



UNIVERSIDAD NACIONAL DEL SUR

DEPARTAMENTO DE CIENCIAS E INGENIERÍA
DE LA COMPUTACIÓN

Sistema de análisis de sentimientos en Reddit

Trabajo final de carrera realizado por
Aristegui Maximiliano Marcos

Directora:

Ana Gabriela Maguitman

Índice

<i>Introducción.....</i>	<i>3</i>
<i>Reddit.....</i>	<i>6</i>
<i>Conceptos y herramientas.....</i>	<i>9</i>
<i>Instalación de librerías y configuración</i>	<i>12</i>
<i>Obtención de los comentarios.....</i>	<i>15</i>
<i>Análisis de los comentarios.</i>	<i>18</i>
<i>Aprendizaje</i>	<i>22</i>
<i>Proceso de visualización.....</i>	<i>24</i>
<i>Conclusiones</i>	<i>27</i>
<i>Futuras mejoras del sistema</i>	<i>29</i>
<i>Referencias.....</i>	<i>30</i>

Introducción – Justificación

“El hombre es un ser social por naturaleza”

Aristóteles (384-322, a. de C.)

Los seres humanos, al igual que muchos animales, nacemos con características sociales que nos motivan a relacionarnos con otros. Desde el inicio de los tiempos hemos ido desarrollando diferentes formas de comunicación, ya sea verbal o no verbal, con el fin de comunicarnos y poder interaccionar entre nosotros.

Con el paso del tiempo, esto se fue desarrollando y derivando en la creación de distintos lenguajes propios de cada región del mundo, algunos similares y otros diferentes ya sea en su estructura gramatical como en su sintaxis. Esto también dio lugar a la existencia de formas de interpretar las cosas, teniendo palabras o expresiones propias de cada lenguaje.

Este lenguaje no solo nos ha permitido interaccionar con otros seres humanos de forma oral, sino también (y con el paso del tiempo) mediante la escritura. Las cartas fueron una nueva forma de comunicarnos entre nosotros que permitían “enviar” palabras a la distancia, pudiendo escribir y trasmitir ideas a personas que, incluso, podrían estar en otro continente. Poco después aparecieron los primeros teléfonos, que a diferencia de las cartas, estas permitían una comunicación en tiempo real. Esto implicaba que se reducían notoriamente los tiempos de respuesta, al no tener que esperar que la carta sea enviada, recibida, leída y respondida. De esta manera, con estos teléfonos, dos personas podían estar manteniendo una conversación a la distancia de una forma muy similar a como lo harían si estuvieran cara a cara.

Con la evolución de la tecnología y la aparición del internet, no tardamos mucho en adoptar una forma nueva de comunicación para aprovechar estos avances. La aparición de los correos electrónicos fue una evolución casi natural de las cartas que se solían escribir con papel y tinta. De esta manera, el proceso de comunicarse con otra persona se reducía en tiempo y esfuerzo gracias a la propia velocidad de internet.

Llego un momento donde gano bastante popularidad la aparición de distintos foros de comunicación, centrados en una comunicación en masa donde un mayor grupo de gente pudiera interactuar. Como era relativamente sencillo crear un foro, e incluso algunas empresas brindaban servicios gratuitos para crear nuevos, su magnitud creció rápidamente hasta el punto donde se podían encontrar lugares para hablar de cualquier tema que uno se imaginara, y si no existía, crearlo.

Dada la naturaleza social de las personas, los foros cobraron muchísima popularidad entre todas las generaciones y gustos. Foros de animales, comidas, series, foros orientados a los más chicos, foros con temáticas científicas. Existían muchos recursos y

eran fácilmente accesibles para todos, brindando un escenario donde todos pudieran encontrar personas con sus mismos gustos o curiosidades con quien interactuar.

Claramente, esto fue el inicio de lo que, a futuro, terminaría evolucionando en lo que hoy son las redes sociales. Las redes sociales no son más que grandes sistemas donde se puede interaccionar con todas las personas del mundo, compartiendo imágenes, videos, comentarios, e incluso teniendo la posibilidad de incluir lista de amigos permitiendo generar círculos sociales.



A principios del año 2020, como consecuencia de un suceso nunca antes visto en el mundo, la forma de comunicación volvió a cambiar. El inicio del aislamiento social a nivel mundial a causa de la pandemia generó un impacto enorme en la comunicación no verbal. Ya no podías estar comunicándote en persona con tus compañeros de trabajo en una oficina, o con amistades en clubes sociales. Las medidas tomadas en todo el mundo para prevenir la expansión del virus generaron una dependencia aún mayor en la comunicación por medio de internet.

Claramente, al no tener la misma facilidad de comunicarse con otras personas, se volvió un poco más difícil enterarse de muchas situaciones que estaban ocurriendo. Las redes sociales y distintos foros de noticias vieron su popularidad incrementada, al ser portales de transmisión de información en tiempo real. Como muchos de estos sistemas brindan la posibilidad a sus usuarios de crear y publicar contenido de forma gratuita, cualquiera

puede estar comentando noticias prácticamente en el momento en el que están ocurriendo.

Durante el desarrollo de este proyecto, y mientras se arrancaba a poder ver los primeros resultados, ocurrieron os sucesos relacionados con el conflicto de Rusia-Ucrania, aumentando la popularidad de algunos subreddits y generando variaciones en las emociones generales obtenidas. La mayor parte de la información mostrada durante este informe fue tomada durante este proceso, por lo cual los resultados (emociones predominantes, palabras aprendidas) van a estar íntegramente relacionadas con este contexto.



Simple_Somewhere_564 · 17 hr. ago

I hate saying this because it's about a war but the way information is organized, delivered concisely and regularly has been absolutely amazing. Ive never been so informed about government actions.



32



Reply



Share



Report



Save



Follow



Tetizeraz · +2 · 16 hr. ago

Brazil "What is a Brazilian doing modding r/ukraine?"

It really is weird, isn't it? I mean, if you look how information is conveyed, how our community is organized. Everyone has a smartphone these days, Ukrainian soldiers (and volunteers) can easily take a picture after a battle, lots of Ukrainians can share real time information and footage of the conflict, etc.



8



Reply



Share



Report



Save



Follow



Simple_Somewhere_564 · 10 hr. ago

It's the whole package really. If they just shot tik toks without a narrative, it would be different. The daily briefings by Zekinsky, the translators, subs like these with fantastic moderators. It's everything.



3



Reply



Share



Report



Save



Follow



Madame_Arcati · +1 · 17 hr. ago

Thank you so much for the distillation of news reports, and for sorting them into subject areas. Posts across reddit seem to have less and less attribution and only a headline, so now I just come here. Really appreciate your work.



