentes relacionadas con la detección de estas categorías, pero uno de los principales problemas encontrados es que todas son en Inglés, por lo que no se han podido usar en la implementación de la herramienta. En los siguientes apartados se explicará con detalle el procedimiento seguido para la detección de cada categoría y los problemas encontrados al investigar otras herramientas relacionadas con esas categorías.

Ubicación

Se entiende que un texto pertenece a la categoría ubicación si revela información sobre localización, como por ejemplo ciudades, países o lugares. Para llevar a cabo esta función se ha utilizado la librería spaCy [6]. Esta librería se utiliza para procesar lenguaje natural, selecciona las palabras y les asigna una etiqueta. SpaCy esta pensada principalmente para ser usada con texto en inglés, debido a que detecta más tipos de información (ej. leyes, eventos o religiones), pero también se puede usar para texto en español. En concreto esta librería incluye un tipo de información que nos interesa para la ubicación, "LOC", en la cual se detectan lugares como montañas, ríos, ciudades y países entre otras cosas.

A continuación se muestra un ejemplo donde un mensaje de texto es procesado por la librería spaCy y como resultado se obtiene un listado con los distintos tipos de información que ha detectado:

[6] https://spacy.io

'RT @fpa: EN DIRECTO en #Periscope: Richard Ford (Letras) a su llegada a Oviedo. #PremiosPrincesadeAsturias':

- DIRECTO - MISC
- Richard Ford - PER
- Letras - MISC
- Oviedo - LOC

Concretamente podemos ver que en el ejemplo se ha detectado 'Richard Ford' como persona (PER), 'Oviedo' como una localización (LOC), y 'letras' y 'DIRECTO' como miscelánea (MISC). En nuestro caso nos interesan las entidades de tipo LOC.

Salud

Un texto pertenecerá a esta categoría si en él se detecta que se habla de alguna enfermedad. Inicialmente para detectar palabras relacionadas con la salud se utilizó un dataset [7] que recogía términos médicos, entre ellos, una gran cantidad de enfermedades y síntomas. El problema de este dataset, es que los términos recogidos eran demasiado técnicos, por lo que no funcionaban bien para detectar temas de salud en las redes sociales donde se utiliza un lenguaje más coloquial.

Al descartar la opción del dataset se planteó el uso de los datos procedentes del Instituto Nacional de Estadística (INE) [8]. Concretamente se utilizaron datos sobre los tipos de enfermedades de los que han habido más altas en España. Una vez obtenidos todos estos términos se identifica si un texto contiene o no estas palabras. A continuación se muestra un ejemplo de texto, en el que se detecta si contiene la categoría salud:

*'RT @tve_tve: El 44 % de los escolares entre 6-9 años tiene **sobrepeso** Atención a los consejos de @luciapediatra para evitarlo @SaberVivirTVE'*

Drogas/Alcohol

Los textos pertenecientes a estas categorías son aquellos que contengan términos relacionados con el alcohol o las drogas. En un primer momento, para obtener un conjunto de palabras con el cual poder trabajar e identificar si un texto pertenece o no a esta categoría, se utilizó una ontología [9] donde venían recogidos términos de drogas y bebidas alcohólicas. El problema que surgió fue similar al de la categoría salud. Esta ontología estaba en inglés y además utilizaba términos demasiado técnicos, por lo que finalmente se utilizo un documento realizado por el Gobierno de España en 2015 [19], en el que se recogen las drogas y bebidas más frecuentes en España.

Se puede ver un ejemplo de la detección de esta categoría en el siguiente texto:

*'Me acosté con un **gin tonic** y ya estoy bebiendo **vino**. Mira, que se acabe ya el finde'*

[7] https://www.kaggle.com/flaredown/flaredown-autoimmune-symptom-tracker
[8] http://www.ine.es/prodyser/espa_cifras/2018/20/
[9] http://knoesis.org/ontology/DAO.owl

Emociones

Un texto se considera que pertenece a la categoría "Emociones" si en él se expresan emociones positivas ó negativas. Nos interesa analizar cuál es el grado de emoción que aparece en los textos para identificar si son muy positivos o muy negativos. Para llevar a cabo esta tarea, se han evaluado dos librerías, la librería de Python Senti-Py [10], la cual se puede descargar en GitHub y trabaja con texto en español, y una librería desarrollada en el grupo GTI-IA de la UPV [3].

La librería Senti-Py hace un preprocesado del texto, para dejarlo lo más simplificado posible, y a continuación siguiendo ciertas técnicas devuelve un valor entre 0 y 1, siendo 0 muy negativo y 1 muy positivo. La librería del GTI-IA, devuelve 0 en caso de que el texto sea positivo o neutro y 1 si es un texto negativo.

Estas librerías tienen sus ventajas y desventajas. Por ejemplo, la librería Senti-Py al devolver un número entre 0 y 1, nos permite filtrar los textos según su sentimiento y considerar sensibles aquellos que superen cierto umbral. Sin embargo la otra librería aunque solamente identifica textos negativos tiene una gran velocidad de procesado.

Se justificará la elección de la librería más adecuada para la herramienta en el apartado de resultados y pruebas.

Ataques personales

Los ataques personales se definen como críticas o ataques directos a personas o grupos. Hemos interpretado que está categoría recoge principalmente los insultos, por lo que un texto pertenecerá a esta categoría si se encuentran insultos en el. Para hacer esto se ha utilizado una base de datos online llamada Hatebase.org en la cual se recogen insultos y faltas de respeto en multitud de idiomas, entre ellos el Español. De esta manera se han recogido todas esta palabras en un fichero, y continuación se ha buscado la presencia de estas palabras en los textos.

Un ejemplo de un texto en el que se ha detectado la presencia de esta categoría es el siguiente:

*'RT @TuiterHits: Como se nota que ser **gilipollas** es gratis.'*

Detalles personales

Un texto pertenece a esta categoría si contiene información como el número de teléfono, código postal, etc, o detalles de relaciones como novio, madre, prima, etc. Debido a que no se ha encontrado ninguna herramienta relacionada con esta función, se ha decidido desarrollar dos funciones en Python. La primera hace uso de expresiones regulares e identifica cualquier tipo de información relacionada con tarjetas de crédito, código postal, número de teléfono, email, etc. La segunda comprueba si hay palabras relacionadas con la familia o personas cercanas y comprueba qué tipos de emoticonos relacionados con las relaciones personales se usan en el texto.

Un ejemplo de la presencia de esta categoría sería la siguiente, donde se revelan información sobre familia o personas cercanas:

*'RT @_MaluOficial_: Como el vino **amigo**... Cada año mejor... Te quiero @MelendiOficial'*

[10] https://github.com/aylliote/senti-py

Información Neutral

Un texto es considerado como neutro si no contiene información de ninguna de las categorías anteriores. Esta categoría se activa en el caso de que ninguna de las anteriores este activada.

Un ejemplo de un texto neutro es el siguiente:

'@RafaLagoon Sabes que esto no lo decides tú, verdad?'

4.3.2. Servicio web para la estimación de la sensibilidad de la información

Medida para la estimación de la sensibilidad

Según Acquisiti et al. [2], no existe un método concreto para establecer un valor para la privacidad y los datos sensibles. La evaluación de la sensibilidad de los datos personales no se refiere a la evaluación de tipos de información individuales, sino a la evaluación de combinaciones de datos personales. Considerando el enfoque que se adopta en trabajos que analizan el valor económico que se asigna a los datos [29], proponemos utilizar una aproximación similar donde la sensibilidad de un mensaje consiste en el valor acumulado de los valores de sensibilidad de las categorías que aparecen en las publicaciones.

Para asignar un valor de sensibilidad a las distintas categorías propuestas se ha utilizado como base el artículo [24] donde se proponen ciertos valores según el tipo de información que se este revelando (i.e., clase social, dirección, edad o raza). En este artículo, para dar el valor de sensibilidad a cada categoría, se pasó a ciertos usuarios una encuesta en la que debían decir según ellos cómo de importante es esta información. En la herramienta propuesta en el TFG hemos considerado siete categorías y se ha hecho uso de los resultados publicados en el artículo [24]. En este artículo, se le preguntó a los usuarios cómo de privada consideraban cierta información, por ejemplo, su localización, detalles médicos, fotos sobre ellos, etc, haciendo uso de los resultados de esas encuestas, se ha asignado un valor promedio a cada categoría. Se pueden observar estos valores en la Figura 4.4.

Para obtener el valor total de la sensibilidad en un texto se sumarán los grados de sensibilidad, es decir, si se detectan dos categorías sumaremos los grados que se le han asignado a cada una y según el resultado obtenido de la suma, diremos que se ha obtenido un grado nulo, bajo, medio o alto, los niveles que se considerarán son los siguientes:

- NULO, = 0
- BAJO, >0 y <=4
- MEDIO, >4 y <=6
- ALTO, >6

La herramienta devolverá un mensaje donde se informará a los usuarios sobre las categorías que están revelando y el grado de sensibilidad.

