TRABAJO DE FIN DE GRADO

DINÁMICA SOCIAL EN YOUTUBE: ANÁLISIS DEL DISCURSO EN LÍNEA

Detección de tendencias en YouTube

Autora: *Pilar Medina Rodríguez - 49249174X*
pilar.medinar@um.es

Tutores: *Rafael Valencia García (valencia@um.es) y*
José Antonio García Díaz (joseantonio.garcia8@um.es)

Facultad de Informática

GRADO EN INGENIERÍA INFORMÁTICA

DIS Departamento de Informática y Sistemas

UNIVERSIDAD DE MURCIA

Murcia, 3 de Junio de 2024

Agradecimientos

Quiero dar las gracias a mis amigas. A Lucía, a María, a Inma y a Sandra, por estar siempre presentes hasta en la distancia, por hacerme sentir acompañada para todo, por las risas y los llantos compartidos a lo largo de los años. Sois lo que significa la palabra *amistad*.

También quiero dedicar unas palabras a Jesús, mi compañero incondicional en este viaje. Gracias por tu amor, tu ayuda y tu apoyo inquebrantable en todas mis batallas. Por estar a mi lado en lo bueno y en lo malo, y por confiar en mí desde el primer día y hasta el último. Eres lo mejor que me llevo de la facultad.

Y, por último, quiero agradecer a mi familia, que me han apoyado siempre en los mejores y peores momentos a lo largo de mi vida, y especialmente, en mis años de universidad. Gracias por inspirarme siempre. A la ambición imparable de mi hermano, a la admirable disciplina de mi hermana, y a mis padres, por el gran trabajo de toda una vida dedicado a su gran proyecto: hacer de nosotros los mejores hijos; vuestro empeño está dando sus frutos.

Sin vosotros no sé cuántos años más habría tardado en sacarme la carrera.

Este logro también es vuestro.

Engineers like to solve problems,
If there are no problems handily available,
they will create their own problems.

Scott Adams.

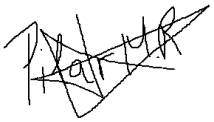
Declaración de originalidad

Dña. Pilar Medina Rodríguez, con DNI 49249174X, estudiante de la titulación de Grado en Ingeniería Informática de la Universidad de Murcia y autora del TFG titulado *"Detección de tendencias en YouTube"*. De acuerdo con el Reglamento por el que se regulan los Trabajos Fin de Grado y de Fin de Máster en la Universidad de Murcia (aprobado C. de Gob. 30-04-2015, modificado 22-04-2016 y 28-09-2018), así como la normativa interna para la oferta, asignación, elaboración y defensa de los Trabajos Fin de Grado y Fin de Máster de las titulaciones impartidas en la Facultad de Informática de la Universidad de Murcia (aprobada en Junta de Facultad 27-11-2015).

Declaro:

Que el Trabajo Fin de Grado presentado para su evaluación es original y de elaboración personal. Todas las fuentes utilizadas han sido debidamente citadas. Así mismo, declara que no incumple ningún contrato de confidencialidad, ni viola ningún derecho de propiedad intelectual e industrial.

Murcia, a 2 de junio de 2024.



Fdo., Pilar Medina Rodríguez
Autora del TFG

Resumen

El presente trabajo de fin de grado aborda el desarrollo de una aplicación para la recolección, análisis y visualización de datos en tiempo real de contenido multimedia y comentarios asociados en plataformas en línea, en este caso concreto, en YouTube. El objetivo principal es proporcionar una herramienta integral para comprender la dinámica de discusión en torno a diversos temas o tópicos de interés, aprovechando técnicas de Procesamiento del Lenguaje Natural (PLN) y herramientas de desarrollo de aplicaciones Web.

El trabajo se estructura en torno a cuatro grandes componentes. El primero de ellos es el módulo de recolección de datos, en el cual se implementa un sistema capaz de recolectar datos de diferentes fuentes en línea, con un enfoque esencial en la API de YouTube para obtener vídeos relevantes y comentarios asociados a un tema específico.

El segundo módulo es el de análisis de texto, en el que se emplean modelos de PLN para realizar un análisis de los comentarios recolectados. Esto incluye técnicas como la clasificación de sentimientos o la identificación de diferentes temas, con el objetivo de poner el foco en las tendencias y opiniones expresadas en la discusión en línea.

El tercer módulo es el cuadro de mando, en el que se construye una interfaz web interactiva utilizando un framework como Flask, permitiendo a los usuarios explorar y visualizar los datos recogidos de manera intuitiva, incluyendo gráficos y tablas que muestran información relevante sobre el tema seleccionado.

Y, por último, el presente documento, que describe en detalle el proceso de desarrollo, las decisiones de diseño tomadas y los resultados obtenidos.

Extended abstract

The advent of digital media and online platforms has revolutionized the way information is disseminated and consumed. Among these platforms, YouTube stands out as a major source of multimedia content and a vibrant community where users actively engage through comments. This engagement generates a vast amount of data that can be analyzed to extract insights about public opinion, emerging trends, and sentiment towards various topics. According to statistics, more than 500 hours of video content are uploaded to YouTube every minute, and the platform has over 2 billion logged-in monthly users. This immense volume of content and user interaction presents both an opportunity and a challenge for data analysis.

The primary objective of this work is to develop an application that facilitates real-time data collection, analysis, and visualization of multimedia content and associated comments on YouTube. By leveraging advanced Natural Language Processing (NLP) techniques and web application development tools, this project aims to provide a comprehensive tool for understanding the dynamics of online discussions. Specific objectives include: (1) developing a robust pipeline for extracting data (videos and comments) from YouTube, and (2) using advanced NLP models to analyze the sentiment of the comments extracted and creating interactive visualizations to present the data in an intuitive manner to give a good experience to any user of the application.

The motivation behind this project is rooted in the growing importance of social media analytics in various fields such as marketing, political science, and social research. Understanding public sentiment and identifying trends can provide valuable insights for businesses, policymakers, and researchers. Moreover, the rapid growth of online platforms and the increasing amount of user-generated content necessitate the development of tools that can efficiently process and analyze large datasets. The ability to analyze and visualize this data in real-time can significantly enhance decision-making processes and strategic planning.

This project represents an intersection of my interests in Artificial Intelligence (AI), data analysis, and web development. The challenge of creating a tool that not only gathers and analyzes

data but also presents it in an intuitive and accessible manner to the user is both intriguing and rewarding. And although my academic background and previous experiences may or may not have equipped me with the skills necessary to tackle this multifaceted project, I am eager to contribute to the field of social media analytics in the best way possible.

The subjects of this thesis are as follows:

- NLP is a field of AI that focuses on the interaction between computers and human language. It encompasses various techniques and models designed to process and analyze large amounts of natural language data. Recent advancements in NLP, particularly the development of deep learning models such as BERT [1], have significantly improved the accuracy and efficiency of text analysis tasks.

- Sentiment Analysis (SA), also known as opinion mining, involves determining the sentiment expressed in a piece of text. This can range from simple polarity classification (positive, negative, neutral) to more complex emotional classifications (joy, anger, sadness, etc.). Currently, there are a lot of tools that focus on this field, Vader and TextBlob are examples of those we've investigated in this thesis.

- Data visualization involves the graphical representation of data to facilitate understanding and insight extraction. Tools like Tableau and libraries such as Dash and Chart.js are widely used for creating interactive and dynamic visualizations. Effective visualization helps in identifying patterns, trends, and anomalies in the data.

To complete these subjects, the first step is to establish a reliable method for collecting data from YouTube. This involved using the YouTube Data API to retrieve information about videos, comments, and user interactions. The API provides a comprehensive set of endpoints for accessing various types of data, including video metadata, comment threads, and user profiles. One of the challenges encountered during this phase was managing the rate limits imposed by the API. To address this, a rate-limiting mechanism was implemented to ensure that data collection could proceed smoothly without exceeding the allowed quotas.

Once the data is collected, the next step is to process it for the analysis. This involved tokenization (breaking down text into individual words or tokens) and stopwords removal (removing common words that do not contribute significantly to the meaning of the text). For these, we used libraries such as NLTK, which provide efficient and reliable tools for natural language processing. The preprocessing steps were crucial for preparing the data for NLP tasks.

For SA, we implemented a SA fine-tuned model based on BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art NLP model developed by Google. BERT has significantly improved the accuracy and efficiency of text analysis tasks due to its ability to understand context and nuances in language. This model was trained on a large dataset of labeled comments to learn the patterns and features associated with different sentiments. During the evaluation phase, the model achieved a high level of accuracy, correctly classifying the sentiment of most comments. However, it was observed that the performance could be affected by the quality

and variability of the comments. To mitigate this, additional preprocessing steps were introduced to handle noisy and inconsistent data.

Another tool that stands out is Flask, which is a microframework for web development in Python that was used in this project to build the user interface for the data visualization system. Flask was chosen for its simplicity and flexibility, allowing developers to create robust and scalable web applications with relatively little effort. In this project, Flask facilitates the creation of an interactive dashboard that enables users to intuitively explore and visualize the data collected from YouTube. Its modular design allowed for seamless integration with other libraries and tools used in data analysis and natural language processing. Furthermore, its support for extensions like Flask-RESTful simplified the creation of RESTful APIs, enhancing communication between the frontend and backend of the system. In conclusion, we can say the Flask framework played a crucial role in the project's implementation by providing the necessary infrastructure for an effective user interface and enabling the integration of multiple system components.