7



Reply



Share



Report



Save



Follow

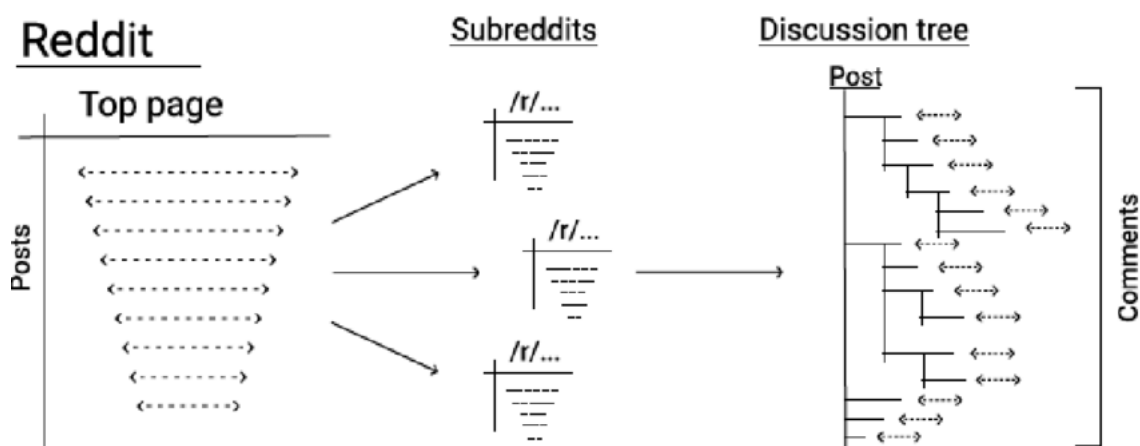
Reddit



En el 2015, con el auge de la masificación de los foros, apareció una nueva plataforma conocida como Reddit. Consiste en una colección masiva de foros, donde en cada uno de estos foros las personas pueden compartir noticias y contenido, además de comentar en las publicaciones de otras personas. Cada subreddit pasa a formar parte de la lista completa de envíos de Reddit, lo cual significa que una publicación en cualquier subreddit (a menos que sea privada) puede llegar a la página principal del sitio web.

La forma en la que se estructura Reddit es bastante sencilla, y respeta una estructura de tipo árbol. Reddit tiene una página principal, desde la cual se pueden acceder a distintos subreddits. Cada subreddit representa un tema o concepto distinto, sobre el cual los distintos usuarios podrán crear discusiones, comentar, votar, etcétera.

Cada subreddit se divide en post, que son un tema de charla puntual iniciado por un usuario. Los post, al igual que los subreddits, se pueden organizar de distinta manera, ya sea por antigüedad (el más nuevo primero) o por popularidad. Cada post puede tener respuestas, conocidas como comentarios, que también se organizan en una estructura de tipo árbol. La idea es que, cuando un usuario responde a un comentario hecho por otro usuario, esta respuesta pasa a depender directamente del comentario original. De esta manera se pueden interpretar los comentarios según el nivel donde se encuentran, siendo los comentarios de primer nivel los comentarios principales del tópico, los comentarios de segundo nivel son las respuestas directas a los comentarios de primer nivel y así sucesivamente.



Estructura de Reddit

Una de las principales razones por la que Reddit se ha convertido en una plataforma extremadamente popular en los últimos años es la facilidad que brinda para crearse un usuario y arrancar a interactuar en los diferentes subreddits. Esto permite que se creen espacios para interactuar sobre diferentes temas, donde cada usuario puede llegar a encontrar subreddits que les resulten más interesantes y donde poder colaborar. El nombre de este sitio es la contracción de la frase en inglés «I already read it», que significa «Ya lo leí». El idioma principal de Reddit es el inglés, aunque existen subreddits para todos los idiomas.

Aunque existen muchos subreddits dedicados específicamente a hobbies, ocio y cosas de humor, también existen otros dedicados a dar noticias sobre distintos acontecimientos del mundo, otros dedicados a la investigación donde se puede interactuar con profesionales de distintas áreas e ir compartiendo (de una manera más informal) las novedades y avances en distintas áreas de conocimiento. Aunque no es recomendable considerarlo una fuente confiable a la hora de sacar conclusiones, si es verdad que es uno de los portales donde más rápido se pueden llegar a difundir distintas noticias, y donde la activa comunidad que posee puede compartir y reaccionar a ellas. Reddit es diferente a otras redes sociales de una manera muy especial. El público de Reddit está preparado e interesado en aprender y tener conversaciones profundas entre sí sobre cualquier tema.

Si una publicación recibe muchos votos, sube en la clasificación de Reddit y más gente puede verla. Por el contrario, si recibe votos a la baja, su visibilidad se reduce hasta desaparecer de la vista de la mayoría de los usuarios. Esto permite generar una calificación dinámica interna sobre que tópicos de conversación son más interesantes para el público en general. Esta clasificación es interna de cada subreddit, con lo que no afecta directamente a la popularidad de un subreddit. Sin embargo, la cantidad de usuarios activos diariamente en un determinado subreddit si puede aumentar su popularidad, brindando de esta manera una distinción entre la popularidad de un subreddit y la popularidad de un tópico:

- Un subreddit gana popularidad en base a la cantidad de usuarios que creen nuevos tópicos, respondan a los ya existentes o brinden votos a favor y en contra de ellos.
- Un tópico gana más popularidad dentro de un subreddit en base a la cantidad de votos que tenga.

Actualmente cuenta con más de un millón de subreddits (también llamadas comunidades) donde cada una engloba un tema diferente. Cualquier persona con una cuenta Reddit puede crear un subreddit para cierto tema sin importar su naturaleza, siempre que se mantenga dentro de las pautas de la comunidad del sitio, y estos subreddits son administrados por moderadores, gente voluntaria que pueden editar la apariencia, dictaminar el contenido permitido, eliminar publicaciones e incluso restringir el acceso a ciertos usuarios.

Esta plataforma ha cobrado tanta popularidad que incluso varias marcas arrancaron a tener interacciones con los usuarios, creando sus propias comunidades. Grandes marcas como pueden ser Toyota, SpaceX, Spotify e incluso diarios como The Economist suelen publicar algunas noticias en Reddit, buscando interaccionar con sus usuarios de una forma mucho más directa. Suelen hacer lo que se conoce como AMA (Ask Me Anything – Pregúntame lo que sea) donde buscan ver el interés por sus usuarios. Esto no solo les sirve como una estrategia de marketing, sino también para realizar estudios de mercado.

↑ Posted by u/toyotausa 3 years ago 🏆

326 ↓ **Hey Reddit, Toyota division group vice president, Jack Hollis here! Join me, Supra expert, Ben Haushalter and Formula Drift driver Fredric Aasbo to talk the new 2020 Supra. The Supra is BACK! Ask Us Anything.**

Specialized Profession

Today we unveiled the fifth-generation GR Supra, the first global GAZOO Racing model, at the North American International Auto Show in Detroit, ending years of anticipation and speculation. You may have seen Akio Toyoda unveil it live on Reddit, but we're here to give you all the details you can't get in a press conference. Toyota division group vice president, Jack Hollis, Supra expert, Ben Haushalter and Formula Drift driver Fredric Aasbo are here to give you all the details. Need information on the Supra's return to the company...ask Jack. Want to know all the specs...ask Ben Got your racing gloves on, but need to know how it drives...ask Fredric

And if you still need more Supra in your life after the AMA, subscribe to our new podcast, Toyota Untold, to hear an exclusive episode dedicated to Supra. Available on iTunes, Google Play, Spotify or wherever you get your podcasts.