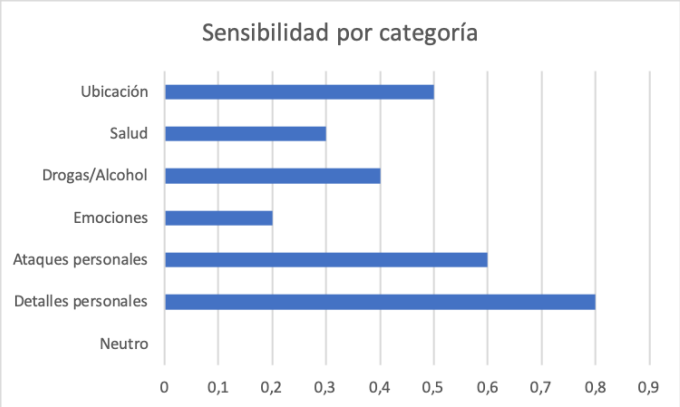


Figura 4.4: Propuesta valores de sensibilidad.

Servicio Web

Para poder acceder a la herramienta a través de Internet y poder utilizarla en la red social Pesedia, se ha desarrollado un servicio web[11] en Python haciendo uso de la librería http.server que viene incorporada en Python.

Para configurar el servicio web, se ha tenido que estudiar el funcionamiento de http.server, debido a que hay que configurar las cabeceras de las peticiones http, y saber cuáles hay que mandar para que la comunicación entre Pesedia y la herramienta se complete correctamente.

El servicio web se ha creado de forma que atienda todas las peticiones en la dirección 0.0.0.0 y el puerto 80, pero solamente procesará aquellas que sean del tipo POST. Una vez llegue una petición, la herramienta procesará el mensaje JSON recibido, extraerá el contenido y en caso de que éste encuentre alguna categoría activa devolverá un mensaje al usuario. En caso contrario devolverá un mensaje vacío.

4.4 Integración en la red social Pesedia

Pesedia es una red social desarrollada por el grupo de investigación GTI-IA de la Universidad Politécnica de Valencia. Pesedia está desarrollada en Elgg, una tecnología usada principalmente para la creación de redes sociales.

Para poder utilizar el servicio web desarrollado desde Pesedia, se ha tenido que implementar un plug-in de Elgg. Los plug-ins de Elgg tienen que seguir cierta estructura. En la raíz de un plug-in deben de aparecer los siguientes archivos:

- **start.php**. En este archivo van todas las llamadas de inicio del plug-in y habilitará la funcionalidad implementada
- **manifest.xml**. En este archivo se debe incluir una descripción del plug-in con los siguientes elementos: id, name, author, version, description y requires.

Los directorios que forman el plug-in son los siguientes:

- **/actions**, lugar donde se definen las acciones que cambien la base de datos.

[11] https://github.com/vbotti/twitterDatos_TFG/blob/master/webService.py

- **/classes**, lugar donde se deben de definir las clases que posteriormente serán reconocidas por Elgg.
- **/languages**, lugar donde se guardan diccionarios de palabras para los idiomas que los desarrolladores quieran soportar.
- **/vendor**, lugar donde se incluyen librerías de terceros
- **/views/default**, lugar donde se guarda todo el código relacionado con la generación de vistas de la red social.

Para la creación del plug-in hará falta utilizar el directorio /views/default, ya que el plug-in a desarrollar modifica la vista de la red social.

Para desarrollar el plug-in se ha hecho uso de JavaScript. El plug-in detecta la escritura de un usuario en un campo de texto y cuando el usuario deja de escribir durante dos segundos envía un mensaje en formato JSON al servicio web con la estructura 'text':'mensaje'. El mensaje se envía mediante una petición POST haciendo uso de la librería de JavaScript XMLHttpRequest. Una vez se recibe la respuesta de la herramienta, en caso de que se haya detectado alguna de las categorías, se le muestra al usuario un mensaje de advertencia. Se puede ver un ejemplo del funcionamiento en las Figuras 4.5, 4.6, y 4.7.

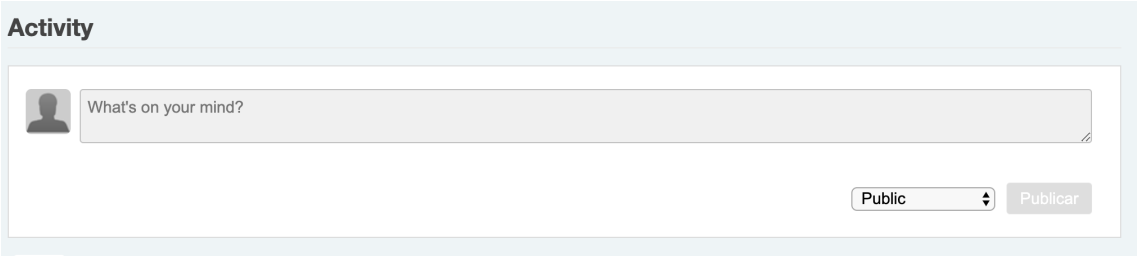


Figura 4.5: Pesedia antes de escribir un mensaje.

En la Figura 4.5 se puede ver la interfaz de Pesedia a la hora de escribir un mensaje. La interfaz está compuesta de un campo para escribir, un desplegable para elegir si queremos que el mensaje sea público o privado, y el botón de publicar, el cual se mantiene desactivado hasta que se recibe una respuesta de la herramienta. Cuando el usuario escribe un mensaje, el botón de publicar cambiará su texto, indicando al usuario que se esta procesando el mensaje (ver Figura 4.6).

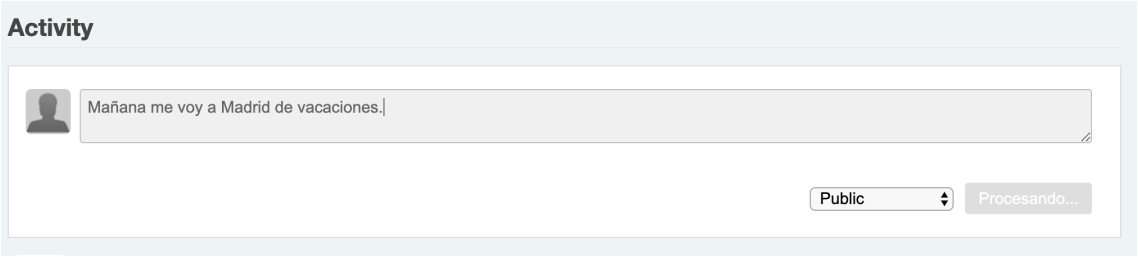


Figura 4.6: Pesedia durante el procesamiento del mensaje.

Una vez se recibe la respuesta de la herramienta se le mostrará al usuario qué categorías se están revelando, y el grado de sensibilidad que se ha encontrado en el texto. Podemos ver un ejemplo de como vería un usuario este mensaje en Pesedia en la Figura 4.7.

Figura 4.7: Mensaje de advertencia en Pesedia.

CAPÍTULO 5
Implantación

Para realizar la implantación del proyecto se deberán seguir los siguientes pasos. Se harán dos instalaciones diferentes. Primero se describirá la instalación del servicio web y sus respectivas dependencias. Después se describirá una instalación de Elgg y la red social PESEDIA con sus respectivos plug-ins, entre ellos el que se ha desarrollado en el proyecto. Todo esto se va a hacer sobre una máquina con Ubuntu 16.04 instalado, por lo que todos los pasos que se describen a continuación serán para ésta, aunque probablemente también se puedan hacer de forma parecida en otras instalaciones Linux.

5.1 Implantación de la herramienta

Lo primero de todo es comprobar que tenemos una versión de Python igual o superior a la 3.5. Para ello, utilizando la linea de comandos, introduciremos la siguiente línea:

```
$ python3 --version
```

En caso de que el comando no funcione, o la versión sea inferior a la 3.5, deberemos instalar una versión más reciente. En ese caso habrá que introducir el siguiente comando:

```
$ install python3.7
```

Una vez se tiene Python actualizado, se descargará la herramienta de Github a través del siguiente enlace https://github.com/vbotti/twitterDatos_TFG. Una vez descargada se debe de descomprimir y dejar la carpeta donde mejor nos parezca. En caso de estar en un ordenador personal, se recomienda el uso de entornos virtuales. Para el desarrollo de la herramienta se utilizó Anaconda. Si no usamos esta opción solamente necesitaremos instalar PIP3. PIP3 es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes que estén escritos en Python.

Para su correcta instalación primero actualizaremos las librerias disponibles utilizando:

```
$ sudo apt-get update
```

Una vez actualizado ya se puede hacer la instalación de PIP3. Para ello utilizaremos el siguiente comando:

```
$ sudo apt-get -y install python3-pip
```

Una vez descargado PIP3, podemos comprobar su correcto funcionamiento haciendo uso de los siguientes comandos:

Verificar que la herramienta PIP3 funciona:

```
$ pip3 --help
```

Comprobar la versión de PIP3 instalada:

```
$ pip3 --version
```

Una vez tenemos PIP3 instalado, ya podemos comenzar a instalar las librerías necesarias para el funcionamiento de la herramienta.