And last but not least, the data visualization subject, which involves the graphical representation of data to facilitate understanding and insight extraction. For this project, we chose to use library Chart.js to create interactive and dynamic visualizations. The visualizations were integrated into a web-based dashboard, allowing users to explore and interact with the data in real-time. The dashboard includes several components to show the sentiment overview and other elements of interest for the study: a bar chart showing the distribution of sentiments (positive, neutral, negative) across all comments, a pie chart that shows the total of comments for each type of sentiment applying the date filter the user chooses, and a bar chart displaying the most frequently used words on the comments registered in the database. These visualizations help users identify patterns, trends, and anomalies in the data, providing valuable insights into the dynamics of online discussions.

The results of the sentiment analysis showed that the distribution of sentiments varied across different videos and topics, reflecting the diverse nature of discussions on YouTube. For instance, videos related to entertainment and lifestyle tended to have more positive or neutral comments, while those related to controversial topics such as politics and social issues had a higher proportion of negative comments and just a little of neutral ones, showing that the discourse of online users on these topics is much more polarized.

The word frequency analysis revealed interesting patterns in the language used by YouTube users. Common words included on the topic analyze (*machismo*: sexism) were nouns such as *mujer* (woman) or *hombre* (man), and the most used verb is *ser* (to be). This fact can reveal that there are certain keywords associated with specific topics or events, indicating their relevance to the ongoing discussions.

Analyzing the data over time provided insights into how discussions evolved. Peaks in sentiment and word frequency often corresponded with significant events or the release of new content. For example, in our case, a major political announcement could result in spikes in both the volume and sentiment of comments. And although this study is not on a very large scale, we

could see that temporal trends highlight the dynamic nature of online discussions and the importance of real-time analysis.

One of the primary applications of this project could be in the field of marketing. Businesses can use the tool to monitor public sentiment towards their products or services, identify emerging trends, and respond to customer feedback in real-time. For instance, a company launching a new product can track the sentiment of comments on promotional videos to gauge the initial reception and make data-driven decisions about marketing strategies. Also, researchers in political science and social studies can use the tool to analyze public opinion on various issues. By examining the sentiment and language used in comments on videos related to political events, researchers can gain insights into public attitudes and behaviors. This information can be valuable for understanding voter sentiment, predicting election outcomes, and studying the impact of social media on political discourse.

Content creators and educators can also benefit from this project. By analyzing the feedback on their videos, they can identify areas for improvement and tailor their content to better meet the needs and preferences of their audience. Additionally, educators can use the tool to analyze discussions on educational videos, gaining insights into student engagement and understanding.

This way, we can see that the developed system successfully collects, analyzes, and visualizes data from YouTube, providing valuable insights into online discussions. The SA model achieved a high level of accuracy, correctly classifying the sentiment of most comments. The interactive dashboard proved to be a useful tool for exploring and visualizing the data. However, there were some limitations and challenges encountered during the project. The performance of the sentiment analysis model could be affected by the quality and variability of the comments. Additionally, the real-time updating of the dashboard required careful management of data processing and API usage.

As we've already seen in the above results of our case study, although the current SA model performs well, there is always room for improvement. Future work could focus on incorporating more advanced NLP techniques and models to enhance accuracy and robustness. For example, fine-tuning the BERT model on domain-specific datasets or exploring other more advanced transformer-based models could yield better results. Additionally, integrating more diverse linguistic features and expanding the training dataset to include more varied and representative samples could further improve the model's performance and its ability to generalize across different contexts and user expressions.

Another aspect to keep in mind is that the current system is limited to YouTube, but the methodology can be extended or replicated to other social media platforms such as Twitter, Facebook, and Instagram if you want to explore other social networks for this same purpose. This would provide a more comprehensive view of online discussions and allow for cross-platform analysis. Integrating data from multiple sources could also improve the robustness of the analysis

and provide deeper insights. This would enable a more holistic understanding of user sentiments and opinions across different online communities and platforms, allowing for a more nuanced analysis of social media trends and public discourse.

On this matter, we can see that real-time data processing and visualization are crucial for timely decision-making. Future work could focus on optimizing the system to handle real-time data streams more efficiently. This could involve using distributed computing frameworks to process and analyze large volumes of data in real-time. Implementing more sophisticated data pipeline architectures and leveraging cloud-based solutions for scalability could significantly enhance the system's performance and its ability to provide timely insights.

Another point that could be improved on the created software is improving the user interface and user experience of the dashboard. Adding more interactive features, customizable views, and advanced filtering options could make the tool more user-friendly and versatile. User feedback and usability testing could guide these enhancements in the future, ensuring that the dashboard meets the needs and expectations of its users and provides an intuitive and efficient way to explore and interpret the data.

In summary, this thesis demonstrates the effectiveness of a system for real-time data collection, analysis, and visualization of multimedia content and associated comments on YouTube. By leveraging NLP techniques and web application development tools, the project provides a comprehensive tool for understanding the dynamics of online discussions. The developed system not only facilitates the identification of trends, sentiment, and key themes within user-generated content but also allows for the detailed examination of how these elements evolve over time. This capability is particularly valuable for tracking changes in public opinion and identifying emerging topics and sentiments.

In conclusion, we highlight that the developed software lays a solid foundation for a powerful analytical tool with wide-ranging applications, and its continued development holds significant promise for both academic research and practical applications in various industries.

Índice

Índice de figuras

1. Introducción

El creciente predominio de las plataformas en línea como fuentes de información y debate ha provocado un interés creciente en comprender y analizar el contenido compartido dentro de estos espacios digitales, ya que proporcionan insights y conocimientos que respaldan la toma de decisiones en diferentes ámbitos, como el empresarial, el académico o el político. Todo esto se debe a la ayuda que proporciona a la hora de comprender a la audiencia en sus interacciones y comentarios, lo que puede ser valioso para diseñar estrategias de marketing, mejorar servicios o adaptar contenido a las necesidades que se observen en ellos.

Entre las diferentes plataformas que existen, YouTube destaca como un importante centro de contenido multimedia y participación de usuarios, lo que la convierte en un recurso valioso para estudiar el discurso en línea. Este proyecto ha sido desarrollado con la finalidad de crear un sistema que fuese capaz de obtener ciertos conocimientos e información procedentes de la API de YouTube a partir del texto extraído de los comentarios de vídeos de la plataforma sobre un tema concreto.

1. 1. Motivación

La motivación detrás de esta investigación surge de la necesidad de comprender la dinámica de las interacciones dentro de las comunidades en línea, particularmente en plataformas como YouTube. Entender cómo los usuarios se relacionan con el contenido, expresan opiniones e interactúan entre sí puede proporcionar información sobre las tendencias sociales, sentimientos que los usuarios quieren hacer públicos y temas de interés emergentes.

Mediante el aprovechamiento del análisis impulsado por datos y técnicas de procesamiento del lenguaje natural, se pondrá el ojo en descubrir patrones, analizar sentimientos y focalizar la atención en un tópico concreto presente en gran cantidad de contenido multimedia y sus correspondientes comentarios en YouTube.

1. 2. Objetivo

La línea de propuesta de este trabajo es la detección de tendencias en YouTube, sobre la que llevar a cabo el desarrollo de un software que recoge el discurso en línea de los usuarios de la red social YouTube sobre un tópico determinado para crear un dashboard tras realizar un análisis del mismo.

Como parte de este análisis, el software desarrollado debe clasificar el texto según ciertos criterios usando como filtro un tópico concreto. La clasificación de texto que se ha decidido llevar a cabo es la de sentimientos, indicando la polaridad del texto, es decir, si es positivo, neutral o negativo. Por otra parte, también habrá que recabar las palabras más usadas por los usuarios a la hora de expresar su opinión en los vídeos dentro del tema escogido.

Por la naturaleza de la propia plataforma de YouTube, el medio de expresión de los usuarios son los comentarios que pueden dejar en los vídeos de la plataforma de forma voluntaria, los cuales serán los objetos principales del proceso de análisis.

A partir de los datos recogidos y analizados, se creará un cuadro de mando que permita visualizar los vídeos y comentarios recolectados por fechas y se mostrarán gráficas para reflejar el estudio realizado según el caso de estudio.

Los objetivos específicos de este proyecto incluyen:

- Diseñar e implementar un sistema robusto de recopilación de datos capaz de recuperar vídeos relevantes sobre un tema concreto y comentarios asociados a ellos obtenidos de la API de YouTube.
- Utilizar técnicas de PLN para analizar sentimientos relacionados con este tópico expresados en los contenidos recopilados.
- Desarrollar un panel de control para ofrecer la posibilidad de visualizar los datos previamente analizados de una manera amigable para el usuario, permitiéndoles explorar y comprender la dinámica del discurso en línea acerca del tema escogido.
- Evaluar los diferentes resultados que ofrece la aplicación desarrollada que engloba los objetivos anteriores con el objetivo de validar su utilidad como herramienta para comprender las discusiones en línea en YouTube y ver si consigue satisfacer la motivación de la misma.

2. Estado del arte

El auge y la popularidad que han conseguido las plataformas digitales en la actualidad han dado lugar a un amplio conjunto de datos textuales generados por los usuarios en línea, lo cual ha llevado a que el estudio de sus discursos e interacciones en este ámbito haya sido un área de investigación en constante evolución en los últimos años.