Apple: <https://itunes.apple.com/us/podcast/id1444305760?mt=2>

Spotify: https://open.spotify.com/show/5E0IdlvTAS7t6Jt5jFqaza?si=FiTvS3o3Rsm_68qvan2PoA

Have a Supra New Year!

Proof: <https://twitter.com/Toyota/status/1083483678028824577>

EDIT: Thanks Reddit so much for all of the questions. That's a wrap for us today on the AMA, but if you're looking for additional info about Supra, we've added a few links below to checkout:

<https://www.toyota.com/upcoming-vehicles/gr-supra/>

<https://www.motortrend.com/cars/toyota/supra/2020/2020-toyota-supra-first-look-review/>

<https://pressroom.toyota.com/releases/legend+returns+2020+toyota+supra+makes+world+debut.htm>

💬 777 Comments ➦ Share 📌 Save 🙋 Hide 🚩 Report 80% Upvoted

AMA de Toyota

Conceptos y herramientas

Python:

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. Es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano, además de ser un lenguaje multiplataforma y de código abierto. Parte de las razones por las que se eligió este lenguaje fue por su simplicidad y su gran número de bibliotecas de procesamiento de datos. Además, Reddit brinda una API para obtener información de su servidor, llamada Python Reddit API Wrapper (PRAW). Esta nos permite iniciar sesión en la API de Reddit para interactuar directamente con el backend del sitio web.

Un componente importante para el que se usó Python en este proyecto fue la generación del **crawler**, que consiste en un sistema encargado de obtener distintas URLs de internet en un formato apto para extraer su contenido. De esta manera, se pueden obtener los distintos comentarios publicados por los usuarios, para luego ser analizados.

MongoDB:

Esta es una base de datos basada en documentos y de uso gratuito, fácil de usar y con buena compatibilidad con Python gracias a la librería PyMongo. Su estructura interna se basa en documentos flexibles similares a los JSON, permitiendo que los campos y la estructura de cada documento puedan cambiar.

El usar una base de datos basado en documentos y que sea Schema Less como lo es MongoDB permite que tu base de datos crezca con la aplicación sin tener que ejecutar scripts que crean campos con valores por defecto cada vez que se quiera agregar un campo nuevo en los registros. Es normal que los documentos dentro de una colección no tengan exactamente los mismos campos, lo que le da mayor flexibilidad a la hora de almacenar información pero aumentan un poco la complejidad a la hora de analizarla.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Se desarrolló como un formato simple de leerlo para los humanos, a diferencia de otros formatos que pueden resultar un poco más complejos, pero que al mismo tiempo le sea fácil a la computadora interpretarlo y generarlo. Aunque está basado en el lenguaje de programación JavaScript, este formato logró ser independiente del lenguaje hace ya varios años, permitiendo que aplicaciones desarrolladas en otros lenguajes de programación (como Python) arrancaran a utilizarlo como medio de intercambio y almacenamiento de información.

La estructura de los objetos de tipo JSON está conformada por:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocidos como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias

Ambas estructuras son universales, esto quiere decir que (virtualmente) todos los lenguajes de programación las soportan de una forma u otra. Esto permite que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

Grafana:

Para la visualización de los datos se optó por el uso de Grafana, que es una herramienta de interfaz de usuario centralizada en la obtención de datos a partir de consultas, como también del almacenamiento de estos y su visualización. Consistentemente como ocurría con Python o MongoDB, esta herramienta es gratuita y fácil de usar, lo que permite generar visualizaciones de datos de forma rápida y elegante en poco tiempo. Aunque Grafana tiene un plugin compatible con MongoDB, lo que permitiría que se accedieran a los datos de forma directa, este plugin requiere un pago, por lo que se optó por usar un segundo motor de base de datos para almacenar la información a mostrar. Esta decisión afecta directamente a la eficiencia, ya que la magnitud del proyecto no es lo suficientemente grande o compleja para realmente necesitar dos motores de base de datos diferentes.

Una alternativa que se evaluó fue desarrollar una interface gráfica desde cero, en algunas herramientas compatibles con Python como puede ser Qt (que es un framework multiplataforma orientado a objetos usado para el desarrollo de interfaces graficas). El problema con esta alternativa radicaba en dos factores:

- Por un lado, la complejidad del proyecto aumentaba. A diferencia de Grafana que está pensado como plataforma para la visualización de datos (y por esto ya tiene muchas opciones pre configurables que ayudan a este proceso), el desarrollo de una interface gráfica desde cero implicaba el aprendizaje de varios nuevos conceptos del diseño de GUIs, uso de librerías y los problemas derivados de la ubicación y coordinación de los objetos visuales.
- Por otro lado, la calidad de la visualización que brinda Grafana ya es bastante elevada, puesto que está siendo optimizada y mejorada desde su publicación haciendo que todo esté mucho más pulido y tenga una terminación visual mucho más agradable.

InfluxDB:

El segundo motor de base de datos utilizado fue InfluxDB, una base de datos orientada a registrar datos científicos o técnicos de medición temporales. Es de código abierto y muy útil cuando se tienen datos que dependen (o se puedan definir) en función del tiempo. En este proyecto en particular, como el sistema analiza los distintos subreddits y post en forma constante y paralela, las respuestas se van generando en función de la cantidad de mensajes cada post.

Como por lo general los tópicos más populares de un subreddit son los que van a tener más comentarios activos, y como resultado de la paralización del proceso de análisis de cada tópico utilizando diferentes hilos, los tiempos contemplados en la base de datos van a estar relacionados con el tiempo que tardó en ejecutar el análisis en ese tópico en particular. Esto quiere decir que el ordenamiento de los tiempos va a estar directamente relacionado con la cantidad de información analizada por cada tópico.

NRC Word-Emotion Association Lexicon

Este recurso consiste en una tabla gratuita disponible en internet, puesta a disposición por Safi Mogammad, en la cual se encuentran clasificadas 14182 palabras del idioma inglés en distintas clasificadas en ocho emociones básicas (enojo, miedo, anticipación, confianza, sorpresa, tristeza, felicidad y disgusto) y dos sentimientos (positivos y negativos). Es usada generalmente en el campo de procesamiento de lenguaje natural (NLP) para distinto análisis de emociones basado en documentos, comentarios, detección de lenguaje abusivo entre otros.

Esta tabla es utilizada como base principal para el análisis de las emociones de los comentarios de Reddit en este proyecto, sin embargo existen muchas más palabras propias del lenguaje usado en internet que no están contempladas en esta tabla. Por esta razón se implementó un pequeño algoritmo de aprendizaje de nuevas palabras basadas en el contexto en el cual fueron utilizadas y la cantidad de veces que aparecieron. De esta manera se logra expandir la cantidad de palabras cuyas emociones y sentimientos son conocidas, logrando a la larga un mejor análisis. Cada palabra, sus emociones y sentimientos asociados, van a formar una entrada en la base de datos de MongoDB, encargada principalmente del análisis de las emociones.