Spacy:

```
$ pip3 install -U spacy
$ python -m spacy downlolad es
```

Pandas:

```
$ pip3 install pandas
```

Unidecode:

```
$ pip3 install Unidecode
```

Senti-py:

```
$ pip3 install spanish_sentiment_analysis
```

Keras:

```
$ pip3 install keras
```

TensorFlow:

```
$ pip3 install tensorflow
```

nltk y ntlk.corpus:

```
$ pip3 install nltk
$ python3
>>> import nltk
>>> nltk.download()
>>> exit()
```

Una vez instaladas todas estas librerías, la herramienta debería funcionar correctamente. Para lanzar a ejecución la herramienta debemos acceder por línea de comandos a la carpeta de la herramienta y ejecutar la siguiente línea:

```
$ python3 webservice.py
```

En el caso de que todo haya funcionado correctamente, ahora habrá un servicio web escuchando a las peticiones que lleguen por la dirección http://IP_ORDENADOR:80/

5.2 Implantación de Pesedia

Para poder instalar una instancia de la red social Pesedia necesitamos cumplir primero ciertos requisitos para conseguir un correcto funcionamiento, estos requisitos son:

- Apache Server
- MySQL 5.7+

- PHP 7+ con las extensiones: gd, json, mysql, imap, ldap, odnbc, xml, xmlrpc, curl, mbstring
- PhpMyAdmin

Una vez sabido esto, lo primero de todo es obtener la última versión del sistema operativo Ubuntu. Para ello se debe utilizar el siguiente comando:

```
$ sudo apt update -y && sudo apt upgrade -y
```

Esta actualización podría tardar varios minutos dependiendo de cómo de actualizado tengamos el sistema. A continuación se deben de instalar los requisitos que se han mencionado anteriormente.

Apache server:

```
$ sudo apt-get install -y apache2
```

MySQL 5.7+:

```
$ sudo apt-get install -y mysql-client
$ sudo apt-get install -y mysql-server
```

PHP 7+ con las extensiones necesarias:

```
$ sudo apt-get install -y php php-mysql php-gd php-imap php-ldap php-odbc php-
    xml php-xmlrpc php-cirl php-mbstring
```

PhpMyAdmin:

```
$ sudo apt-get install -y phpmyadmin
```

Una vez tenemos todas estas aplicaciones instaladas, debemos de instalar Elgg. Lo primero que debemos hacer es descargarlo:

```
$ swget https://elgg.org/about/getelgg?forward=elgg-2.3.10.zip
```

Una vez se haya descargado lo descomprimimos:

```
$ unzip elgg-2.3.10.zip
```

Movemos a la raíz /var/www. Para lograrlo haremos lo siguiente:

```
$ mv elgg-2.3.10 /var/www/COPIA_PESEDIA/public_html
```

Para el correcto funcionamiento de Elgg, la estructura de los directorios en la ruta /var/www/COPIA_PESEDIA/ debe ser la siguiente:

- elggdata/
- public_html/
- sitebackups/

En el caso de que nos falte alguna de las carpetas se pueden crear haciendo uso del comando mkdir.

Una vez tengamos todas las carpetas creadas, debemos asignar el permiso 750 y el usuario www-data a todos los directorios y ficheros para que puedan ser utilizados por Apache:

```
$ chmod -R 750 /var/www/COPIA_PESEDIA/
$ chown -R www-data:www-data /var/www/COPIA_PESEDIA/
```

Ahora debemos de crear una base de datos para la instancia de Pesedia. Este paso se puede hacer de dos formas, mediante la línea de comandos o mediante phpMyAdmin. A continuación se explicarán las dos formas.

Crear base de datos mediante linea de comandos

Para crear una base de datos utilizando la línea de comandos debemos de ejecutar mysql:

```
$ mysql -u root -p
```

A continuación crearemos la base de datos:

```
mysql> CREATE DATABASE BD_PESEDIA;
```

Crear base de datos mediante phpMyAdmin

Para poder usar phpMyAdmin, primero debemos de lanzar el servidor Apache:

```
$ sudo service apache2 restart
```

Una vez se haya lanzado correctamente el servidor, deberemos de acceder a la dirección http://localhost/phpmyadmin/. Iniciamos sesión y una vez dentro hacemos lo siguiente (ver Figura 5.1):

- Seleccionamos "nueva" en el panel de la izquierda para poder acceder a la creación de la BD.
- Le ponemos un nombre a la BD, en este caso se le ha puesto "BD_PESEDIA" y seleccionamos la opción "utf8_general_ci".
- Una vez hecho lo anterior le damos a crear.

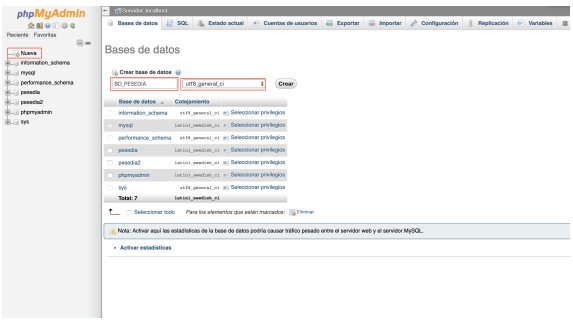


Figura 5.1: Creación de BD mediante phpMyAdmin

Una vez creada la base de datos, se debe de configurar Apache para poder acceder al sitio web. Para ello, se debe de crear un fichero de configuración llamado *copia.pesedia.conf* haciendo uso del siguiente comando:

```
$ vim /etc/apache2/sites-available/copia.pesedia.conf
```

Este comando lo que se hace es crear y escribir en el fichero, ahora debemos añadir las siguientes lineas como contenido del fichero:

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/PESEDIA_DIR/public_html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        <Directory /var/www/PESEDIA_DIR/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow, deny
                allow from all
                Require all granted
        </Directory>
</VirtualHost>
```

A continuación debemos habilitar esta configuración y otros módulos (PHP y Rewrite) para asegurar un correcto funcionamiento de la web. Para ello se deben utilizar los siguientes comandos:

```
$ sudo a2ensite copia.pesedia.conf
$ sudo a2enmod php && sudo a2enmod rewrite
$ sudo service apache2 restart
```

Si todo se ha hecho correctamente, podríamos acceder a la dirección `http://localhost/` y debería aparecer la imagen que se muestra en la Figura 5.2.

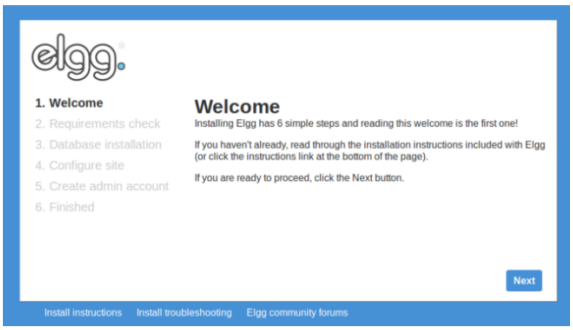


Figura 5.2: Pantalla de bienvenida Elgg.

A continuación se deben completar los campos que se indican en las siguientes pantallas, que aparecen numeradas en las Figuras 5.3 a la 5.7.

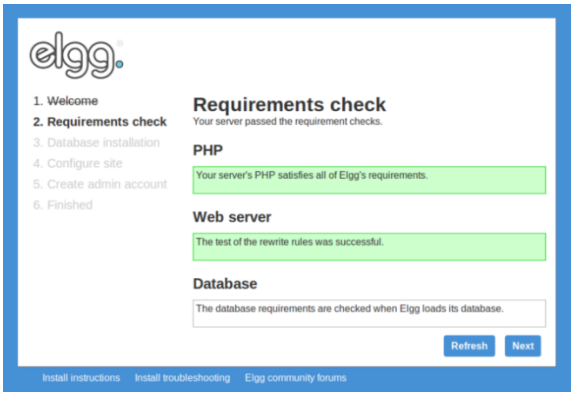


Figura 5.3: Comprobación de requisitos de Elgg.

En la pantalla Requirements check (ver Figura 5.3) Elgg comprueba que todos los requisitos necesarios estén instalados. En el caso que estén todos los requisitos habrá que pulsar 'next' para continuar con la instalación.

Figura 5.4: Instalación de la BD de Elgg.

Seguidamente (ver Figura 5.4), Elgg pide que se añada la información de la base de datos para poder hacer la instalación. Para ello le indicaremos el usuario y la contraseña de la BD que hemos creado anteriormente, el nombre de la BD y la dirección donde se encuentra el servidor MySQL.

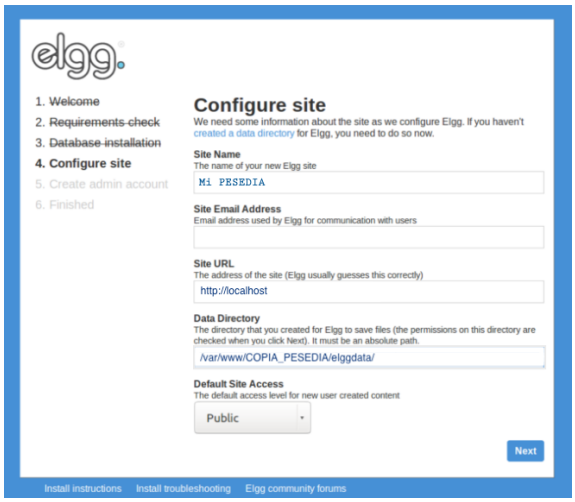


Figura 5.5: Configuración del sitio Elgg.

Si la instalación de la base de datos ha terminado correctamente, el siguiente paso es configurar el sitio web, indicando el nombre que queremos darle al sitio, la url y la ruta donde se encuentra la carpeta de Pesedia (ver Figura 5.5).

Ahora deberemos crear la cuenta de Administrador del sitio web. Se debe indicar el nombre que aparecerá en pantalla, un email, usuario y contraseña (ver Figura 5.6).

En caso de que todos los pasos se hayan realizado correctamente y no hayan surgido problemas, llegaremos a una pantalla como la de la Figura 5.7.

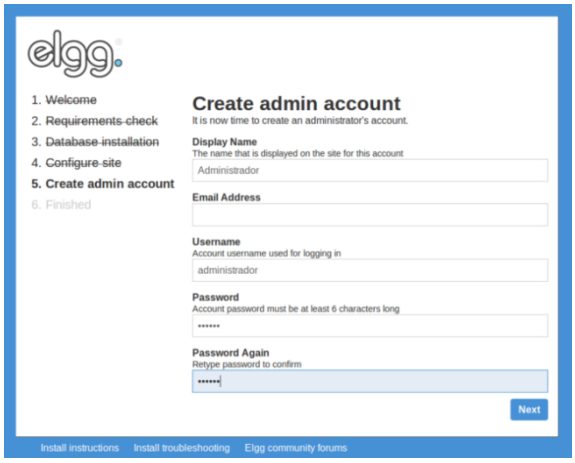


Figura 5.6: Crear cuenta de administrador en Elgg.

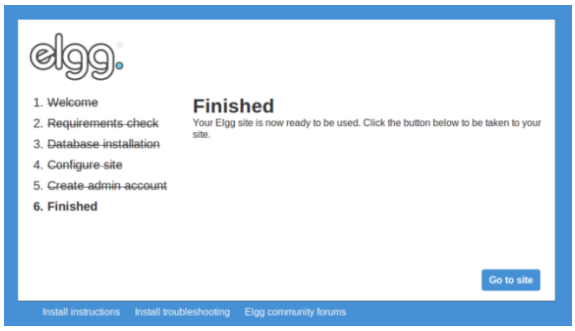


Figura 5.7: Completada la instalación de Elgg.

Una vez terminada la instalación ya podremos acceder al sitio. Ahora ya es posible instalar el plug-in desarrollado. Para ello se debe ir al apartado de Administración y una vez aquí ya podemos ir al apartado plug-ins.

Para poder activar los plug-ins debemos moverlos a una carpeta en concreto. Se puede usar el mismo comando para todos los plug-ins que se quieran instalar:

```
1 $ sudo mv plug-in /var/www/COPIA_PESEDIA/public_html/mod
```

CAPÍTULO 6
Pruebas

En este capítulo se van a realizar varios tipos de pruebas. Primero se evaluará el etiquetado del dataset utilizado para comprobar el nivel de acuerdo entre los etiquetadores. Lo siguiente será evaluar la herramienta desarrollada haciendo uso de este dataset. A continuación se medirán los tiempos de ejecución utilizados para detectar cada categoría. En el caso de la categoría de emociones, se analizarán los tiempos según la librería de emociones utilizada (Senti-Py o librería del GTI-IA). Después, haciendo uso de la herramienta JMeter, se harán pruebas de carga del servicio web para comprobar cuál es el número máximo de peticiones que la herramienta puede atender simultáneamente y los tiempos de respuesta del servicio. Finalmente, se han realizado pruebas para comprobar el funcionamiento de la medida de la sensibilidad propuesta.

6.1 Evaluación del etiquetado del dataset

Este dataset, elaborado por el grupo GTI-IA, ha sido generado recientemente. Antes de ser utilizado se debe de analizar la calidad del etiquetado, es decir, el nivel de acuerdo de los etiquetadores. Si el etiquetado no es bueno, puede llevar a confusión en el momento de probar la herramienta.

El dataset usado está compuesto por 3707 tweets. Estos tweets fueron etiquetados por cuatro etiquetadores en las categorías que se muestran en la Figura 6.1. Cada categoría se representa con un número del 0 al 9, donde la categoría de 'Ubicación' se corresponde con 0 y 'Información Neutral/Objetiva' con 9.

Un ejemplo de como está etiquetado un tweet es el siguiente:

"RT @Rafael_Vidac: Si tu pasión va en sentido contrario al de tus pasos, no dudes en seguirla. https://t.co/QCURA0LIiW"

La lista de etiquetas resultante sería ['9', '3', '3', ' '] que contiene uno o varios valores de categorías por cada etiquetador. En este caso el etiquetador 1 ha considerado que el mensaje pertenecía a la categoría 9. El etiquetador 2 y 3 han clasificado el texto como de la categoría 3 y el etiquetador 4 ha considerado que era neutro. Con este ejemplo podemos ver que cada etiquetador ha tenido sus propios criterios a la hora evaluar a qué categoría pertenece un tweet.

Para evaluar la calidad del etiquetado se ha utilizado el nivel de acuerdo que ha habido entre los etiquetadores para cada una de las categorías. Existen varios métodos para calcular el nivel de acuerdo, pero debido a las condiciones en las que se ha hecho el etiquetado (un etiquetador puede seleccionar más de una categoría) y que el número de

Ubicación	El tweet revela información de una ubicación.
Salud	El tweet contiene información médica/salud personal.
Drogas/Alcohol	El tweet contiene información personal sobre el uso de alcohol/drogas o revela información bajo su influencia.
Emociones	El tweet revela emociones claras de alegra, frustración, enfado, etc.
Ataques Personales	El tweet es una crítica o ataque directo a una persona, grupo, asociación, etc.
Estereotipar	El tweet contiene referencias estereotípicas étnicas, raciales, etc. sobre un grupo o persona.
Detalles de Relación	El tweet revela detalles de relación o vinculos entre personas o asociaciones (ej. pareja, suegro, empleado, miembro, etc.).
Detalles Personales	El tweet contiene detalles personales (ej. estado civil, orientación sexual, idología, creencias, formación, trabajo/ocupación, datos economicos, contenido embarazoso o inapropiado, etc.)
Datos Personales Identificables	El tweet contiene información identificable (ej. DNI, número de la seguridad social, número de tarjeta de crédito, domicilio, dirección de correo,fecha de nacimiento, número de teléfono, etc.)
Información Neutral/Objetiva	El tweet es neutro u objetivo que no revelan información confidencial o privada.

Figura 6.1: Categorías de etiquetado del dataset

tweets por categorías es muy desbalanceado, (ver Tabla 6.1) se ha utilizado el coeficiente PABAK [7], el cual esta basado en el coeficiente Fleiss Kappa.

Categoría	# tweets de la categoría
Categoría 0	491
Categoría 1	271
Categoría 2	108
Categoría 3	2405
Categoría 4	705
Categoría 5	464
Categoría 6	783
Categoría 7	1306
Categoría 8	107
Categoría 9	2726

Tabla 6.1: Coeficiente PABAK para cada categoría.

Este coeficiente PABAK se utiliza para comprobar el nivel de acuerdo de los etiquetadores, para calcularlo se debe de usar la siguiente fórmula 6.1

$$PABAK = 2 * \bar{P} - 1 \tag{6.1}$$

Como se ha dicho anteriormente, este coeficiente se calcula a partir del coeficiente Fleiss Kappa, exactamente, se puede ver en la ecuación anterior que aparece una variable $\bar{P}$ (es la media de los etiquetadores que están de acuerdo en cada categoría, posteriormente se define en la ecuación 6.4), esta es una de las variables usadas para el cálculo del coeficiente, para explicar que es esta variable $\bar{P}$, se va a explicar como se calcula Fleiss Kappa [10, 23].

Suponer que tenemos un número 'N' de tweets, 'n' es el número de etiquetadores y 'k' el número de categorías. Los tweets están indexados desde $i = 1, ...N$ y las categorías desde $j = 1, ...k$. El parámetro n_{ij} representa el número de etiquetadores que han asignado el tweet 'i' a la categoría 'j'.

Primero se debe de calcular p_j, la proporción de etiquetaciones que ha habido en la categoría j según la ecuación 6.2.

$$p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij} \tag{6.2}$$

A continuación se calcula P_i que representa qué etiquetadores están de acuerdo en la categoría de cada tweet i (ver ecuación 6.3).

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1) \tag{6.3}$$

Por último calculamos $\bar{P}e$ y $\bar{P}$ que es la variable que necesitamos para calcular PABAK. Estas dos se utilizarán para el cálculo de k (ver ecuación 6.4 y 6.5).

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i \tag{6.4}$$

$$\bar{P}e = \sum_{j=1}^{k} P_j^2 \tag{6.5}$$

Ahora ya podriamos calcular el coeficiente Fleiss Kappa en caso de ser necesario(ver ecuación 6.6).

$$k = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{6.6}$$

El coeficiente PABAK se ha usado sobre cada categoría utilizando la fórmula 6.1. De esta manera obtenemos el nivel de acuerdo entre los etiquetadores por categoría. Los resultados se pueden ver en la Tabla 6.2.

Categoría	PABAK
Categoría 0	0,86
Categoría 1	0,92
Categoría 2	0,97
Categoría 3	0,40
Categoría 4	0,81
Categoría 5	0,87
Categoría 6	0,78
Categoría 7	0,63
Categoría 8	0,97
Categoría 9	0,26

Tabla 6.2: Coeficiente PABAK para cada categoría.

Según el artículo [7], donde se propone el uso del coeficiente PABAK, se indica que este coeficiente puede tener valores desde -1 a 1, siendo 0 un 50 % de acuerdo entre los anotadores. Con esto podemos ver que el nivel de acuerdo entre los cuatro anotadores ha sido bastante alto prácticamente en todas las categorías. Solamente hay dos valores que son más bajos que el resto, las categorías 3 y 9, se puede ver que esto tiene relación con el desequilibrio del dataset, ya que estás categorías son las que tienen más tweets, superando los 2400. Aún así podemos llegar a la conclusión de que el etiquetado realizado sobre el dataset es bueno, por lo tanto podremos utilizarlo para evaluar la herramienta.