Los datos que son generados por los usuarios en la red abarcan una gran variedad de géneros discursivos. Algunos ejemplos de esto son los comentarios en redes sociales, las opiniones que dejan en reseñas de productos, las discusiones en foros, o distintas publicaciones en blogs. Este es uno de los motivos por los que surge el estudio enfocado al análisis de las interacciones en línea, que es crucial para comprender mejor cómo se comunican, interactúan y expresan los usuarios en el entorno digital. Del mismo modo ocurre con la representación gráfica de los análisis que se realizan, que es vital para obtener un recurso visual que determine de forma mucho más directa cuál es la información que se obtiene de ellos.

A continuación, se presentan algunos de los aspectos clave del estudio del discurso en línea y los avances de los últimos años en los campos sobre los que se va a trabajar posteriormente. De la misma forma se exponen algunos ejemplos de herramientas que han sido creadas con el propósito de desempeñar algunas de las diferentes funciones relacionadas con ellos.

2. 1. Recopilación de datos en plataformas

La recopilación de datos en plataformas en línea es el proceso de recoger información de diversas fuentes disponibles en internet, como pueden ser datos de redes sociales, sitios web, foros, blogs o plataformas de comercio electrónico [2]. La recopilación de datos en línea es una práctica común en diversos campos, incluyendo la investigación académica, el análisis de mercado, la

inteligencia empresarial y el desarrollo de aplicaciones y servicios en línea. Existen varios elementos que intervienen a la hora de realizar esta función que se desarrollan a continuación:

2. 1. 1. Web Scraping

El *web scraping* [3] es una técnica que consiste en extraer información de páginas web de manera sistemática y programática. Los investigadores y desarrolladores utilizan scripts y herramientas especializadas para realizar esta tarea, que implica enviar solicitudes HTTP a la página web objetivo, analizar el contenido HTML de la respuesta y extraer los datos relevantes utilizando patrones de búsqueda o selectores específicos. Debe realizarse de forma que se acaten los términos de servicio de la plataforma objetivo, que además pueden implementar medidas de seguridad para evitar el acceso no autorizado a sus datos, lo que puede complicar el proceso.

En plataformas en línea como YouTube, el web scraping se utiliza para extraer metadatos de vídeos, comentarios, listas de reproducción, estadísticas de visualización y otros elementos disponibles públicamente en la interfaz de usuario de la plataforma. Esto permite a los investigadores recopilar grandes volúmenes de datos de manera eficiente y automatizada, lo que es fundamental para poder realizar los distintos análisis que sean requeridos dentro de la aplicación.

2. 1. 2. Acceso a APIs

Algunas plataformas en línea ofrecen interfaces de programación de aplicaciones (API) que permiten a los desarrolladores acceder a sus datos de manera estructurada y programática. Estas APIs proporcionan endpoints (URLs) y métodos de solicitud que permiten recuperar información específica de la plataforma de forma eficiente y confiable.

El uso de APIs tiene varias ventajas sobre el web scraping, incluida una estructura de datos más consistente y fácil de manejar, un acceso más rápido a la información y un soporte oficial proporcionado por el proveedor de la plataforma. Sin embargo, el acceso a APIs puede estar sujeto a límites de uso y restricciones de acceso, y es posible que algunos datos no estén disponibles a través de la API pública.

En el caso de YouTube, Google ofrece la *YouTube Data API* [4], que permite a los desarrolladores acceder a una amplia gama de datos y funcionalidades de la plataforma, incluidos vídeos, comentarios, listas de reproducción, estadísticas y más. Los desarrolladores pueden enviar solicitudes HTTP a la API utilizando diferentes parámetros de consulta para recuperar datos específicos según sus necesidades, lo cual facilita la tarea de recopilación de datos.

2. 1. 3. PLN

El PLN [5] es una rama de la IA y la lingüística computacional que se enfoca en la interacción entre las computadoras y el lenguaje humano natural. Su objetivo es permitir a las computadoras comprender, interpretar y generar lenguaje humano de manera efectiva. El PLN abarca una variedad de tareas y técnicas que son de suma importancia en la recopilación y análisis de datos en plataformas en línea.

Una vez recopilados los datos, las herramientas de PLN son las que entran en juego para realizar análisis avanzados. Entre las distintas aplicaciones destacan la traducción automática a diferentes idiomas o los algoritmos de generación de resúmenes automáticos, que son útiles para condensar grandes volúmenes de información en un formato más conciso para agilizar la comprensión de tendencias emergentes en los datos recopilados.

Otra aplicación relevante es la clasificación de textos, que permite etiquetar documentos según su contenido. Esto es útil en la categorización de comentarios o publicaciones en redes sociales en diferentes grupos acerca de un tema. Esta función también puede utilizarse para identificar contenido inapropiado o nocivo dentro de las plataformas, como la detección de discurso de odio, spam, o cualquier otro tipo de violación de las políticas de la plataforma.

Es por esto que los algoritmos de PLN pueden ayudar a mejorar los sistemas de recomendación, haciendo posible personalizar las sugerencias de contenido para cada usuario y mejorar su experiencia, además de poder automatizar funciones que optimicen el funcionamiento dentro de la propia plataforma.

2. 2. Análisis de sentimientos y opiniones

Los avances en técnicas de procesamiento del lenguaje natural han permitido desarrollar modelos cada vez más precisos que ofrecen la posibilidad de analizar y comprender el tono emocional de los comentarios en línea. El *análisis de sentimientos* [6] se refiere a la identificación y clasificación de las emociones expresadas en el texto, comúnmente según su polaridad, es decir, como positivas, negativas o neutrales.

Además de la expresión de sentimientos, el estudio del discurso en línea también aborda la identificación de opiniones y actitudes hacia temas específicos, que incluye técnicas como la minería de opiniones, la cual busca identificar las posturas o evaluaciones expresadas por los usuarios sobre productos, servicios o eventos.

El proceso típico de análisis de sentimientos sigue un pipeline que incluye varios pasos clave: (1) se recolectan grandes cantidades de datos textuales de las fuentes correspondientes; (2) los datos se limpian y preparan mediante técnicas como eliminación de ruido, tokenización, eliminación de stopwords y lematización; (3) se extraen características relevantes del texto, como n-gramas y embeddings; (4) se entrenan modelos de *machine learning* o *deep learning* utilizando las

características extraídas; (5) se evalúa el rendimiento del modelo utilizando métricas como precisión, recall y F1-score; y (6) el modelo entrenado se utiliza para predecir los sentimientos en nuevos datos textuales.

Entre las tecnologías actuales destacan los transformers, especialmente los modelos basados en arquitecturas como BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer) [7], y RoBERTa [8], siendo las más avanzadas en el campo del SA. Estos modelos han demostrado una capacidad superior para comprender el contexto y la semántica del lenguaje, lo que mejora significativamente la precisión en la clasificación de sentimientos.

Antes de la popularización de los transformers, el SA se basaba en técnicas más simples como Bag of Words (BoW) [9], que representa los textos como una colección de palabras sin tener en cuenta el orden; TF-IDF (Term Frequency-Inverse Document Frequency) [10], técnica para evaluar la importancia de una palabra en un documento dentro de un corpus; o en modelos de machine learning clásicos como SVM [11] y Naive Bayes [12], que se entrenaban sobre características extraídas manualmente del texto.

Aunque ya se han mencionado algunos de entre todos los posibles usos para los que se pueden emplear estas herramientas, también destacan los siguientes:

- Las empresas pueden identificar rápidamente las quejas o los elogios de los clientes sobre un producto, un servicio o la propia entidad según el tono emocional de sus comentarios al respecto en redes. Esto lo pueden hacer con el fin de gestionar la satisfacción de los mismos, la reputación que tienen entre ellos, identificar potenciales problemas, áreas de mejora o hacer comparaciones con la competencia.

- Los equipos de campañas políticas y los analistas de este sector pueden evaluar la opinión pública sobre distintos temas y candidatos políticos en las publicaciones de los usuarios en línea. De esta forma, pueden identificar sus preocupaciones, evaluar el apoyo a diferentes políticas y anticipar posibles problemas de percepción pública.

- Los inversores y analistas financieros pueden evaluar el sentimiento del mercado y predecir movimientos futuros en los precios de las acciones, divisas y otros activos financieros para encontrar posibles oportunidades de inversión o riesgos potenciales, analizando para ello el tono de las noticias financieras, los informes de analistas y los comentarios al respecto en redes sociales.

Sin embargo, las herramientas de análisis de sentimientos tienen también algunas limitaciones. Entre ellas, la principal es que el lenguaje es inherentemente ambiguo y puede contener expresiones difíciles de interpretar correctamente incluso para los humanos, lo que acarrea errores en la clasificación del sentimiento, especialmente en casos de ironía, sarcasmo o doble sentido. Además, existen varios factores que dificultan la generalización de los resultados, como las diferencias lingüísticas en la expresión de emociones, que pueden interpretarse de manera distinta en otras culturas o lenguas, o el que no se tiene en cuenta el contexto en el que se

escribe un mensaje, haciendo que la interpretación sea subjetiva y pueda variar entre analistas o algoritmos.

Asimismo, de forma común, este proceso implica el procesamiento de grandes cantidades de datos de texto, lo que plantea preocupaciones sobre la privacidad y la ética en el uso de la información personal de los usuarios. Esto es algo a tener en cuenta hoy día, ya que es importante garantizar que se respeten los derechos de privacidad y se cumplan las regulaciones aplicables al manejar datos sensibles para cumplir satisfactoriamente con la protección de los mismos.