Instalación de librerías y configuración

En primera instancia, la principal tarea inicial radicaba en poder encontrar una manera de obtener los comentarios de Reddit. Para esto la principal herramienta usada consiste en una API llamada PRAW, la cual fue desarrollada y actualmente mantenida por la comunidad. Esta herramienta permite generar una conexión con Reddit de forma sencilla, desde la cual podemos iniciar diferentes peticiones las cuales nos permitirán irnos moviendo en toda la estructura de Reddit y sus subreddits.

La instalación, como suele ser bastante consistente con muchas de las librerías de Python, sigue la siguiente expresión:

```
pip install praw
```

Luego de esto, ya vamos a poder importarla en nuestro proyecto y comenzar a utilizarla. Aunque antes de poder arrancar a interactuar con Reddit, primero vamos a tener que brindar alguna forma de autenticación. Esta autenticación se puede generar de cuatro maneras distintas:

- *Web Applications*
- *Installed Applications*
- *Script Applications*
- *Password Flow*

La última opción es la más simple de todas, y la cual se utilizó en este proyecto. Password Flow involucra la generación de un **token de acceso**, el cual requiere alguna información brindadas por Reddit. Cuando uno quiere interaccionar con la plataforma de manera consistente, Reddit brinda la posibilidad de solicitar la autorización para poder realizar peticiones con el servidor. Esta autorización se realiza desde la página principal de Reddit, en la sección de aplicaciones (previamente, se tiene que estar registrado y conectado en una cuenta de Reddit). Esta autorización brinda acceso a diferentes funcionalidades de Reddit, como se puede apreciar en la siguiente imagen.



reddit on mobile web (installed) reddit on mobile web

Spend reddit gold creddits
Mod Note
Approve and ban users
New Modmail
Moderate Subreddit Configuration
Edit My Subscriptions
Edit structured styles
Read Wiki Pages
Wiki Editing
Vote
My Subreddits
Submit Content
Moderation Log
Moderate Posts
Moderate Flair
Save Content
Invite or remove other moderators
Post Ad Conversions
Read Content
Private Messages
Report content
My Identity
Manage live threads
Update account information
Subreddit Traffic
Edit Posts
Moderate Wiki
Make changes to your subreddit
moderator and contributor status
Manage My Flair
History

[revocar acceso](#)

Developers: [reddit](#)


Lista de funcionalidades

Una vez realizado este proceso, nos permitirá generar los datos de autenticación que son requeridos por PRAW para iniciar la conexión. Estos datos son:

- Client_id: Una cadena de 14 caracteres, indicando el uso del script para la aplicación
- Client_secret: Un identificador de 27 caracteres
- Username: El nombre de usuario de la cuenta de Reddit
- Password: La contraseña del usuario en Reddit

Los dos primeros parámetros tienen como objetivo identificar a la aplicación de forma específica, ya que un usuario puede tener distintas autorizaciones de uso para distintas aplicaciones. Los siguientes parámetros son los de conexión, que van a permitir autenticar al usuario en Reddit.

developed applications



Final project
personal use script

Computer science final project for the University of the South, Argentina.

change icon

secret [REDACTED]

name

description

about url

redirect uri

update app

delete app

developers [REDACTED] (that's you!) remove

add developer:

create another app...

Datos de configuración de la aplicación

La instalación de las librerías para usar MongoDB y InfluxDB en Python, que son PyMongo y InfluxDB, es bastante sencilla. Al igual que con la instalación de la mayoría de las librerías de Python, esto se realizará con la ejecución de los siguientes comandos:

```
python -m pip install pymongo
pip install influxdb
```

Para que estas librerías funcionen adecuadamente, los servicios relacionados con ambas bases de datos deben estar ejecutados previamente. Esto es bastante sencillo ya que, una vez instalado cada motor de base de datos, este tendrá un ejecutable que inicia el servicio que nos permite interactuar con la base de datos.

La instalación de las bases de datos de MongoDB y InfluxDB son bastante estándares y sencillas, Desde la página principal de cada uno de los motores tendremos a disposición un enlace para descargar el motor correspondiente, que consiste también en una guía tradicional de donde queremos instalar y ubicar los datos almacenados en la base de datos.

Obtención de los comentarios

Una vez tengamos los parámetros requeridos por PRAW, ya podemos arrancar a generar la conexión de nuestro código con el servidor de Reddit usando el método **praw.Reddit()**, lo que retorna una conexión con el servidor. Toda la conexión y los requerimientos al servidor se van a realizar mediante comunicación HTTPS, y la autenticación de los parámetros es realizada mediante certificados SSL.

Algo importante a tener en cuenta es que, para evitar posibles inconvenientes relacionados con un exceso de peticiones. Aunque estos valores suelen ir cambiando con el tiempo, a medida de que la infraestructura de Reddit va cambiando, suelen estar en el orden de los 1000 resultados por petición.

PRAW nos brinda distintas maneras de arrancar a interaccionar con la estructura de Reddit. Para el caso de este proyecto, la interacción es con los subreddits y tópicos más populares. La interacción con los subreddits más populares se genera utilizando la siguiente instrucción:

```
redditConection.subreddits.popular(limit = X)
```

Esto nos va a retornar una estructura con los subreddits más populares, siendo esta cantidad parametrizable. Los elementos de esta estructura son llamados **Subreddit**, que son objetos conteniendo diferentes atributos como por ejemplo el nombre del subreddit, el identificador, la fecha de creación, la cantidad de subscriptores.

A su vez, se nos brindan diferentes métodos aplicables a un subreddit, como puede la posibilidad de obtener estructuras con los tópicos más populares, más controversiales o más nuevos. En este caso, al igual que como ocurrió con los subreddits, se analizaron los tópicos más populares. La instrucción para lograr esto respeta la siguiente forma:

```
redditConection.subreddits(SubredditName).hot (limit = Y)
```

Donde, nuevamente, la cantidad de tópicos a obtener forma parte de la petición. En este caso, esta llamada nos retornara una estructura conteniendo objetos del tipo **Submission**, contando con atributos tales como el nombre del tópico, el autor, la cantidad de respuestas entre otros.

Finalmente, para obtener los comentarios de un tópico se requiere usar la siguiente instrucción:

```
redditConection.subreddit(SubredditName).Submission(SubmissionName).comments()
```

Esto nos va a retornar una instancia de un objeto llamado **CommentForest**, cuya estructura interna es un árbol con cada nivel de comentarios.

En el caso de no estar en un proceso iterativo, una alternativa a la eficiencia de las solicitudes anterior es utilizar el ID de cada subreddit y tópico, simplificando un poco la

instrucción. Como el proceso realizado en este proyecto se analizaron aproximadamente los 100 tópicos más populares de los 100 subreddits más populares, la petición y uso de los IDs de cada elemento tendía a complejizar y retrasar un poco más los tiempos de ejecución del sistema en general.

```
for subreddit in reddit.subreddits.popular(limit=cantidadSubreddits):
    hot_posts = reddit.subreddit(subreddit.display_name).hot(limit=cantidadTopics)
    for post in hot_posts:
        t = Thread(target = threaded_function, args = (post, subreddit,result ))
        t.start()
        threads.append(t)
    t = Thread(target = hiloSubreddit, args = (subreddit,result ))
    t.start()
    threads.append(t)
```

Para mejorar un poco la optimización y los tiempos de respuesta de la aplicación se intentaron dos alternativas. Por un lado se intentó manejar cada tópico en un hilo de ejecución diferente (lo que implicaba que, si se analizaban 10 tópicos de 5 subreddits, se crearán 50 hilos). Este método tenía un rendimiento adecuado siempre y cuando la cantidad de hilos a crear no superaran los límites propios del sistema operativo en el cual se estaba ejecutando la aplicación. Como no es posible crear hilos infinitos, existe un número máximo de hilos en el cual la pérdida de rendimiento arranca a ser bastante más elevada, e incluso llegando a presentarse errores a la hora de generar nuevos hilos.