6.2 Evaluación de la herramienta

Para realizar la evaluación de la herramienta se ha utilizado el dataset evaluado previamente. Estos tweets son procesados por el módulo el cual generará por cada tweet un vector de dimensión igual al número de categorías. Cada posición del vector indica mediante el uso de 1 y 0 si las categorías están presentes o no respectivamente. Este vector se compara con el vector de categorías generado en base a la información que proporcionaron los etiquetadores del dataset (ver Figura 6.2). Debido a que los etiquetadores clasificaron los tweets en 9 categorías, y nuestra herramienta categoriza en 7, se ha transformado el vector de 9 a 7, juntando las categorías detalles de relación, detalles personales y detalles personales identificables, y eliminando la categoría estereotipar. Para evaluar la herramienta, además se utilizarán las medidas de precisión, exhaustividad (recall), acierto y F1.

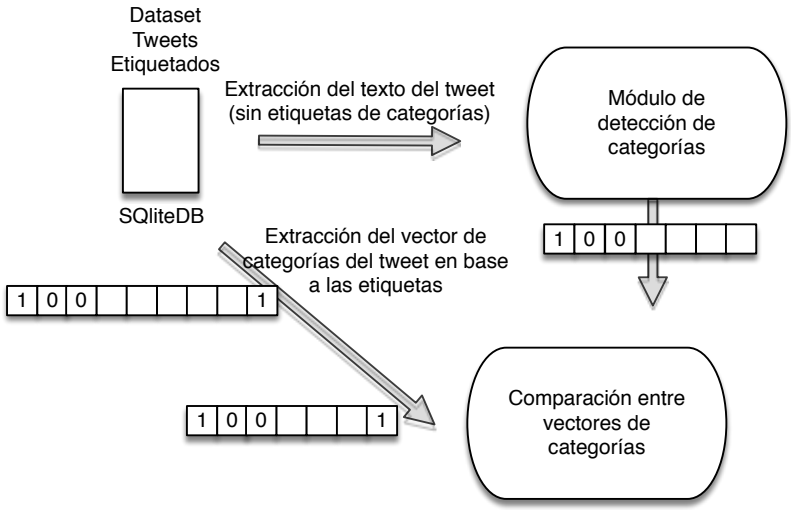


Figura 6.2: Evaluación del módulo de detección de categorías.

Precisión y exhaustividad (recall) son métricas empleadas en la medida del rendimiento de los sistemas de búsqueda y recuperación de información y reconocimiento de patrones.

Precisión (Falsos positivos) intenta responder a la pregunta ¿qué proporción de identificaciones positivas fue correcta? Un modelo que no produce falsos positivos tiene una precisión de 1.0. La precisión se calcula siguiendo la fórmula 6.7,

$$Precision = \frac{VP}{VP + FP} \tag{6.7}$$

donde VP representa a los verdaderos positivos y FP representa a los falsos positivos.

En el ejemplo de la Figura 6.3 se ilustra el cálculo de estas medidas con un ejemplo. Se presenta un dataset de tweets que contiene 3200 tweets etiquetados. Dentro de este dataset hay tweets que pertenecen a distintas categorías (A, B y C). Concretamente, de la categoría A hay etiquetados 400 tweets. Para calcular las medidas de precisión y recall del módulo de detección de categorías se pasan como entrada 3200 tweets. El módulo detecta que 700 tweets pertenecen a la categoría A. Comprobando si realmente los tweets que se detectaron como pertenecientes a la categoría A realmente pertenecen a esa categoría nos damos cuenta que sólo 300 pertenecían realmente a A (VP). Si de los 700 tweets clasificados como A sólo 300 fueron realmente de categoría A, entonces 400 fueron falsos positivos (FP). En base a esto y a la fórmula de precisión podemos decir que el módulo

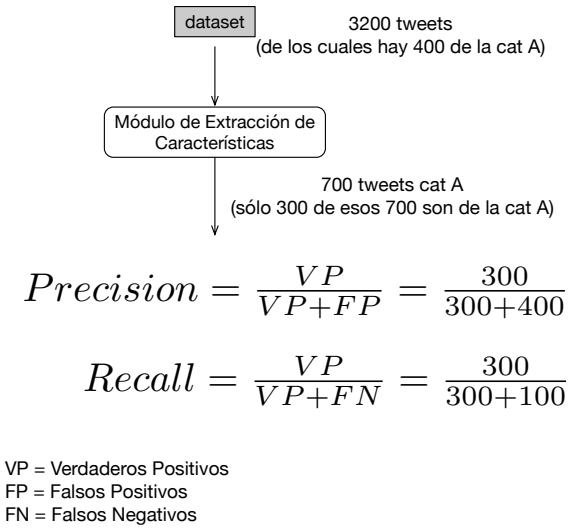


Figura 6.3: Ejemplo de cálculo de las medidas precisión y recall.

tiene una precisión del 42 % para esa categoría, es decir que cuando el módulo de extracción de características predice que un mensaje es de la clase A, acierta el 42 % de las veces.

Exhaustividad (Falsos negativos) intenta responder a la pregunta ¿qué proporción de positivos reales se identificaron correctamente? Un modelo que no produce falsos negativos tiene un recall de 1.0. El recall se calcula siguiendo la fórmula 6.8,

$$Recall = \frac{VP}{VP + FN} \tag{6.8}$$

donde VP representa a los verdaderos positivos y FN representa a los falsos negativos. En el ejemplo de la Figura 6.3, nos damos cuenta que el módulo de detección de categorías se dejó por detectar algunos de los tweets que eran de la clase A. Concretamente en el dataset había 400 tweets de la clase A y el módulo sólo llegó a detectar correctamente (VP) 300 tweets. Por lo tanto hubo 100 tweets que el módulo no detectó como pertenecientes a la clase A. En base a la fórmula del recall podemos decir que el recall del módulo de detección de categorías es del 75 %, es decir, identifica correctamente el 75 % de los mensajes de la clase A.

El acierto o accuracy, se trata del porcentaje de textos que nuestra herramienta clasifica correctamente, es decir, del número total de textos que fueron clasificados en una categoría, cuantos acierta la herramienta desarrollada. La fórmula para su cálculo es la siguiente 6.9:

$$Accuracy = \frac{VP + VN}{FP + FN + VP + VN} \tag{6.9}$$

Después de aplicar precisión, exhaustividad y acierto sobre el dataset etiquetado, obtenemos los resultados en la Tabla 6.3:

Con estos resultados podemos ver que la herramienta ha obtenido unos buenos resultados en la precisión, es decir, que el porcentaje de predicciones positivas han sido realmente positivas. Solamente se ha obtenido un nivel de precisión más bajo en la categoría 0, es decir, la ubicación. Con respecto a la exhaustividad, indica qué porcentaje de aciertos se han conseguido. En este caso no se han obtenido tan buenos resultados. En

	Precisión	Exhaustividad	Acierto
Categoría 0	0,28	0,39	78,90 %
Categoría 1	0,86	0,32	94,7 %
Categoría 2	0,95	0,33	98 %
Categoría 3 (Senti-Py)	0,65	0,57	51,9 %
Categoría 3 (librería GTI-IA)	0,61	0,14	39 %
Categoría 4	0,49	0,06	80,95 %
Categoría 5	0,60	0,265	56,64 %
Categoría 6	0,79	0,28	41,9 %

Tabla 6.3: Resultados precisión y exhaustividad.

todas las categorías se han dado valores bastante bajos. Por último, al observar el acierto, podemos ver que la herramienta acierta bastante en la mayoría de las categorías.

Una vez obtenidos estos resultados, como complemento, se han calculado la cantidad de falsos/verdaderos positivos/negativos (ver Tabla 6.4).

Categoría	FN	FP	VN	VP	# tweets de la categoría
Categoría 0	299	483	2733	192	491
Categoría 1	182	14	3422	89	271
Categoría 2	72	2	3597	36	108
Categoría 3 (Senti-Py)	1036	748	554	1369	2405
Categoría 3 (librería GTI-IA)	2076	210	1092	329	2405
Categoría 4	662	44	2958	43	705
Categoría 5	1298	309	1631	469	1767
Categoría 6	1956	198	783	770	2726

Tabla 6.4: Verdaderos/Falsos negativos/positivos.

Puede verse que la cantidad de Verdaderos Negativos (VN) es muy elevada. Esto provoca que el accuracy o acierto tenga valores más altos de los que debería. Esta medida es aconsejable usarla cuando los verdaderos negativos y positivos tienen la misma relevancia y cuando el dataset no está desbalanceado, pero en nuestro caso esto no es así. Por esta razón en trabajos futuros, cuando se mejore la herramienta, no será aconsejable utilizar el acierto para comprobar si las modificaciones realmente mejoran la herramienta o no. Por esto la medida más apropiada es es F1/media harmónica. Este valor se calcula utilizando precisión y recall, y como se ha mencionado antes, funciona mejor que el acierto en los casos en los que una respuesta predomina sobre la otra. Este coeficiente se calcula siguiendo la fórmula 6.10.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{6.10}$$

Los resultados al aplicar esta fórmula sobre los datos obtenidos anteriormente pueden verse en la Tabla 6.5.