2. 2. 1. Creación de aplicaciones web

La visualización de los datos en tiempo real juega un papel fundamental en la comprensión y el análisis de la información en un mundo cada vez más impulsado por los datos. En la actualidad, existen numerosas herramientas que permiten presentar de forma clara y concisa información clave, convirtiéndose en una parte fundamental en la toma de decisiones basada en datos.

Los paneles de control interactivos ofrecen a los usuarios la capacidad de explorar y analizar datos en tiempo real de forma dinámica. Proporcionan gráficos interactivos, tablas y filtros que permiten a los usuarios personalizar su análisis según necesidades específicas, para hacer descubrimientos personalizados.

Las visualizaciones dinámicas son representaciones visuales de datos que cambian y se actualizan en tiempo real a medida que se reciben nuevos datos. Incluyen: (1) mapas de calor para visualizar la distribución espacial de los datos y detectar áreas de alta densidad; (2) gráficos de redes para representar relaciones y conexiones entre diferentes entidades usando nodos y enlaces; o (3) diagramas de dispersión animados que muestran la evolución de las relaciones entre variables a lo largo del tiempo.

2. 2. 1. 1. Dash

Dash [13] es una biblioteca de Python que se utiliza para crear aplicaciones web interactivas y paneles de control de datos de manera fácil y rápida. Está construida sobre Flask, Plotly.js y React, lo que la convierte en un framework de Python enfocado a la web moderna. Proporciona una amplia gama de componentes interactivos para construir aplicaciones, incluyendo gráficos interactivos, controles deslizantes, botones o tablas que permiten crear interfaces de usuario complejas y dinámicas para visualizar y explorar datos. Hace uso de un enfoque declarativo y reactivo para construir aplicaciones web, es decir, se define la estructura y el comportamiento de la aplicación utilizando Python puro, y luego Dash se encarga de la actualización automática de la interfaz de usuario en función de los cambios que se produzcan en los datos o la interacción del usuario. Se integra de forma sencilla con Plotly para poder usar la variedad de gráficos con los que cuenta. Facilita el despliegue de aplicaciones web en la nube o en servidores locales, pudiendo

empaquetar cualquier aplicación como un archivo Python estándar y ejecutarla en cualquier lugar donde Python esté instalado, facilitando compartir y distribuir las aplicaciones creadas.

2. 2. 1. 2. Tableau

Tableau [14] es una herramienta de visualización de datos que genera visualizaciones interactivas y paneles de control dinámicos a partir de una amplia variedad de fuentes de datos y, gracias a su interfaz de usuario intuitiva, no requiere escribir código para ello, abriendo este campo a usuarios de todos los niveles de habilidad. Cuenta con multitud de opciones para crear visualizaciones avanzadas, incluyendo gráficos de barras, líneas, mapas geoespaciales, diagramas de dispersión, etc. Permite personalizar el aspecto y el comportamiento de las visualizaciones con filtros, controles deslizantes, parámetros y otras herramientas para la identificación de tendencias, patrones, relaciones en los datos y realizar análisis comparativos a lo largo del tiempo. Además tiene opciones de colaboración para trabajar en equipo, donde compartir visualizaciones y paneles de control con partes interesadas y colaborar en tiempo real mediante comentarios y anotaciones. Se puede integrar con bases de datos, archivos planos, servicios en la nube y aplicaciones web, de forma que se puedan importar datos de múltiples fuentes y combinarlos en visualizaciones integradas para obtener una visión completa de éstos.

2. 3. Ejemplos de herramientas

Debido a la gran variedad de aplicaciones que existen para aplicar las técnicas que hemos comentado, a día de hoy existen muchos ejemplos de herramientas que se han desarrollado con el objetivo de desempeñar este tipo de tareas cuyo uso está bastante extendido. A continuación, se describen algunas de ellas:

2. 3. 1. Vader

En términos de popularidad, especialmente entre la comunidad de Python, destaca *VADER (Valence Aware Dictionary and sEntiment Reasoner)* [15], que es una herramienta dedicada al SA ampliamente utilizada en el procesamiento de texto, especialmente en el análisis de contenido en redes sociales. Fue desarrollada específicamente para abordar las peculiaridades del lenguaje utilizado en plataformas como X (anteriormente conocida como Twitter), donde es común encontrar texto informal, abreviaciones y otros elementos que pueden dificultar el análisis de sentimientos con herramientas convencionales.

La principal fortaleza de VADER se encuentra en su capacidad para capturar la intensidad del sentimiento expresado en un texto. Esto se logra mediante el uso de un diccionario léxico especialmente diseñado que asigna puntuaciones de polaridad a palabras individuales en función de su contexto y la intensidad del sentimiento que expresan. Por ejemplo, palabras como "bueno" o

"feliz" tendrían puntuaciones positivas altas, mientras que "malo" o "triste" tendrían puntuaciones negativas altas. Además, tiene en cuenta reglas gramaticales y de puntuación que pueden influir en el análisis de sentimientos, como el uso de letras mayúsculas, signos de exclamación o emojis, que también puede indicar un mayor nivel de intensidad emocional del texto.

Una característica distintiva de VADER es su capacidad para manejar correctamente el lenguaje sarcástico y las expresiones ambiguas, ya que utiliza un enfoque basado en reglas para interpretar el contexto y determinar el verdadero significado detrás de las expresiones lingüísticas, pudiendo identificar diferentes formas de ironía que podrían pasar desapercibidas para otros sistemas de este tipo.

2. 3. 2. TextBlob

También en el mismo grupo que Vader, destaca *TextBlob* [16], la cual es una potente biblioteca de PLN para Python que ofrece variedad de funcionalidades. Se utiliza comúnmente en aplicaciones que requieren comprender la opinión que se ha expresado en un texto, ya sea en comentarios, reseñas de productos o publicaciones en redes sociales de los usuarios. Su uso está extendido por la facilidad de integración con Python que posee, ya que solo se requieren unas pocas líneas de código para utilizar el análisis de sentimientos, lo que lo hace ideal para aplicaciones rápidas de prototipado y desarrollo.

La función de SA que tiene TextBlob utiliza un enfoque híbrido que combina reglas lingüísticas y un clasificador de aprendizaje automático para determinar la polaridad de un texto. Por un lado, usa un conjunto de reglas predefinidas para identificar palabras clave y patrones lingüísticos que pueden indicar el tipo de sentimiento, que incluyen consideraciones gramaticales (presencia de adjetivos o adverbios de negación entre otros) y también el uso de emoticonos o de diferentes signos de puntuación. Por otra parte, el clasificador de aprendizaje automático con el que cuenta ha sido entrenado en grandes conjuntos de datos etiquetados para reconocer el sentimiento en el texto y utiliza técnicas de aprendizaje supervisado para asignar automáticamente una etiqueta de polaridad a cada frase o documento de texto que se le presenta.

2. 3. 3. IBM Watson NLU

IBM Watson Natural Language Understanding [17] es una herramienta avanzada de procesamiento de lenguaje natural de las más utilizadas en entornos empresariales y proyectos que requieren análisis a gran escala en los que se trabaja con grandes volúmenes de datos y se necesita un soporte robusto y escalable. Está basada en la nube y se puede integrar fácilmente con otras herramientas y plataformas a través de API.

Cuenta con una gran capacidad para realizar análisis de sentimientos en el texto y utiliza técnicas de aprendizaje automático para identificar las emociones expresadas en el contenido, lo cual es fundamental para comprender la actitud y las opiniones expresadas en el mismo.

Otra característica importante es su idoneidad para extraer entidades y conceptos clave del texto utilizando técnicas de procesamiento de lenguaje natural, lo que permite hacer una clasificación de entidades como personas, lugares, organizaciones, fechas y cantidades, para obtener los temas destacables en el texto y extraer información relevante para su posterior análisis.

2. 3. 4. Google Cloud Natural Language API

El *Google Cloud Natural Language API* [18] es una herramienta poderosa y versátil que ofrece una amplia gama de funcionalidades para el análisis de lenguaje natural. Se integra fácilmente con otras plataformas y servicios de Google Cloud, así como con aplicaciones de terceros a través de API, permitiendo a los desarrolladores incorporar fácilmente capacidades avanzadas de análisis de lenguaje natural en sus propias aplicaciones y servicios, mejorando la funcionalidad y la experiencia del usuario.

Para el análisis de sentimientos en el texto, utiliza modelos de aprendizaje automático entrenados con grandes volúmenes de datos para determinar el tono emocional del contenido, clasificándolo según la polaridad. Su uso está enfocado en empresas y organizaciones, para que éstas puedan comprender la actitud de los usuarios hacia un producto, servicio o tema específico.

Está capacitada para extraer entidades clave del texto, identificando nombres de personas, organizaciones, lugares y fechas por ejemplo, así como las posibles relaciones entre ellas y ofrece funciones de clasificación de contenido, que pueden utilizarse para categorizar el texto, permitiendo organizar grandes conjuntos de datos de texto. De esta forma, se extrae información estructurada, lo que puede facilitar enormemente la comprensión y el análisis de la información.

3. Arquitectura y metodología

Este capítulo trata de explorar de una forma más detallada los pasos llevados a cabo para lograr alcanzar los objetivos propuestos en el desarrollo del proyecto propuesto, así como representar la arquitectura del mismo. Concretamente, se centra en los métodos, técnicas y herramientas escogidas que se han puesto en práctica durante el período dedicado a la creación de los módulos que lo componen y del papel que desempeñan en el funcionamiento de la aplicación.