La segunda alternativa, utilizada al final, fue manejar cada subreddit en un hilo diferente, pero permitiendo que el análisis de todos los tópicos pertenecientes a un mismo subreddit fuera secuencial. Esto reducía la cantidad de hilos a crear pero tenía un impacto en la forma en la que los resultados se analizaban, ya que generalmente los tópicos más populares son los que más comentarios tienen, haciendo que el análisis de los tópicos “no tan populares” se retrasara hasta que terminaran los anteriores. Aun con este problema, esta solución permitía obtener mayor cantidad de tópicos y subreddits a analizar.

El siguiente código representa la forma en la que se maneja cada subreddit por separado. Los sentimientos resultantes de cada tópico son mantenidos en una lista hasta que se terminen de analizar todos los tópicos. Luego de esto, se genera el resultado final del subreddit sumando los valores de esta lista.

```
def hiloSubreddit(subreddit, result):
    hot_posts = reddit.subreddit(subreddit.display_name).hot(limit=cantidadTopics)
    threads = []
    result = []

    for post in hot_posts:
        hiloTopico(post,subreddit, result2)
        t = Thread(target = hiloTopico, args = (post, subreddit,result ))
        t.start()
        threads.append(t)
```



```

for t in threads:
    t.join()

sentimientos = sumaEmocion(result)
result.append(sentimientos)
insertarInflux(subreddit, sentimientos)

```

El siguiente código representa la forma en la que se maneja el análisis de cada tópico por separado. Como los comentarios tienen forma de árbol, donde a medida de que un usuario responde a otro comentario se van generando nuevos niveles, se genera una cola de mensajes con los métodos *replace_more* y *comments*. De esta manera, se tiene una única estructura con todos los comentarios independientemente del nivel.

```

def hiloTopico(topic, subreddit, result):

    sentimientos2 = [0,0,0,0,0,0,0,0,0,0]
    topic.comments.replace_more(limit=None)
    comment_queue = topic.comments[:]

    while comment_queue:
        comment = comment_queue.pop(0)
        sentimientos = stringEmotions.getEmotions(comment.body)
        sentimientos2 = np.add(sentimientos2,sentimientos)
        comment_queue.extend(comment.replies)

    result.append(sentimientos2)
    insertarInflux(subreddit, topic, sentimientos2)

```

El método *getEmotions()* se encarga de, dado un comentario en formato String (*comment.body*), este elimina los caracteres innecesarios y separa las palabras. Como interesa analizar dos situaciones distintas (por un lado, las emociones de un subreddit, por otro lado las emociones de un tópico), también se genera en este caso una inserción a la base de datos con el resultado del análisis.

Análisis de los comentarios:

En este punto, ya se cuenta con una forma consistente de conectarse con el servidor de Reddit y obtener los distintos comentarios que van a ser analizados a futuro. El siguiente punto a desarrollar es el análisis de las emociones de cada comentario.

Como en este proceso se va requerir acceder a la base de datos MongoDB, es necesario tener una conexión activa previa a las consultas. La librería usada para esto es **PyMongo**, librería disponible en Python que brinda herramientas para acceder, consultar y modificar la base de datos.

Un comentario puede interpretarse como una secuencia de caracteres, donde un espacio nos permite indicar el inicio y fin de cada palabra. Sin embargo, existen diferentes caracteres que ensucian la interpretación de las palabras, como pueden ser signos de puntuación, números o caracteres especiales. Estos caracteres ocasionan que algunas secuencias no sean reconocidas por la base de datos debido caracteres extras, interpretándose entonces como una palabra nueva.

El proyecto permite no solo analizar las emociones de cada comentario de forma individual, sino también las emociones generales de todo un subreddit en particular. Esto se lleva a cabo mediante la paralización del análisis de los subreddits, donde se genera un nuevo hilo por cada subreddit. Posteriormente, cada hilo se encarga de obtener y analizar los tópicos y comentarios de ese subreddit en particular.

Previo al análisis de una palabra, es necesario analizar la cadena de caracteres en busca de caracteres especiales que no sean relevantes, como signos de puntuación, apostrofes, símbolos matemáticos, etcétera. Esto tiene varias funcionalidades:

- Los caracteres especiales y de puntuación rara vez tienen relación sentimental.
- Considerar una palabra con algún símbolo de puntuación puede dar resultados incorrectos al consultar la base de datos. La palabra "World*" y la palabra "World" estarían consideradas como diferentes sintácticamente, y en este caso la primera no forma parte de la base de datos.
- Si se considera cada variante de cada palabra y cada carácter especial, la base de datos se volvería extremadamente redundante.

Es por esto que previo a revisar cada palabra, los caracteres especiales son removidos de la cadena de caracteres. Esta forma no está libre de fallas, puesto que en parte del lenguaje de internet se suelen usar números reemplazando letras, y en este caso esas palabras no se podrían estar interpretando. Esto es uno de los problemas que conlleva la interpretación del lenguaje natural y todas las formas distintas de las que se pueden expresar cada palabra.

Esta limpieza se realiza eliminando varios caracteres de puntuación y números. El problema está que esto puede generar más problemas si no se los reemplaza

adecuadamente (un ejemplo puede ser un punto al final de una oración, Hello.World). Si solo se remueven estos caracteres, se podrían estar generando secuencias igualmente inválidas, por esta razón estos caracteres son reemplazados por un espacio. De esta manera se genera una mayor consistencia entre las palabras en el caso de que se utilicen caracteres como conectores (algo que, dependiendo el lugar de internet que se revise, suele ser común en el lenguaje informal).

Una vez realizado este reemplazo, se cuenta con un comentario conformados por letras y espacios únicamente. En este momento, con el uso de la función **Split(' ')** provista por Python, se puede generar una estructura donde cada elemento sea una palabra. De esta forma, ya vamos a disponer de cada palabra concretamente separada del resto y lista para ser consultada en la base de datos.

```
def getEmotions(string):
    string = deleteCharacters(string).lower().split(' ')
    sentimientos = [0,0,0,0,0,0,0,0,0,0]
    listaPalabrasNuevas = []
    for palabra in string:
        mydoc = mycol.find_one({"English":palabra})
        if (mydoc != None):
            mydoc = str(mydoc).split(',')
            for i in range(0,9):
                sentimientos[i]=sentimientos[i]+int(mydoc[i+2][-1])
            sentimientos[9]=sentimientos[9]+int(mydoc[11][-2])
        else:
            mycol2 = mydb["forbidden"]
            if ((mycol2.find_one({"English":palabra}) == None) and
                (len(palabra)>3)):
                listaPalabrasNuevas.append(palabra)
    insertarPalabras(listaPalabrasNuevas,sentimientos)
    return sentimientos
```

También se cuenta con una tabla de palabras “prohibidas”, palabras insertadas manualmente que ya se sabe que no poseen una emoción que las represente. Estas palabras pueden ser conectores o pronombres, palabras que no nos interesa que estén asociadas a alguna emoción en particular. Además, y como una forma de reducir la cantidad de casos diferentes, las palabras nuevas se limitan a tener como mínimo 3 caracteres.