La variable F1 es usada principalmente para la comparación de clasificadores, siendo el clasificador con mayor F1 el que tiene un mejor funcionamiento. Por esta razón podemos usar F1 para comparar las dos librerías de sentimiento. Podemos ver que la librería Senti-Py tiene un mayor F1. Además si observamos los verdaderos positivos de la Tabla 6.4, podemos ver que ésta identifica mejor los textos. Sin embargo, esto no asegura que funcione mejor, debido a que la librería del GTI-IA no identifica los textos positivos,

Categoría	F1
Categoría 0	0,33
Categoría 1	0,47
Categoría 2	0,49
Categoría 3 (Senti-Py)	0,60
Categoría 3 (librería GTI-IA)	0,22
Categoría 4	0,11
Categoría 5	0,37
Categoría 6	0,42

Tabla 6.5: Resultados del coeficiente F1.

por lo que es seguro que acertará en menos casos. En este caso, no nos podemos fijar solamente en el F1 ya que las librerías no ofrecen una funcionalidad exactamente igual.

6.3 Tiempos de ejecución

Esta prueba consiste en medir los tiempos de ejecución en dos casos: cuando se analiza un solo mensaje y cuando se analiza un dataset con 3707 mensajes. También se evaluará el tiempo de ejecución con la librería Senti-Py y GTI-IA.

6.3.1. Librería Senti-Py

En este apartado se analizan los tiempos de ejecución de la herramienta para 3707 mensajes y para un mensaje seleccionado de forma aleatoria haciendo uso de la librería Senti-Py.

Categoría	Tiempo de ejecución
Categoría 0	21s
Categoría 1	0,48s
Categoría 2	0,45s
Categoría 3 (Senti-Py)	337s
Categoría 4	0,46s
Categoría 5	0,44s
Categoría 6	0,12s
Tiempo total	$359,95s \approx 360s$

Tabla 6.6: Tiempos de ejecución para 3707 textos (con librería Senti-Py.)

En la Tabla 6.6 se muestran los resultados obtenidos al categorizar los 3707 mensajes. Se puede ver que los tiempos de ejecución más elevados se obtienen en la categoría 0 (21s), en la que se usa la librería spaCy, y en la categoría 3, en la que se usa la librería Senti-Py (337s). La herramienta en total tiene un tiempo de ejecución de 6 minutos, lo cual es un tiempo muy alto.

En la Tabla 6.7 tenemos los resultados para un solo mensaje. Al igual que en la anterior tabla, podemos ver que los dos tiempos de ejecución más altos siguen siendo los obtenidos para las categorías 0 y 3. La herramienta tarda un total de 3,6 segundos en analizar las categorías de un mensaje.

Categoría	Tiempo de ejecución
Categoría 0	0,60s
Categoría 1	0,004s
Categoría 2	0,0003s
Categoría 3 (Senti-Py)	3s
Categoría 4	0,0005s
Categoría 5	0,0003s
Categoría 6	0,0012s
Tiempo total	$3,6062s \approx 3,6s$

Tabla 6.7: Tiempos de ejecución para 1 texto (con librería Senti-Py).

6.3.2. Librería GTI-IA

En este apartado se van a analizar los tiempos de ejecución de la herramienta para un mensaje y para 3707 mensajes haciendo uso de la librería GTI-IA. En las Tablas 6.8 y 6.9 se pueden observar los resultados obtenidos. Al igual que con la anterior librería, las dos categorías que tienen el mayor tiempo de ejecución son la 0 y la 3. Sin embargo, con la librería GTI-IA, tanto para un mensaje como para los 3707, el tiempo de la categoría 3 es muy inferior comparado con los obtenidos con la librería Senti-Py. El tiempo de ejecución de la herramienta para los 3707 mensajes es de 26 segundos, y para un solo mensaje es de 0,34 segundos.

Categoría	Tiempo de ejecución
Categoría 0	21s
Categoría 1	0,50s
Categoría 2	0,47s
Categoría 3 (librería GTI-IA)	2,62s
Categoría 4	0,45s
Categoría 5	0,46s
Categoría 6	0,13s
Tiempo total	$25,93s \approx 26s$

Tabla 6.8: Tiempos de ejecución para 3707 textos (con librería GTI-IA).

Categoría	Tiempo de ejecución
Categoría 0	0,27s
Categoría 1	0,004s
Categoría 2	0,0002s
Categoría 3 (librería GTI-IA)	0,062s
Categoría 4	0,0005s
Categoría 5	0,0002s
Categoría 6	0,0012s
Tiempo total	$0,3381s \approx 0,34s$

Tabla 6.9: Tiempos de ejecución para 1 texto (con librería GTI-IA).

6.4 Pruebas de carga

Para del servicio web se han realizado pruebas de carga. Con estas pruebas se podrá comprobar cual es el máximo número de peticiones por segundo que puede soportar y cómo aumenta el tiempo de respuesta al aumentar el número de peticiones a atender.

Para ello se ha hecho uso de la herramienta JMeter. Esta herramienta permite realizar peticiones al servicio web simulando una gran cantidad de usuarios. Como se ha hecho en la anterior prueba, se van a hacer las pruebas de carga usando las dos librerías de emociones: Senti-Py y GTI-IA.

Para las pruebas se han ido haciendo peticiones HTTP y se han recogido los tiempos de respuesta y posteriormente se ha hecho la media. En concreto se han hecho dos tipos de pruebas. Por un lado se ha realizado una prueba para ver el número máximo de peticiones en un segundo que puede procesar. Por otro lado, debido a que este servicio web va a ser utilizado en talleres educativos, también interesa conocer cuántas peticiones simultáneas son soportadas en un minuto. Para realizar esta prueba se van a medir los tiempos petición/respuesta según el número de peticiones realizadas.

6.4.1. Librería Senti-Py

Los resultados obtenidos en las pruebas de carga utilizando la librería Senti-Py en el servicio web, se muestran en la Figura 6.4. El número máximo de peticiones que soporta el servicio en un segundo es de 10. Si se aumenta este número, el servicio web deja de estar disponible.

Figura 6.4: Tiempos de respuesta con la librería Senti-Py.

También se ha realizado la prueba de cuántas peticiones soporta en un minuto. Al igual que la anterior prueba, han sido 10 peticiones y los tiempos de ejecución han sido prácticamente los mismos.

6.4.2. Librería GTI-IA

Con la librería Senti-Py, se van a realizar dos pruebas (como con la anterior librería). Comenzamos por ver cuántas peticiones son soportadas en un segundo y sus respectivos tiempos de respuesta (ver Figura 6.5).

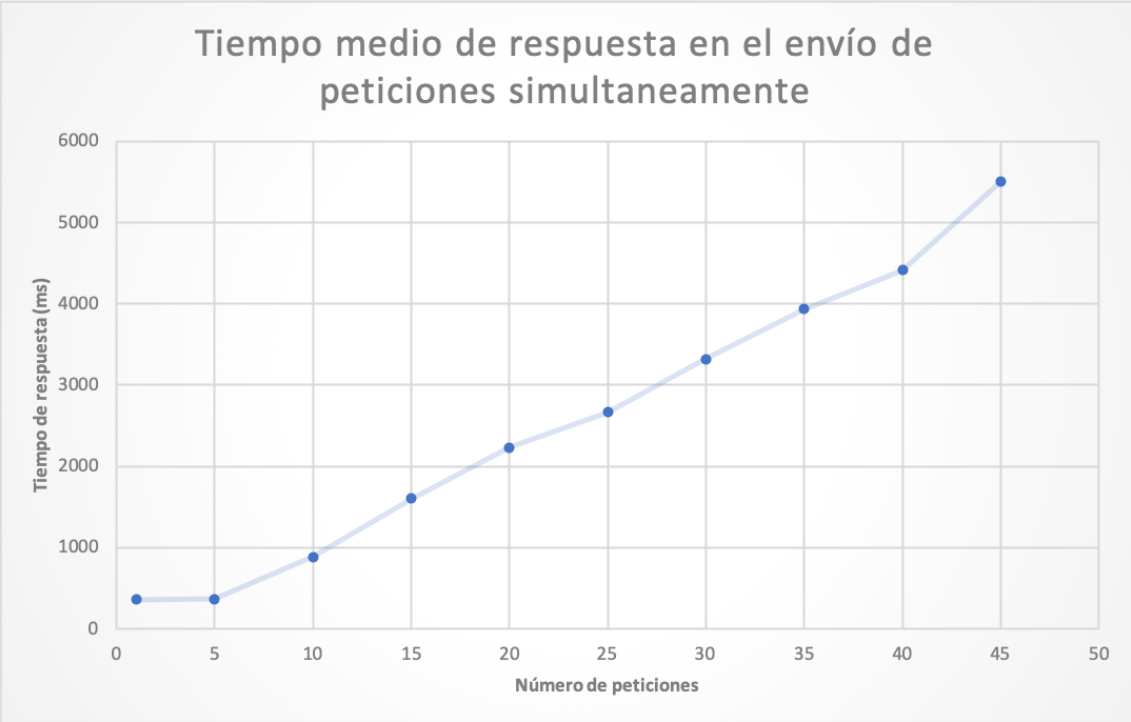


Figura 6.5: Tiempo medio de respuesta con la librería GTI-IA con peticiones simultáneas.

Se puede ver que el aumento del tiempo es lineal y se soportan un máximo de 45 peticiones en un segundo. Si aumentamos más este número se comienzan a producir errores y se pierden parte de las respuestas del servicio web.

A continuación, se muestra cuántas peticiones se pueden procesar como máximo en un minuto y los tiempos medios de respuesta (ver Figura 6.6).

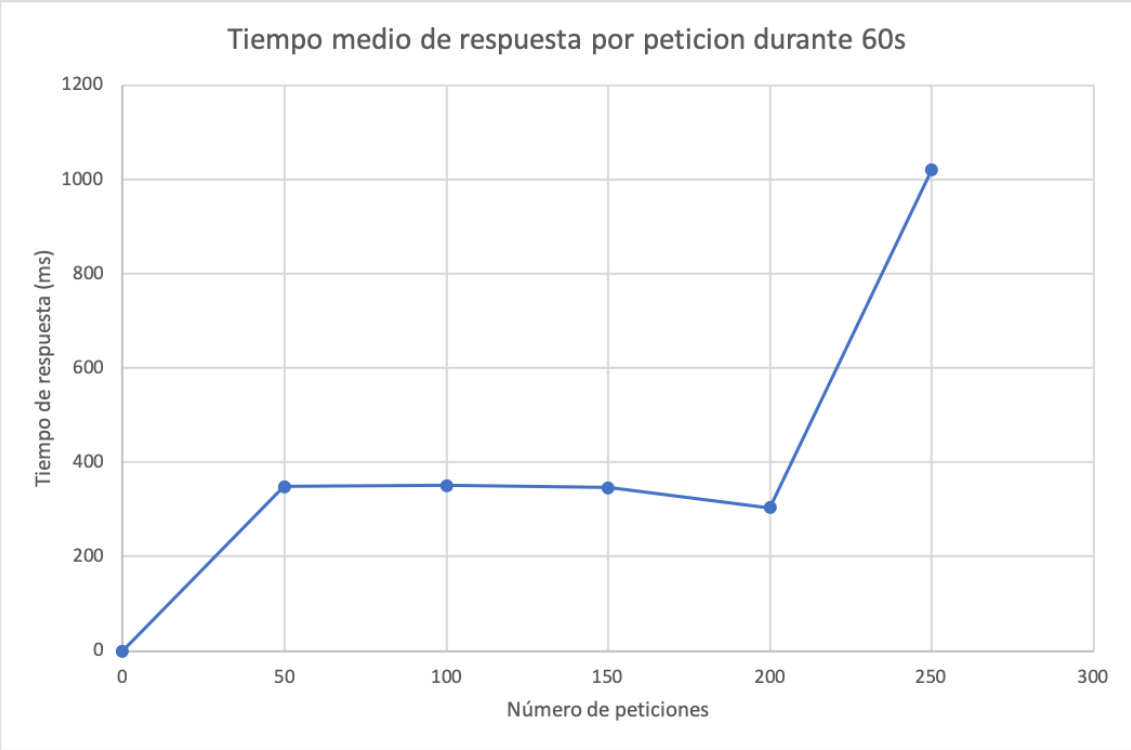


Figura 6.6: Tiempo medio de respuesta con la librería GTI-IA durante 60s.

Se puede ver que a partir de las 250 peticiones hay un gran aumento en el tiempo de respuesta. Si aumentamos este número comienzan a perderse paquetes, por lo que el máximo de peticiones por minuto que podemos hacer al servicio web ronda las 250 peticiones.

6.5 Análisis de los resultados del servicio web y librería escogida

Se puede ver claramente que la librería del GTI-IA tiene unos resultados mucho mejores que Senti-Py, tanto en tiempos de respuesta como por máximo de peticiones atendidas. Además cumple con los requisitos de rendimiento que se había mencionado anteriormente en el apartado 3.1.3 donde se dice que se esperaba que se pudieran procesar por lo menos entre 180 y 200 peticiones por minuto. Si el servicio web hace uso de Senti-Py solamente soporta alrededor de unas 10 peticiones, lo cual haría imposible el uso de este servicio en las actividades de talleres.