3. 1. Diagrama y módulos del proyecto

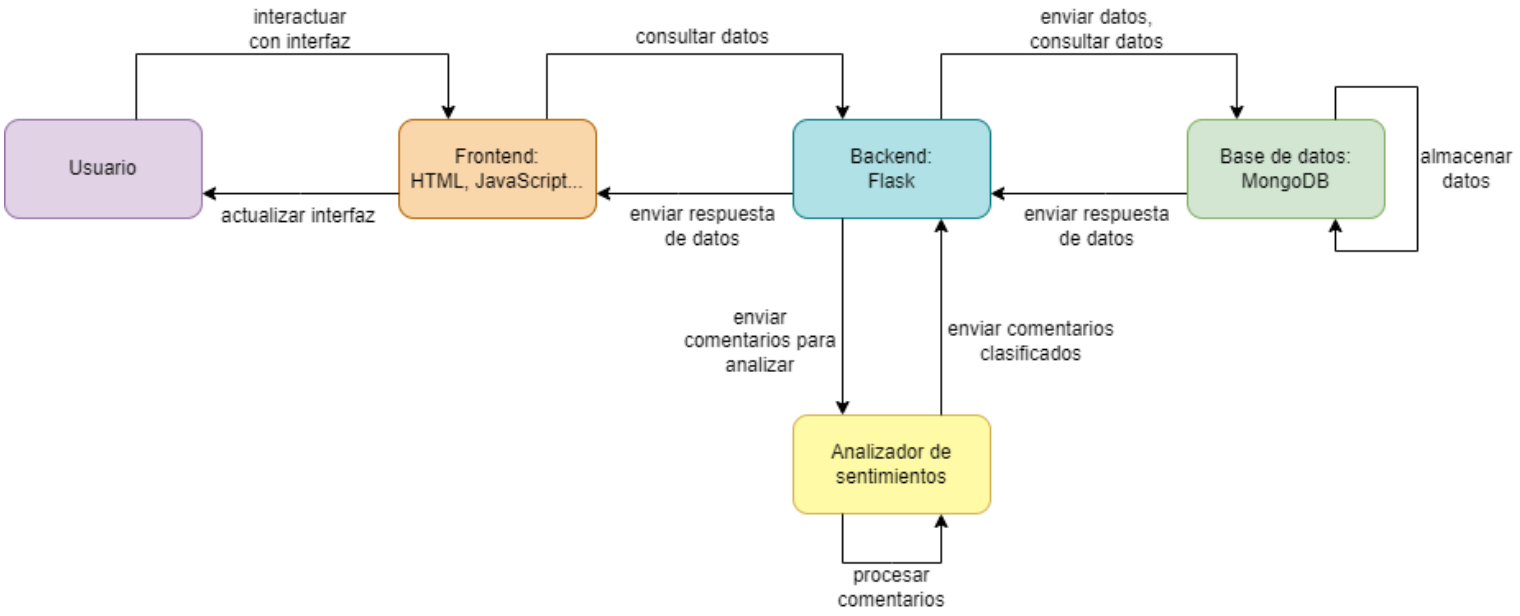


Figura 3. 1.- Diagrama de flujo del proyecto

Para poder llevar a cabo el desarrollo del proyecto de una forma estructurada, éste se organiza en módulos que construyen las diferentes piezas del puzle de la aplicación, tal y como se

muestra en la Figura 3.1. A continuación, se describe brevemente la estructura de cada uno de ellos de forma introductoria:

- **Backend (servidor)**: Se encarga de manejar las solicitudes de los usuarios que recibe a través del frontend, interactuar con la base de datos y proporcionar la lógica del negocio necesaria para el análisis de los comentarios de YouTube.
- **Base de datos**: Se encarga de almacenar los datos recogidos de la API de YouTube junto con los resultados del análisis de los mismos se usa MongoDB.
- **Frontend**: Se encarga de la visualización de los datos y la interacción del usuario a través de la interfaz proporcionada. Está dedicado a la representación gráfica de los datos, donde los usuarios tienen acceso a las posibilidades que ofrece el software. Las tecnologías usadas por este módulo son las siguientes:
- **Analizador de sentimientos**: Se encarga de analizar los comentarios de los vídeos de YouTube para determinar el sentimiento expresado en el texto, distinguiendo entre positivo, negativo y neutral usando un modelo basado en Transformers, destilado y multilingüe [19].

3. 2. Desarrollo del proyecto

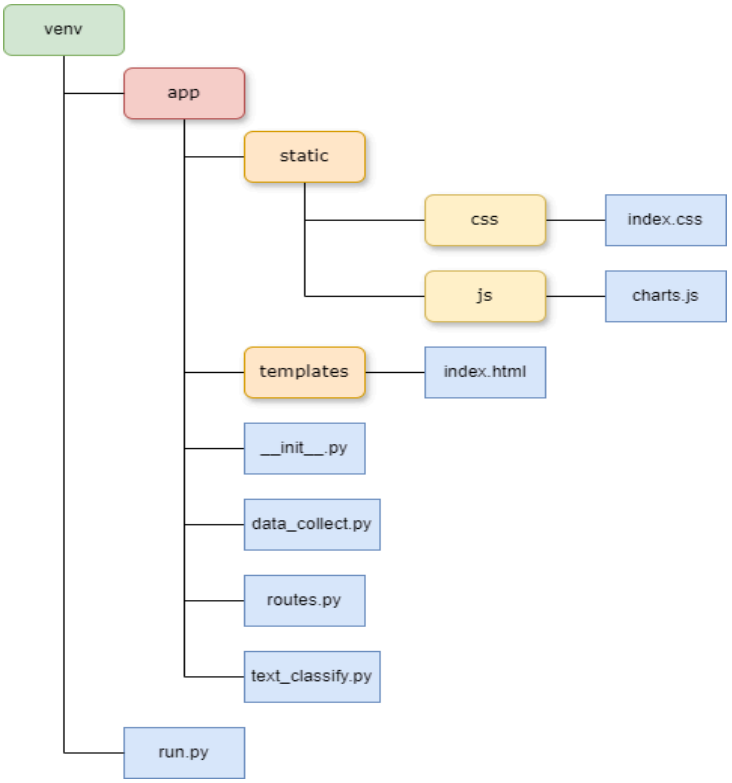


Figura 3. 2.- Estructura del código del proyecto

Tal y como puede verse en la Figura 3.2., el código del proyecto se estructura dividiendo las diferentes funcionalidades que lleva a cabo la aplicación en los diferentes archivos que lo componen, teniendo en cuenta para ello el módulo al que pertenecen y cuál es el trabajo que desempeñan dentro de ésta, que se explicará detalladamente más adelante.

Cabe mencionar que el desarrollo de la aplicación ha tenido lugar creando para ello un entorno virtual en el que se instalan y configuran los elementos necesarios para llevarlo a cabo, de ahí que los archivos cuelguen de un entorno virtual de Python.

El uso de un entorno virtual proporciona el aislamiento de las dependencias del proyecto, donde las bibliotecas y dependencias necesarias se instalan localmente para prevenir conflictos de versiones entre las distintas bibliotecas que puedan estar instaladas globalmente en el sistema operativo.

El contenido de cada uno de los archivos que componen el proyecto se describe de la siguiente manera:

- **`run.py`**: es el punto de entrada para ejecutar la aplicación Flask.
- `app/`**`__init__.py`**: inicializa la aplicación Flask y configura la base de datos.
- `app/`**`data_collect.py`**: funciones para interactuar con la API de YouTube y almacenar y tratar los datos recogidos.
- `app/`**`routes.py`**: define las rutas y los endpoints de la aplicación.
- `app/`**`text_classify.py`**: contiene las funciones para realizar el análisis de sentimientos de los comentarios de los vídeos aplicando el modelo correspondiente.
- `app/static/css/`**`index.css`**: alberga la hoja de estilos que completa el diseño de la interfaz de usuario.
- `app/static/js/`**`charts.js`**: archivo con el `eventListener` encargado de cargar los gráficos del dashboard a partir de las rutas de `routes.py`.
- `app/templates/`**`index.html`**: plantilla que se usa para crear la página principal de la aplicación para estructurar la información del dashboard adecuadamente.

3. 2. 1. Backend

Este proyecto utiliza servicios de tipo RESTful para exponer las funcionalidades del backend a través de endpoints, permitiendo una comunicación más directa y eficiente entre el frontend y el backend para crear los componentes gráficos. Se utiliza el patrón Modelo-Vista-Controlador (MVC), que se detalla a continuación:

- **Modelo**: Se encarga de la representación de los datos y la lógica de negocio.

- **Vista**: Se encarga de mostrar gráficamente los datos al usuario. Aunque Flask gestiona algunas partes de la vista, la mayor parte de la presentación de datos se realiza en el frontend. Las plantillas HTML definen la estructura de la página web y se rellenan con los datos proporcionados por el controlador, creando gráficos dinámicos basados en los datos recibidos del backend.
- **Controlador**: Se encarga de gestionar la lógica de negocio y la comunicación entre el modelo y la vista. En este caso, es Flask el que maneja las solicitudes HTTP y dirige las rutas adecuadas a los controladores correspondientes.