Aunque la base de datos inicial contiene una gran variedad de palabras, existen muchas palabras que no están incluidas. En internet se usan una gran variedad de palabras características del lenguaje virtual, incluido el uso de distintos acrónimos y abreviaciones. En un primer análisis, estas palabras no aportarían significancia al resultado final del análisis para un comentario específico, sin embargo muchas de estas palabras pueden ser lo suficientemente recurrentes como para querer intentar contemplar en qué circunstancias se utilizan.

Por cada comentario, se va a contar con una estructura de 10 valores (uno por cada emoción y sentimiento presente). Estos valores van a ser numéricos, inicializados en 0, y se van a ir incrementando a medida de que el comentario sea analizado.

Al analizar un comentario, nos podemos encontrar con dos tipos distintos de palabras:

- Palabras conocidas: Aquellas cuyas emociones ya conocemos.
- Palabras a aprender: Palabras cuyas emociones desconocemos.

Al analizar un comentario en particular, vamos guardando información temporal de las emociones de las palabras conocidas. Cada vez que se consulta la base de datos y la palabra existe, se retorna una estructura con las emociones y sentimientos de esa palabra. Esta representación va a estar dada por un 0, en el caso de que esa palabra no represente la emoción/sentimiento, o un 1 indicando que si se representa. Estos valores se van a ir sumando a la estructura auxiliar donde se almacena información de las emociones generales del comentario, incrementando los valores de las emociones en función de su ocurrencia. Este proceso, a la larga, nos va a permitir identificar las emociones predominantes en un comentario, ya que van a ser las que presenten un valor numérico mayor.

Esto nos da información general del contexto del comentario, y una aproximación de las emociones que este refleja. Las palabras cuyas emociones no conocemos son separadas del resto y almacenadas en una lista temporal. Cuando ya se terminó de analizar todo un comentario, se va a tener por un lado una lista de palabras cuyas emociones desconocemos, y por otro lado las emociones generales del resto de las palabras conocidas.

Previo a la generación de las emociones en la base de datos, hay que considerar un último detalle sobre la forma que tiene la estructura de emociones.

Luego de analizar todo un tópico (o subreddit), la estructura conteniendo la información resultante del análisis tiene la siguiente forma, donde cada uno de los componentes de la estructura representa una emoción o sentimiento particular:

123	198	269	11	252	121	440	489	196	12
-----	-----	-----	----	-----	-----	-----	-----	-----	----

Aunque pudiera resultar un poco clara, ya que se puede interpretar que los valores más grandes están relacionados a una mayor aparición de esa emoción en el contexto, no resulta muy claro e incluso un poco confuso tomar conciencia de la relación entre todas las emociones. Por esta razón, se transforman estos valores a porcentajes para lograr una mejor consistencia y que resulte más fácil poder visualizar la relación entre estas emociones.

5.83	9.38	12.74	0.52	11.94	5.73	20.84	23.16	9.28	0.57
------	------	-------	------	-------	------	-------	-------	------	------

Ahora que la estructura resultante tiene un formato que resulta más fácil de interpretar, ya puede ser generada la inserción en la base de datos. Esto también simplifica el procesamiento a posterior, a la hora de visualizar, que se tendrá que hacer en Grafana.

Aprendizaje

La forma de aprendizaje de palabras se basa en el hecho de la repetición. A medida de que una palabra desconocida va apareciendo, esta se va a ir almacenando en una tabla aparte en la cual se irán incrementando los valores asociados a las distintas emociones en relación al análisis del comentario en general. De esta manera, a medida de que una palabra se vaya repitiendo, los valores emocionales asociados a esa palabra van a ir aumentando en función del contexto en el que se usó esa palabra en cada repetición.

Esta lista de palabras desconocidas, así como la estructura de las emociones del comentario, van a ser usadas en el proceso de aprendizaje de palabras. Para esto se tiene una tabla secundaria cuya funcionalidad es ir manteniendo un registro de las palabras desconocidas, su frecuencia y las emociones del contexto en el que aparecían.

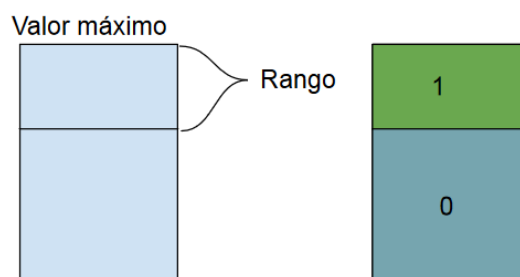
En primer lugar, cada palabra desconocida será buscada en la tabla secundaria. Hay dos posibles escenarios que pueden ocurrir a partir de esta situación:

- La palabra no estaba en la tabla: Esto nos indica una palabra nueva. En este caso se creará una nueva entrada en la tabla conteniendo la palabra junto con las emociones del contexto en la cual apareció.
- La palabra existía en la tabla: En este caso, los valores asociados a las emociones serán incrementados en función al nuevo contexto de la palabra. Así mismo, la cantidad de apariciones de la palabra también se verá incrementada.

A lo largo del tiempo, y a medida de que una palabra se fuera apareciendo repetidas veces, los valores asociados a cada emoción irían aumentando. Cuando se llega a una cantidad de repeticiones preestablecida (para este proyecto, este límite se estableció en 250), se inicia un análisis de los valores de todas sus emociones.

Primero, se define la **emoción característica**. Esta emoción va a ser la que presente el mayor valor dentro de todos los valores emocionados asociados hasta ese momento. Esta emoción se va a usar como "techo", lo que va a representar la emoción con la que más frecuencia se utiliza esa palabra.

Luego se define un **rango**, para determinar si existe alguna otra emoción que sea representativa de esa palabra. Para esto calcula un valor "piso" para el cual se va a considerar como válida la emoción.



De esta manera, se genera un rango de valores para los cuales se va a poder considerar que esa emoción particular es representativa para la palabra que se quiere aprender, mientras que cualquier valor inferior va a ser considerado como inconcluyente.

En este punto, la palabra es removida de la tabla de aprendizaje y es insertada en la tabla principal. De esta manera, y como en sistema va a seguir corriendo de forma paralela, se permite que nuevas apariciones de esa palabra sean consideradas como conocidas y pasen a formar parte del nuevo contexto del análisis de cada comentario.