Por estas razones, la librería que se va a usar en el servicio web va a ser la del GTI-IA, debido a que soporta una mayor capacidad para atender peticiones y tiene unos tiempos de respuesta mucho menores.

CAPÍTULO 7
Conclusiones

En este TFG se ha desarrollado una herramienta para la ayuda a la toma de decisiones de usuarios que publican información textual en redes sociales. La herramienta consiste en un módulo de detección de categorías sensibles (ubicación, datos personales, emociones, etc.) que se ha implementado como un servicio web. Para la detección de las distintas categorías se han utilizado librerías de reconocimiento de entidades, ontologías, diccionarios y librerías de análisis de sentimientos. Además, una vez analizado el texto, la herramienta genera un mensaje informativo sobre la sensibilidad del contenido. Este mensaje contiene las categorías detectadas y el grado de sensibilidad, y tiene como objetivo persuadir de la publicación si esta tiene información sensible. La herramienta se ha integrado dentro de la red social Pesedia a través de un plug-in.

Respecto a los objetivos planteados al comienzo del trabajo, se puede afirmar que se han completado de manera satisfactoria. Durante el desarrollo del TFG se han ido cumpliendo los objetivos indicados para llevar a cabo el desarrollo de la herramienta.

Inicialmente se realizó una investigación buscando artículos y trabajos relacionados con la privacidad y la sensibilidad. Esta investigación permitió detectar qué funcionalidades no proporcionaban estos trabajos y que serían relevantes para el problema que planteaba el proyecto.

Antes de empezar el desarrollo de la herramienta se realizó una especificación formal de requisitos con el objetivo de determinar las características y los aspectos funcionales de la herramienta.

El proceso de implementación de la herramienta implicó el desarrollo de un módulo de extracción de categorías sensibles de un texto. Para ello, fue necesario preprocesar y limpiar textos procedentes de redes sociales haciendo uso de expresiones regulares y librerías como Pandas o Stopwords. También fue necesario establecer qué categorías eran sensibles en base a trabajos previos. Finalmente se decidió por la consideración de siete categorías de información: ubicación, salud, drogas/alcohol, emociones, ataques personales, detalles personales y neutro. Para la detección automática de las categorías hubo que analizar si ya existían herramientas y/o librerías que facilitaran esta tarea y en caso contrario implementarlas. Se encontraron librerías que si fueron útiles (i.e., spaCy, SentiPy y una librería desarrollada por el grupo GTI-IA). Al buscar herramientas se tuvo que investigar el funcionamiento de las ontologías. En función de las categorías detectadas, se planteó una medida de sensibilidad. El valor de esta medida junto con las categorías generadas fueron utilizadas para generar un mensaje informativo para el usuario.

Para poder integrar la herramienta desarrollada en la red social Pesedia se encapsuló el módulo de extracción de categorías en un servicio web y se creó un plug-in utilizando PHP y JavaScript para lograr la comunicación entre la herramienta y Pesedia.

Finalmente se realizó una evaluación de la herramienta. Para ello fue necesario la utilización de un dataset de textos procedentes de la red social Twitter. Este dataset era muy reciente y no se había evaluado el nivel de acuerdo en su etiquetado, por lo que se tuvo que investigar qué medidas existían para ello. Concretamente se encontró la medida PABAK basada en otra medida Fleiss Kappa. A continuación se buscaron técnicas para evaluar el correcto funcionamiento de la herramienta, y se encontraron las medidas de precisión, exhaustividad, acierto y media harmónica. Por último, se realizaron pruebas para comprobar el rendimiento de la herramienta haciendo uso de la herramienta JMeter.

Una vez finalizado el proyecto, se puede afirmar que se han completado todos estos objetivos que se habían planteado inicialmente. En el proyecto se han aprendido nuevas tecnologías y herramientas las cuales será útiles en el futuro y se han utilizado los conocimientos aprendidos en el grado, como se menciona más adelante.

A lo largo del proyecto se han ido encontrado ciertos problemas que se han tenido que solucionar o encontrar otra forma de hacer las cosas. Por ejemplo, uno de los problemas ha sido el idioma. Esto se debe a que la mayor parte de librerías que existen para analizar el texto son para texto en inglés y no en español. Por está razón se han usado unas técnicas distintas para la detección de categorías de las que se habían pensado en un primer momento.

Se han tenido que aprender nuevas tecnologías, por ejemplo, Python y sus múltiples librerías. También se han adquirido conocimientos sobre ontologías, que aunque finalmente no se han utilizado en la herramienta, si que se ha estudiado su funcionamiento. Se han utilizado herramientas para la evaluación de aplicaciones como Jmeter, una herramienta muy útil para hacer pruebas de carga que se ha usado para verificar el correcto funcionamiento del servicio web, cumpliendo con todos los objetivos que se habían planteado inicialmente.

7.1 Relación del trabajo desarrollado con los estudios cursados

Para desarrollar este proyecto se han necesitado conocimientos sobre ciertas tecnologías y herramientas. Por esta razón, muchas asignaturas cursadas del grado han sido de gran utilidad para llevar a cabo este proyecto. En concreto las más relacionadas con el proyecto han sido: programación, estadística, desarrollo web, diseño web, redes, bases de datos e ingeniería del software.

Para finalizar, se van a mencionar algunas de las competencias transversales que se han evaluado durante el grado y que han ido alcanzando del proyecto:

- CT_03 - Análisis y resolución de problemas: a lo largo del proyecto han ido apareciendo diferentes problemas que se han tenido que ir resolviendo.
- CT_07. Responsabilidad ética, medioambiental y profesional: se ha hecho uso de datos de Twitter que han sido anonimizados para que no puedan ser relacionados con el usuario que lo publicó.
- CT_10 - Conocimiento de problemas contemporáneo: se ha visto los problemas que tienen los usuarios al hacer uso de las redes sociales.
- CT_11 - Aprendizaje permanente: se han aprendido nuevas tecnologías y herramientas las cuales se usarán en un futuro.
- CT_12 - Planificación y gestión del tiempo: ha sido necesario llevar una planificación y una buena gestión del tiempo para poder acabar el trabajo a tiempo para su entrega.

7.2 Trabajos futuros

Para finalizar esta sección, se van a plantear posibles mejoras y/o ampliaciones que se podrían desarrollar partiendo del proyecto desarrollado.