3. 2. 1. 1. API de YouTube

Para poder hacer el análisis del discurso en línea de los usuarios de la red social YouTube, se ha tenido que hacer uso de la API de la plataforma.

La API de YouTube permite acceder a diferentes datos de YouTube mediante solicitudes HTTP. En el caso de este proyecto, es la encargada de servir los vídeos y comentarios de YouTube específicos del tópico que se esté estudiando junto a diferentes características de los mismos y extraer sus comentarios para el análisis de sentimientos. También se extraen atributos como las fechas de publicación, el título del vídeo y su descripción, el usuario autor de los comentarios…

Se utiliza *googleapiclient.discovery* [20], una herramienta proporcionada por Google para facilitar el acceso a sus APIs, permitiendo a los desarrolladores interactuar con diversos servicios de Google mediante solicitudes HTTP.

El uso de la API de YouTube ofrece varias ventajas: (1) es gratuita; (2) permite obtener comentarios actualizados en tiempo real para un análisis más preciso y realista; (3) facilita la búsqueda y recuperación de grandes volúmenes de datos de manera eficiente y personalizar las consultas y filtrar los resultados según criterios específicos; e (4) incluye un apartado con muestras de código para hacer una simulación de cómo los datos son devueltos según los criterios de búsqueda que ayuda a la comprensión del funcionamiento.

El encargado de hacer las peticiones de datos es el backend de la aplicación, que utiliza esta API para buscar y recuperar vídeos y comentarios usando una palabra clave (la palabra que representa el tópico que se está analizando). Su puesta en práctica tiene lugar tal y como se describe a continuación:

1. Para configurar el acceso a la API se debe usar una clave a modo de credenciales [21]. Para este proyecto se pide una *clave de API*, que no identifica un principal (identidad a la que se le puede otorgar acceso a un recurso) y proporciona un proyecto de Google Cloud para fines de facturación y cuotas. Esta clave proporcionada se configura como una variable de entorno para mantener la seguridad.

2. Se usa `googleapiclient.discovery.build` para crear un cliente de servicio de YouTube utilizando la clave API que se obtiene del Google Cloud Console.

3. Se realizan solicitudes HTTP autenticadas con la clave de API correspondiente, para obtener vídeos y comentarios, extraídos usando una palabra clave (tópico del caso de estudio), junto con las características que puedan ser de interés de los mismos. La extracción de los datos se realiza en el archivo `data_collect.py`.

    - La función `youtube_service.search().list([...]).execute()` dentro de la función `collect_and_store_data` realiza una búsqueda en YouTube utilizando un término clave de búsqueda (`q=topic`) y devuelve una lista de vídeos relevantes del tamaño que se le indique.

    - La función `get_comments` implementada toma el ID de un vídeo y recupera la lista de comentarios asociados a éste aplicando un límite de resultados máximos.

4. Los datos obtenidos se procesan y almacenan en la base de datos de MongoDB del proyecto para su posterior análisis extrayéndoles información que pudiera ser relevante dentro de la aplicación.

3. 2. 1. 2. Flask

Flask [22] es un framework web ligero y flexible para Python, conocido por su simplicidad y extensibilidad. No viene con muchas herramientas preconfiguradas, lo que permite elegir y personalizar las extensiones según las necesidades específicas del proyecto desarrollado. Por este motivo, es el framework escogido para el desarrollo web de la aplicación, que se utiliza de (1) definir las rutas que responden a las diferentes URL en la aplicación mediante endpoints, que simbolizan los controladores de diferentes partes de la aplicación.; (2) manejar las solicitudes HTTP (GET, POST…)..; (3) e implementar la lógica del servidor, que incluye la interacción con la base de datos, el procesamiento de datos y la generación de respuestas dinámicas.

En la aplicación se usan varias rutas, las cuales se definen e implementan en el archivo `routes.py` de la carpeta `app`. Para configurarlas adecuadamente, se utilizan siguientes endpoints, cada uno dedicado a un propósito diferente:

- `/`: es el controlador de la ruta principal de la aplicación que manda renderizar la página principal del dashboard llevando a cabo para ello las siguientes funciones usando el método GET:

    - Si se proporcionan fechas de inicio y fin (`start_date` y `end_date`), filtra los vídeos y comentarios dentro de ese rango de fechas y se los pasa al frontend en las variables `videos` (lista de vídeos filtrados) y `comments` (lista de comentarios filtrados).

    - Calcula la distribución de sentimientos (positivo, neutral, negativo) de los comentarios filtrados a través de `sentiments` (diccionario con el conteo de comentarios por cada sentimiento).
    - Renderiza la página del dashboard completo (`index.html`) pasando los datos de vídeos, comentarios y sentimientos para su correspondiente visualización.
- **`/data`**: este endpoint usa HTTP GET, y es el encargado de recopilar todos los vídeos almacenados en la base de datos, obteniendo de cada uno los comentarios asociados y sus sentimientos para formatearlos en un JSON que devuelve para ser consumido por las visualizaciones del frontend dentro del contenedor de datos generales.
- **`/top_words`**: es el controlador de las palabras más repetidas. Devuelve las palabras más repetidas en los comentarios, utilizadas para el gráfico de barras de palabras más frecuentes usando para ello HTTP GET. Se encarga de realizar la consulta a la colección de palabras comunes en la base de datos, ordenándose por frecuencia de manera descendente, para devolver las cinco palabras más frecuentes junto con sus frecuencias en formato JSON.

3. 2. 1. 3. PyMongo

PyMongo [23] es una biblioteca de Python que facilita la interacción con MongoDB. Proporciona una interfaz intuitiva para realizar operaciones CRUD (Create, Read, Update, Delete) y ejecutar consultas en MongoDB desde aplicaciones Python.

En la aplicación se hace uso de ella para establecer y gestionar la conexión entre la aplicación Flask y la base de datos MongoDB. Además, permite que se realicen operaciones CRUD en la base de datos para, por ejemplo, agregar nuevos comentarios o consultar los datos almacenados. En el proyecto, su utilización tiene lugar en el archivo `data_collect.py` y en `routes.py`.

Asimismo, proporciona métodos para realizar las operaciones de agregación y filtrado de datos que permiten obtener estadísticas y hacer el análisis de los mismos, como contar la cantidad de comentarios por tipo de sentimiento o filtrar los comentarios por el rango de fechas que introduzca el usuario.

3. 2. 1. 4. NLTK

NLTK (Natural Language Toolkit) [24] es una de las bibliotecas más populares usadas para el procesamiento de lenguaje natural en Python. Esta biblioteca ofrece módulos y recursos para realizar análisis de sentimientos en texto, incluyendo la capacidad de determinar la polaridad del texto para la monitorización de redes sociales, la detección de la satisfacción del cliente y el análisis de opiniones de productos.

Entre sus características, destaca la de proporcionar una serie de herramientas para limpiar y preprocesar texto, incluyendo tokenización, eliminación de *stopwords* (tipos de palabras menos relevantes del contenido analizado, como pueden ser los determinantes), lematización y segmentación de frases, lo cual es fundamental para preparar el texto antes de realizar análisis más complejos. También destaca por disponer de gran variedad de recursos lingüísticos, como corpus de texto en varios idiomas, diccionarios, modelos de lenguaje o herramientas para el análisis gramatical.

En el caso de nuestro proyecto, NLTK se usa para preprocesar el texto y hacer el conteo de las palabras más comunes entre los comentarios. En concreto, se eliminan las stopwords y los signos de puntuación del texto de los mismos para tener en cuenta las palabras que sean verdaderamente relevantes.

3. 2. 2. Base de datos (MongoDB)

MongoDB [25] es la base de datos escogida para el almacenamiento de datos de esta aplicación. Es una base de datos NoSQL orientada a documentos, que utiliza un formato similar a JSON para guardar los datos. La elección se debe principalmente a que la estructura que sigue es adaptable y permite tener documentos almacenados con diferentes esquemas sin necesidad de modificar la estructura de la propia base de datos.

En este proyecto se usa para almacenar los resultados del análisis de sentimientos de la aplicación junto con los comentarios y las palabras más repetidas en ellos. Se almacenan en una base de datos llamada **`youtube_db`**. Cada comentario se guarda con su correspondiente etiqueta de sentimiento, lo que permite una rápida consulta y visualización de los resultados. La base de datos del proyecto alberga las siguientes colecciones dentro de ella:

- **Colección de `videos`**: almacena documentos que representan cada uno de los vídeos recogidos que son del tópico elegido, incluyendo algunos detalles del mismo, como el título, la descripción, la fecha de publicación y los comentarios asociados, que contienen a su vez la fecha de subida, el autor y el sentimiento con el que ha sido clasificado.
- **Colección de `palabras_comunes`**: almacena en documentos las palabras más frecuentes encontradas en los comentarios junto con su frecuencia de aparición en los mismos.

Además, este tipo de base de datos soporta operaciones de consulta complejas, como agregaciones y búsquedas, lo cual es de gran utilidad, ya que permite que se puedan filtrar y recuperar datos específicos, como es el caso de los comentarios y vídeos dentro de un rango de fechas por ejemplo.

Por otro lado, MongoDB está diseñado para escalar horizontalmente, lo que hace que pueda manejar un creciente volumen de datos y tráfico aumentando la cantidad de servidores si se quisiera, ofreciendo mayor escalabilidad que otras alternativas.