Gracias a esto, se pueden aprender acrónimos y abreviaciones sin tener conocimiento real sobre su significado. Uno de los principales problemas de esta forma de aprendizaje es que es muy dependiente del contexto social en el que se estén realizando los análisis. Por ejemplo, en el caso de la situación de conflicto actual entre Rusia y Ucrania, las palabras aprendidas tendían a ser más orientadas al lenguaje cultural y político del conflicto y de ambos países. Esto genera cierto grado de sesgo para las palabras aprendidas en este contexto.

```
Aprendida palabra: serves
Aprendida palabra: forbidden
Aprendida palabra: influenced
Aprendida palabra: utter
Aprendida palabra: earning
Aprendida palabra: scheduled
Aprendida palabra: celebrate
Aprendida palabra: munitions
Aprendida palabra: pakistan
Aprendida palabra: executed
Aprendida palabra: burns
Aprendida palabra: masks
Aprendida palabra: palestinians
```

Algunas de las palabras aprendidas

Llegado a este punto, ya se tiene la recolección de datos y en análisis de las palabras, junto con el mecanismo de aprendizaje. Al finalizar en análisis de un comentario, las emociones representadas en el mismo son agrupadas en una estructura junto con las emociones del resto de los comentarios de un tópico.

Esto se hace porque, en este proyecto, se le dio más prioridad al proceso de analizar cada tópico y cada subreddit, lo que implica que el objetivo principal es identificar las emociones y sentimientos asociadas a cada tópico y subreddit.

Una vez que se cuente con todas las emociones de un tópico, se genera una nueva entrada en la base de datos InfluxDB, conteniendo el tópico analizado, el subreddit al que pertenece y más emociones asociadas a ese tópico.

Proceso de visualización

Llegados a este punto, ya se han obtenido y analizado los comentarios de distintos tópicos y subreddits, y se tiene la información a visualizar en una base de datos que está directamente conectada con Grafana.

Como se mencionó al principio de este informe, Grafana permite generar visualizaciones de forma bastante rápida y sencilla. Para esto, Grafana define lo que se conocen como **Dashboards**, que son una vista que contiene múltiples gráficos y paneles individuales organizados en forma de grilla.

Cada gráfico es, de cierta manera, independiente de los demás. Esto quiere decir que, aunque todos puedan obtener la información a visualizar de la misma fuente, los cambios a esta información que se realice en uno de ellos no van a afectar al resto. Esto es bastante útil, ya que nos asegura que cualquier cambio que se realice en uno de todos los paneles no va a causar un impacto en el resto.

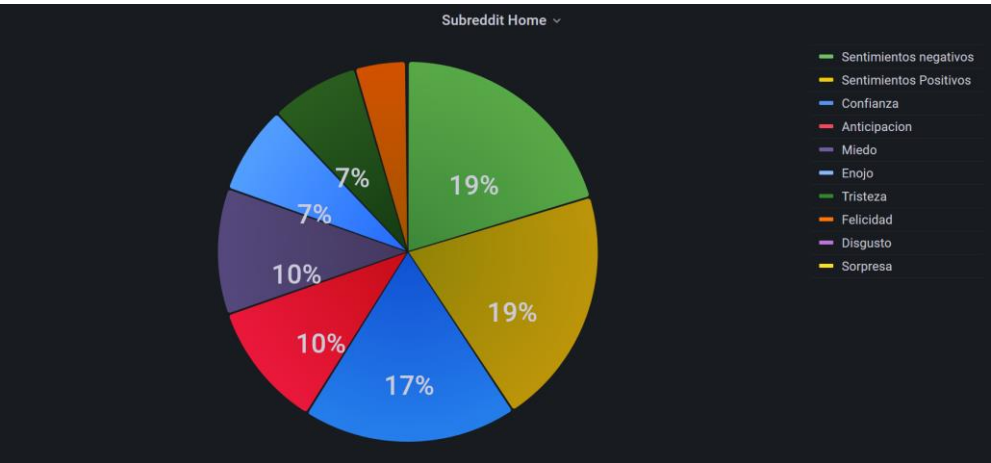
La elección del gráfico también es algo a tener en cuenta. Grafana cuenta con cientos de formatos de gráficos distintos, cada uno con opciones para configurarse y adaptarse a las necesidades del momento. Además, existe una comunidad donde es posible encontrar nuevos tipos de gráficos creados por distintas personas, que son puestos a disposición de cualquiera que desee utilizarlo en su proyecto.

Un ejemplo de como la elección puede afectar a la interpretación es el siguiente, donde se pueden apreciar tres tipos de visualizaciones distintas para el mismo conjunto de datos.



Grafana tiene la funcionalidad de, dependiendo los tipos de datos que quieras visualizar, recomendarte algunas alternativas al diseño de los gráficos. Estas alternativas no siempre se adaptan exactamente al estilo de visualización que uno quiere. La imagen anterior es un ejemplo de una visualización muy usada para mostrar métricas de desempeño de algunos sistemas (por ejemplo, para mostrar temperatura, humedad, tiempo de uso de un sistema, entre otros).

Este tipo de gráficos, aunque permite ver los valores asociados a cada emoción y sentimiento, tiene el problema de que no es fácil ver la proporción entre estos valores de una forma mucho más directa. Aunque no puede resultar muy difícil comparar los valores numéricos, esto impone otra tarea a realizar (consiente o subconscientemente) por la persona que está viendo la información. Una alternativa a esto es usar un gráfico estilo torta como el siguiente.



Emociones de un subreddit específico

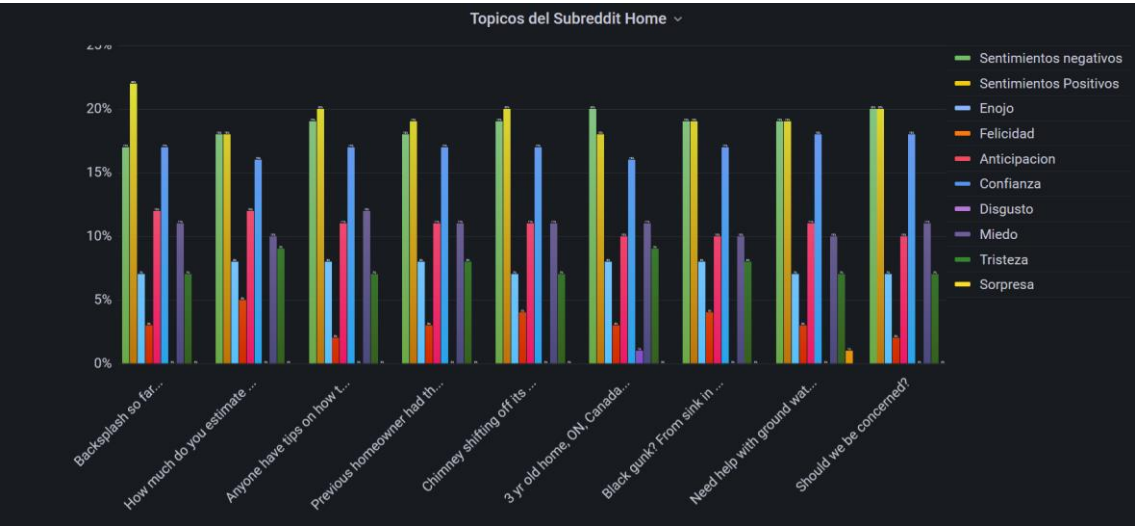
En este tipo de gráficos, aparte de estar viendo los porcentajes como ocurría en el modelo anterior, el hecho de que estos valores estén mapeados visualmente al tamaño de cada porción del grafico facilita ver la relación entre las implicancias de cada emoción en el subreddit específico.