Una de estas mejoras tiene que ver con el servicio web, debido a que soporta un máximo de 250 usuarios, lo cual está bien para el uso que se le quiere dar actualmente, pero en caso de querer usarse en otros sitios, como por ejemplo en un instituto, el número de usuarios debería de ser mucho mayor. Por esta razón una de las soluciones es desplegar esta herramienta en la nube, haciendo uso de servicios como el de Amazon Web Services o Google Cloud. También se plantea como un trabajo futuro, almacenar datos sobre los mensajes procesados para poder generar estadísticas y comprobar si la herramienta provoca que los usuarios piensen mejor que es lo que van a publicar.

Como posible ampliación de la herramienta, se ha pensado en el desarrollo de análisis de imágenes, debido a que actualmente varias redes sociales que están creciendo rápidamente están basadas en la publicación de imágenes que pueden revelar información sensible sobre los usuarios.

Por último, en un futuro la herramienta será integrada como parte de un sistema de argumentación. La herramienta servirá para la identificación de ciertas categorías que servirán para poder dar argumentos a los usuarios sobre lo que publican y su privacidad.

Código

El código del proyecto está disponible en el repositorio de Github: https://github.com/vbotti/twitterDatos_TFG

Bibliografía

[1] A. Acquisti, I. Adjerid, R. Balebako, L. Brandimarte, L. F. Cranor, S. Komanduri, P. G. Leon, N. Sadeh, F. Schaub, M. Sleeper, et al. Nudges for privacy and security: Understanding and assisting users' choices online. *ACM Computing Surveys (CSUR)*, 50(3):44, 2017.

[2] A. Acquisti, C. Taylor, and L. Wagman. The economics of privacy. *Journal of Economic Literature*, 54(2):442–92, June 2016.

[3] G. Aguado, V. Julian, and A. Garcia-Fornes. Towards aiding decision-making in social networks by using sentiment and stress combined analysis. *Information*, 9(5):107, 2018.

[4] J. Alemany, E. del Val, J. Alberola, and A. García-Fornes. Estimation of privacy risk through centrality metrics. *Future Generation Computer Systems*, 82:63–76, 2018.

[5] J. Alemany, E. del Val, J. Alberola, and A. García-Fornes. Enhancing the privacy risk awareness of teenagers in online social networks through soft-paternalism mechanisms. *International Journal of Human-Computer Studies*, 2019.

[6] Boletín Oficial del Estado. Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. https://boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf, 12 2018.

[7] T. Byrt, J. Bishop, and J. B. Carlin. Bias, prevalence and kappa. *Journal of clinical epidemiology*, 46:423–9, 06 1993.

[8] A. Caliskan Islam, J. Walsh, and R. Greenstadt. Privacy detective: Detecting private information and collective privacy behavior in a large social network. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 35–46. ACM, 2014.

[9] E. del Val, C. Martínez, and V. Botti. Analyzing users' activity in online social networks over time through a multi-agent framework. *Soft Computing*, 20(11):4331–4345, 2016.

[10] R. Falotico and P. Quatto. Fleiss' kappa statistic without paradoxes. *Quality & Quantity*, 49(2):463–470, 2015.

[11] B. Goodman and S. Flaxman. European union regulations on algorithmic decision-making and a right to explanation". *AI Magazine*, 38(3):50–57, 2017.

[12] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.

[13] A. Lenhart. *Teens, Online Stranger Contact & Cyberbullying: What the Research is Telling Us–*. Pew Internet & American Life Project, 2008.

[14] S. Livingstone. Taking risky opportunities in youthful content creation: teenagers' use of social networking sites for intimacy, privacy and self-expression. *New media & society*, 10(3):393–411, 2008.

[15] M. Madden, A. Lenhart, S. Cortesi, U. Gasser, M. Duggan, A. Smith, and M. Beaton. Teens, social media, and privacy. *Pew Research Center*, 21:2–86, 2013.

[16] H. Mao, X. Shuai, and A. Kapadia. Loose tweets: an analysis of privacy leaks on twitter. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 1–12. ACM, 2011.

[17] C. O. Marañón. Redes sociales y jóvenes: una intimidad cuestionada en internet. *Aposta. Revista de Ciencias Sociales*, (54), 2012.

[18] G. Méndez. Especificación de requisitos según el estándar de ieee 830. *Facultad de Informática, Universidad Complutense de Madrid*, 2008.

[19] Ministerio de Sanidad, Servicios Sociales e Igualdad. Alcohol, tabaco y drogas ilegales en España. www.pnsd.mscbs.gob.es/profesionales/sistemasInformacion/informesEstadisticas/pdf/2017OEDA-INFORME.pdf.

[20] D. L. Mothersbaugh, W. K. Foxx, S. E. Beatty, and S. Wang. Disclosure antecedents in an online service context: The role of sensitivity of information. *Journal of service research*, 15(1):76–98, 2012.

[21] Pantallas Amigas. Privacy and data protection. https://www.pantallasamigas.net/en/privacidad-y-proteccion-de-datos/#googtrans(es|en). Accessed: 2019-06-14.

[22] R. G. Pensa and G. Di Blasi. A privacy self-assessment framework for online social networks. *Expert Systems with Applications*, 86:18–31, 2017.

[23] D. M. Powers. The problem with kappa. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 345–355. Association for Computational Linguistics, 2012.

[24] J. M. Rumbold and B. K. Pierscionek. What are data? a categorization of the data sensitivity spectrum. *Big Data Research*, 12:49 – 59, 2018. Big Data Centric Computational Intelligence in Bioinformatics and Healthcare.

[25] D. Sánchez and A. Viejo. Privacy risk assessment of textual publications in social networks. In *ICAART (1)*, pages 236–241, 2015.

[26] A. Sengupta and A. Chaudhuri. Are social networking sites a source of online harassment for teens? evidence from survey data. *Children and Youth Services Review*, 33(2):284–290, 2011.

[27] M. Sleeper, J. Cranshaw, P. G. Kelley, B. Ur, A. Acquisti, L. F. Cranor, and N. Sadeh. I read my twitter the next morning and was astonished: A conversational perspective on twitter regrets. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 3277–3286. ACM, 2013.

[28] Statista. Social media statistics & facts. https://www.statista.com/topics/1164/social-networks/. Accessed: 2019-06-14.

[29] E. Steel, C. Locke, E. Cadman, and B. Freese. How much is your personal data worth. *Financial Times*, 12, 2013.

[30] N. Talukder, M. Ouzzani, A. K. Elmagarmid, H. Elmeleegy, and M. Yakout. Privometer: Privacy protection in social networks. In *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)*, pages 266–269. IEEE, 2010.

[31] E. Vanderhoven, T. Schellens, and M. Valcke. Educating teens about the risks on social network sites. an intervention study in secondary education= enseñar a los adolescentes los riesgos de las redes sociales: Una propuesta de intervención en secundaria. *Comunicar*, 22(43):123–132, 2014.

[32] E. Vanderhoven, T. Schellens, R. Vanderlinde, and M. Valcke. Developing educational materials about risks on social network sites: a design based research approach. *Educational technology research and development*, 64(3):459–480, 2016.

[33] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. Cranor. I regretted the minute i pressed share: A qualitative study of regrets on facebook. In *Proceedings of the seventh symposium on usable privacy and security*, page 10. ACM, 2011.