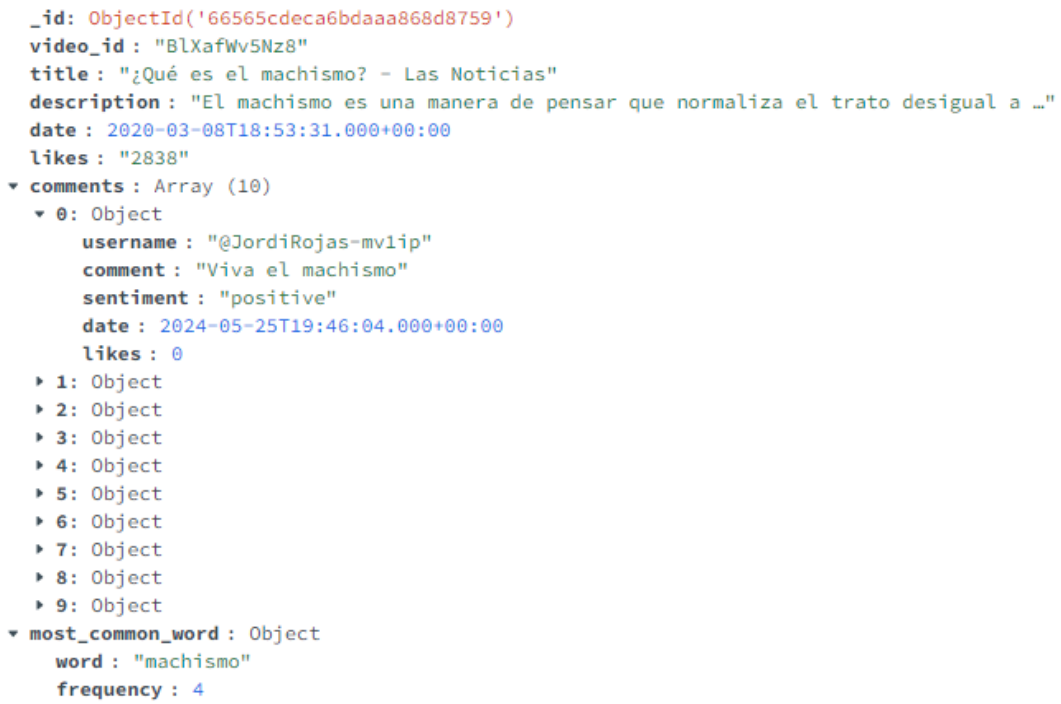


Figura 3. 3.- Ejemplo de vídeo en la base de datos

3. 2. 3. Frontend

El frontend es la cara visible de una aplicación web. Es la primera impresión que los usuarios tienen al interactuar con un sitio o una aplicación, por lo que juega un papel crucial en la experiencia del usuario.

El módulo dedicado al frontend de esta aplicación está diseñado para proporcionar una interfaz de usuario intuitiva y eficiente para visualizar el análisis de los vídeos y comentarios de YouTube de forma sencilla.

Para mostrar los datos de la aplicación, se lleva a cabo la comunicación e intercambio de la información necesaria con el backend para construir adecuadamente la página principal de la aplicación, permitiendo al usuario interactuar con ella y poder efectuar las correspondientes actualizaciones. Su desarrollo se apoya en una variedad de herramientas y tecnologías para crear interfaces de usuario interactivas y atractivas que se explican más detalladamente a continuación.

3. 2. 3. 1. HTML

HTML (HyperText Markup Language) [26] es el lenguaje estándar para crear y estructurar páginas web. Es un lenguaje de marcado que utiliza etiquetas para definir elementos dentro de un documento web que estructuran la información y establecen la semántica del contenido para mejorar la accesibilidad y los motores de búsqueda.

También soporta la inclusión de elementos multimedia como imágenes, vídeos y audio, así como gráficos interactivos a través de tecnologías como *Canvas* (`<canvas>`), de la cual se hace uso para dibujar los gráficos del dashboard. Ofrece la posibilidad de crear formularios (`<form>`) que permiten a los usuarios enviar datos al servidor, permitiendo usar varios tipos de entradas para ello que facilitan la interacción del usuario con la página web. Además, utilizando la etiqueta `<a>`, permite crear enlaces que conectan diferentes páginas web, facilitando la navegación entre ellas.

En el caso de este proyecto, se utiliza para estructurar la página web que contiene el dashboard con los datos correspondientes. Esto ocurre dentro del archivo `index.html`, donde se define la organización de los elementos en la interfaz de usuario, incluyendo el formulario para incluir un rango de fechas, los gráficos, y los diferentes contenedores de datos que se reciben del backend. La estructura de la página web se organiza en varias secciones principales:

- Contenedor general (**general-container**): una sección principal que incluye gráficos que proporcionan una vista general de los datos (sin filtros). Concretamente, se divide la información en dos contenedores:
    - **chart-container**: contiene un gráfico de barras verticales que, para cada vídeo recogido, muestra la clasificación por sentimientos con el número de comentarios asociado a cada uno de ellos.
    - **word-container**: muestra un gráfico de barras horizontales con el top 5 de las palabras más repetidas en los comentarios del tópico junto con su frecuencia de aparición.
- Contenedor de datos filtrados (**filtered-container**): es la sección dedicada a las funcionalidades de filtrado de los datos y la visualización de los resultados correspondientes. La información se reparte del siguiente modo:
    - **filter-container**: este contenedor incluye un formulario para que el usuario pueda filtrar los vídeos y los comentarios por rango de fechas.
    - **vídeo-container**: para mostrar el listado resultado de los vídeos filtrados por fecha.
    - **comment-container**: para listar los resultados de los comentarios aplicando el filtro de fechas.
    - **pie-chart-container**: en él se dibuja un gráfico circular con los sentimientos de los comentarios filtrados.

3. 2. 3. 2. CSS

CSS (Cascading Style Sheets) [27] es el lenguaje utilizado para describir la presentación y el diseño de una página web escrita en HTML, permitiendo separar el contenido HTML de su presentación, mejorando la mantenibilidad y flexibilidad del código.

CSS define los estilos visuales de los elementos HTML, incluyendo colores, fuentes, márgenes, rellenos, o bordes, por citar algunos ejemplos. Utilizando selectores se pueden aplicar estilos concretos a elementos específicos o grupos de elementos. Facilita el diseño de sitios web que funcionan bien en una variedad de dispositivos y tamaños de pantalla, y soporta animaciones y transiciones que permiten agregar efectos visuales dinámicos a los elementos web, mejorando la experiencia del usuario.

Por otro lado, presenta técnicas avanzadas como flexbox y grid, que permiten la creación de diseños complejos y responsivos eficientes, y se usan para la disposición de elementos en una única dimensión o para diseños bidimensionales respectivamente.

En el caso de nuestra aplicación, se usa para dar estilo y mejorar la presentación visual de la página. Estiliza los elementos que la componen y aseguran que el diseño sea atractivo para el usuario. Para ello, en el archivo `index.css` se define el diseño, colores, fuentes y disposición de los diferentes componentes de HTML a excepción de la personalización que ofrece la librería Chart.js para dibujar los gráficos de datos.

3. 2. 3. 3. JavaScript

JavaScript [28] es un lenguaje de programación versátil y dinámico que se utiliza para agregar interactividad y comportamiento a las páginas web. A diferencia de HTML y CSS, que se enfocan en el contenido y la presentación, JavaScript permite crear experiencias de usuario interactivas, ya que está orientado a responder a eventos del usuario mediante el uso de *event listeners* y controladores de eventos.

El Document Object Model (DOM) es una representación estructural del documento HTML. JavaScript tiene acceso a él y puede modificarlo en tiempo real, permitiendo actualizar el contenido y el diseño de la página sin necesidad de recargarla manualmente. Asimismo, con la tecnología AJAX (Asynchronous JavaScript and XML), JavaScript puede realizar solicitudes al servidor y actualizar la página web dinámicamente, muy relevante cuando las webs requieren comunicación en tiempo real.

Javascript se usa para añadir esa interactividad y dinamismo a la página web. Se usa para manejar los eventos del usuario, actualizar los gráficos en tiempo real y realizar solicitudes asincrónicas al servidor. Esto se implementa en el archivo `charts.js`, donde se maneja la lógica de negocio, procesando los datos de sentimientos y palabras según las necesidades del usuario y actualizando los gráficos de Chart.js en tiempo real usando para ello los datos que recibe de las distintas rutas definidas en la aplicación.

En el archivo JavaScript de la web es donde se especifican también las características y rasgos que pueden ser personalizados en los gráficos de Chart.js a representar en el dashboard. En concreto, se escogen los colores que se le dan a los datos, el grosor de los bordes que los contienen, si no se desea incluir título o leyenda.