El grafico de tipo torta va a ser utilizado específicamente para visualizar las emociones en un determinado subreddit. Esto permite analizar cada subreddit por separado del resto, ya que al tratarse de temas diferentes puede no existir alguna relación entre ellos.



Muestra de seis de los subreddits más populares en el momento de realizar la prueba.

Por otro lado se utilizara un gráfico de barras para mostrar varios tópicos relacionados con un mismo subreddit. Este tipo de gráficos permite agrupar todas las emociones de un mismo tópico en un grupo y visualizarlo en conjunto con los otros tópicos del mismo subreddit. De esta manera se logra ver cómo, entre tema y tema, las emociones pueden ir variando.



Visualización de las emociones de distintos tópicos de un mismo subreddit

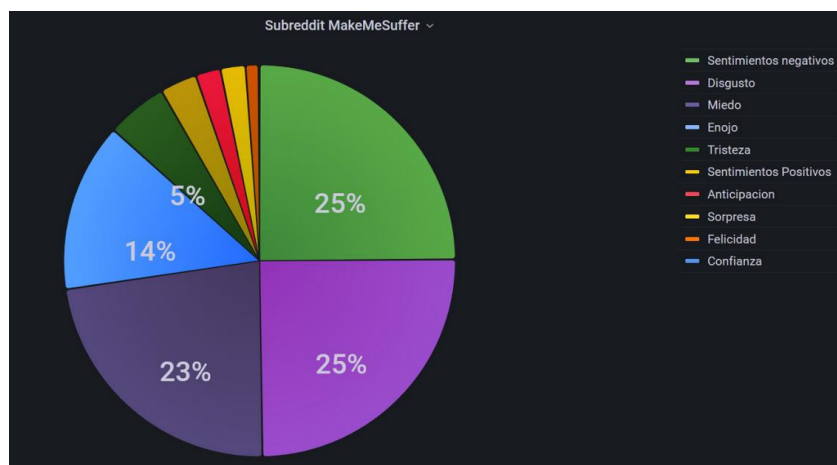
De esta manera, la visualización de las emociones de subreddit tiene dos formas. Por un lado la visualización por subreddit, donde se puede apreciar una media de emociones generalizadas entre los tópicos más populares, y por otro lado las emociones de cada tópico en particular. Los gráficos consecuentes con esto son los siguientes.



Emociones de los tópicos más populares de seis distintos subreddits

Conclusiones

Uno de los aspectos más interesantes que se pudo apreciar de este análisis es la consistencia entre las emociones de los subreddits más populares. Como esta popularidad es definida como la cantidad de usuarios activos que tiene cada comunidad, y considerando el hecho de que Reddit es una plataforma donde acceden cientos de personas de todas las edades, los subreddits más populares tienden a tener una igual cantidad de sentimientos negativos y positivos, con una pequeña tendencia hacia los positivos. Las emociones más frecuentes en los subreddits más populares eran principalmente felicidad, confianza, anticipación. Esto generó cierta curiosidad, ya que los resultados entre los distintos subreddits eran demasiado consistentes entre ellos.



MakeMeSuffer es un subreddit orientado a mostrar heridas o enfermedades de una forma muy visual

Sin embargo, si se analizan subreddits menos populares como puede ser “MakeMeSuffer”, vemos que esta tendencia cambia. Los subreddits que tienden a presentar sentimientos negativos y emociones como miedo, enojo o disgusto tienden a no tener una popularidad tan elevada. Sin embargo, Reddit no ofrece una forma de obtener de forma automática los subreddits menos populares (como existen cientos de comunidades, muchas de ellas están prácticamente olvidadas), por esta razón para realizar este tipo de análisis fue necesario indicarle concretamente los subreddits a analizar a la aplicación.

Esto también puede deberse a la forma de moderación de Reddit, donde comentarios que resulten muy ofensivos o inapropiados son eliminados por los moderadores. De esta manera, se genera un filtro extra a las emociones que se podrían llegar al detectar, debido a que varias de las palabras con connotaciones negativas podrían estar siendo eliminadas.

Por otro lado, una de las pruebas realizadas consistió en análisis de los 100 tópicos más populares dentro de los 100 subreddits más populares. Luego del análisis de los 10.000 tópicos, consistiendo en cerca de 4000 millones de palabras, el sistema había aprendido aproximadamente entre 2500 y 3000 palabras nuevas. Como el sistema no está pensado

para que pueda interpretar las palabras aprendidas por fuera de lo relacionado con las emociones del contexto de su uso, muchas de las palabras aprendidas fueron nombre de personas (como Putin o Biden) o bien palabras como rusos, inglés, celebridades, mascara o acrónimos del estilo OTAM, GTFO, ILVL entre otros.

Futuras mejoras del sistema

Una de más optimizaciones que se puede realizar al sistema sería el uso del plugin de MongoDB para Grafana, consiguiendo la licencia. De esta manera ya no sería necesaria la utilización de un segundo motor de base de datos (InfluxDB) para la visualización.

También se podría mejorar el control de las emociones para las palabras ya aprendidas. Como el algoritmo de aprendizaje de basa principalmente en las emociones presentadas por las palabras conocidas de cada mensaje, si las emociones relacionadas con estas palabras (ya sea las que inicialmente se encuentran en la tabla NRC o las que se aprende con el tiempo) presentan algún tipo de sesgo con alguna emoción particular, entonces esto va a afectar al resultado final de todo el sistema.

Una alternativa sería la implementación de un algoritmo que vaya revisando que las palabras aprendidas aparezcan en los contextos que representan. Esto no silo implicaría estar realizando un análisis por cada palabra que se está queriendo aprender, sino también otro sistema que se encargue de buscar y analizar el contexto de las palabras en la tabla principal y modificarlas en caso de ser necesario.

Por último, dada la funcionalidad que pueden presentar cierto tipo de gráficos en Grafana para trabajar con el tiempo, se podría implementar una visualización en tiempo real de los tópicos. La aplicación de Reddit PRAW brinda, entre otros datos, la posibilidad de obtener la fecha en la que se realizó el último posteo en un determinado tópico. Con esta fecha se podría determinar si, en un lapso de tiempo, ocurrieron nuevos comentarios en un tópico ya analizado. También, para que pueda ser vista en tiempo real, se requeriría reducir el tiempo de procesamiento, ya sea implementando mejoras en el código o bien a través de un hardware más rápido.

Referencias

- [1] NRC Word-Emotion Association Lexicon -
<https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>
- [2] PRAW: The Python Reddit API Wrapper - *<https://praw.readthedocs.io/en/stable/>*
- [3] Sanjeev Dhawan, Kulvinder Singh, Deepika Sehrawat - Emotion Mining Techniques in Social Networking Sites
- [4] Francisca Adoma Acheampong, Henry Nunoo-Mensah, Chen Wenyu-Text-based emotion detection: Advances, challenges, and opportunities
- [5] MongoDB Community Server -
<https://www.mongodb.com/try/download/community>