3. 2. 3. 4. CHART.JS

Chart.js [29] es una librería de JavaScript que facilita la creación de gráficos interactivos y responsivos en páginas web. Ha sido la herramienta elegida en la aplicación para llevar a cabo la parte enfocada en crear los gráficos que representan los datos resultados del análisis a mostrar al usuario. Algunas de las características destacables que han influido a tomar esta decisión son: (1) facilidad para integración y uso, pudiendo crear gráficos con unas pocas líneas de código, ofreciendo una forma rápida y eficiente de visualizar datos; (2) soportar una amplia gama de tipos de gráficos, tales como barras, líneas, circulares, gráficos de radar; (3) sus gráficos son interactivos y responsivos, pudiendo responder a eventos de usuario y adaptándose automáticamente al tamaño que tengan en la pantalla; y (4) tiene muchas opciones de personalización, permitiendo ajustar la apariencia y el comportamiento de los gráficos en cuanto a colores, fuentes, leyendas…

De esta forma, en el proyecto se usa para renderizar los datos de manera visual, permitiendo a los usuarios interactuar con los gráficos y proporcionando una comprensión más profunda de los datos. La biblioteca se utiliza para crear los siguientes gráficos:

- El **gráfico general de sentimientos** es un diagrama de barras verticales que representa los datos de sentimientos asociados a cada vídeo de la base de datos, indicando el número de comentarios según polaridad, usando el color verde para los positivos, el amarillo para los neutrales y rojo para los negativos.
- Un diagrama de barras horizontales para mostrar el **top 5 de las palabras más repetidas** en los comentarios del tópico, usando un color para cada una de ellas, ordenadas de mayor a menor según el número de veces que aparecen.
- Un **gráfico circular** que se utiliza para la visualización de los **sentimientos asociados a los comentarios filtrados** por fecha, usando los mismos colores para ello que en el gráfico principal.

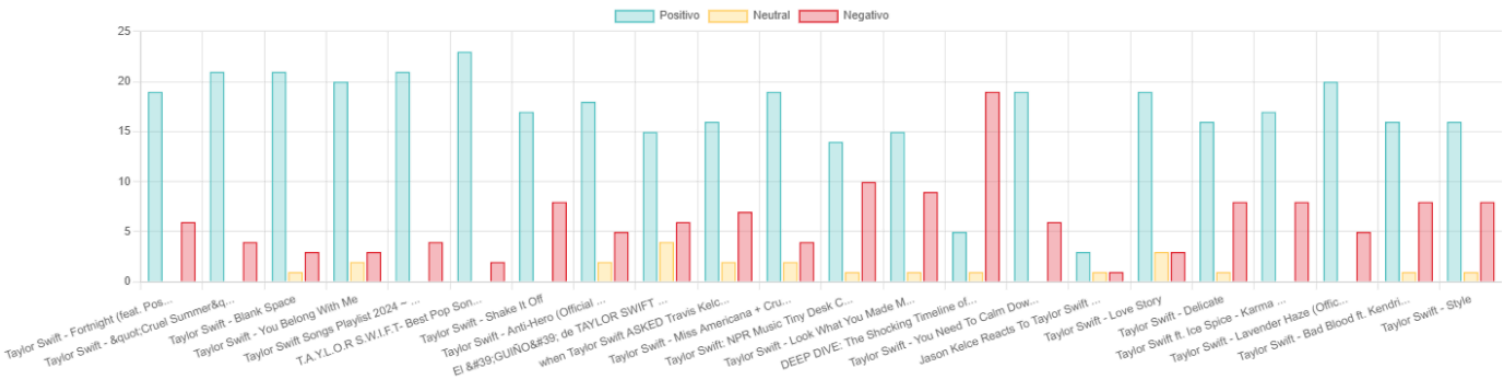


Figura 3. 4.- Ejemplo de gráfico general de sentimientos del tópico “Taylor Swift”

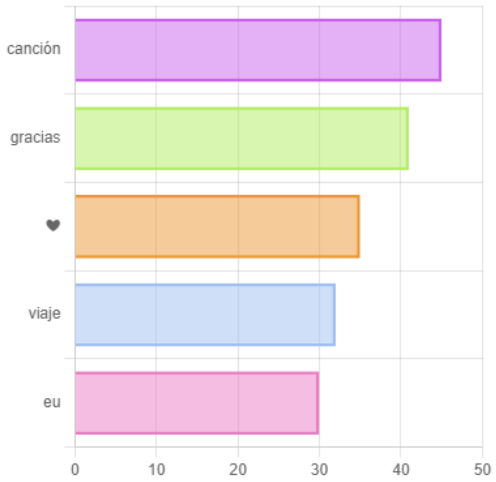


Figura 3. 5.- Gráfico ejemplo del top 5 palabras más repetidas del tópico “viaje”

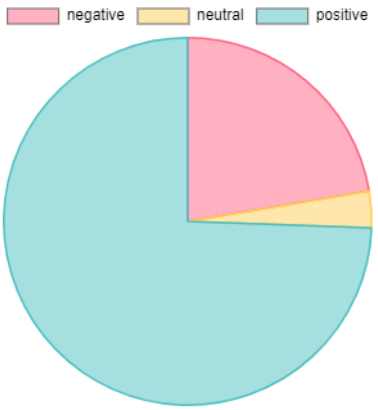


Figura 3. 6.- Gráfico circular ejemplo de sentimientos filtrados del tópico “viaje”

3. 2. 4. Análisis de sentimientos

El módulo dedicado al análisis de sentimientos en este proyecto se realiza utilizando el modelo *lxyuan/distilbert-base-multilingual-cased-sentiments-student* de Hugging Face para la clasificación de sentimientos en múltiples idiomas según la polaridad del texto.

El modelo en cuestión es una variante destilada de BERT (Bidirectional Encoder Representations from Transformers), que está preentrenado para tareas de procesamiento de lenguaje natural y, más específicamente, adaptada para el análisis de sentimientos de texto según se considere negativo, neutral o positivo.

DistilBERT es una versión reducida de BERT, creada mediante un proceso llamado destilación de conocimiento, donde un modelo más pequeño (estudiante) se entrena para imitar a un modelo más grande (maestro). Esto se traduce en un modelo más rápido y que consume menos recursos, pero conservando la mayoría de las capacidades del original. Además, este modelo ha sido

entrenado en datos multilingües, pudiendo manejar y analizar texto en varios idiomas. Es más ligera y eficiente, y mantiene un equilibrio entre rendimiento y velocidad, lo que la hace ideal para aplicaciones en tiempo real como es el caso del análisis de comentarios en YouTube.

Para usarlo en la práctica dentro de la aplicación, cuando se recogen los comentarios de cada vídeo de la API de YouTube, se llama para cada uno a la función `analyze_sentiment` del archivo `text_classify.py`, donde se llevan a cabo los siguientes pasos:

1. Se comienza importando: `DistilBertForSequenceClassification` y `DistilBertTokenizer`, que son clases de la biblioteca transformers de Hugging Face que proporcionan las implementaciones del modelo que se quiera aplicar.
2. Se cargan el tokenizador y el modelo preentrenado mencionado, que se utilizan para convertir el texto en una representación numérica que el modelo entiende y para realizar la predicción del sentimiento basado en el texto de entrada respectivamente.
3. Además, se define un diccionario `label_mapping` que mapea los identificadores de clase predichos por el modelo a etiquetas de sentimiento con 0, 1 y 2, que corresponden a sentimientos negativos, neutrales y positivos.
4. La función recibe el texto contenido en el comentario y lo tokeniza, pasa los tokens al modelo para obtener las predicciones de sentimiento, y finalmente mapea el identificador de clase predicho a una etiqueta de sentimiento utilizando el diccionario con el resultado correspondiente.

Los resultados del análisis de sentimientos sobre cada comentario se almacenan junto a éste en la base de datos MongoDB como una propiedad más, lo que facilita su extracción y representación para generar la parte gráfica de la aplicación y poder exponer los resultados del estudio al usuario de una forma cómoda.

3. 2. 5. Flujo de ejecución de la aplicación

En este apartado se describe el proceso que tiene lugar en la aplicación desde el momento en que se ejecuta desde la consola y cubre tanto la interacción entre el usuario con la misma, como la de los componentes del backend y frontend (*).

1. Una vez activado del entorno virtual del proyecto, se ejecuta **`python run.py`** en la consola del mismo. En el archivo que se ejecuta, se hace la **configuración** previa a iniciar el servidor web, definiendo los valores de: la clave de API de YouTube, el nombre del tópico que se va a analizar, los nombres de las colecciones para los datos a recoger y la URI de la base de datos.
2. Se procede al **llenado de la base de datos** llamando a la función `collect_and_store_data` de `data_collect.py`.

- Se recaban los datos de la API según el tópico usando la palabra clave, haciendo la **inserción de vídeos y comentarios** (si están habilitados) tal y como ya se ha comentado.
- Se produce el **análisis de sentimientos de los comentarios** en `analyze_sentiment` de `text_classify.py`, tal y como ya se ha explicado para clasificarlos entre positivos, neutrales o negativos.
- Se realiza el **cálculo y registro de las palabras más comunes** usadas en los comentarios recogidos usando las funciones `count_words` y `calculate_global_word_frequencies`, haciendo un **preprocesamiento del texto** con NLTK en `preprocess_text` y se guardan los resultados.

3. Se llama a la función `create_app` de `__init__.py` que crea el objeto `app`, registrando la ruta y *blueprint main*. A partir de este objeto, **Flask inicia el servidor web** (`app.run`) para escuchar las solicitudes en el puerto especificado (por defecto, el 5000).
4. Para mostrar la **página principal** desde la URL ***localhost:5000*** primero se debe establecer la conexión con la base de datos en el archivo `routes.py` para responder la solicitud.
    - La ruta `/` maneja la solicitud GET para renderizar la página principal (`index.html`) con los vídeos y comentarios (susceptibles a aplicación de filtros).
    - Desde el frontend se pide la resolución de las rutas `/data` y `/top_words` que manejan las solicitudes GET para obtener los datos de vídeos y comentarios analizados, y las palabras más frecuentes en ellos.
5. Una vez resueltas las rutas, el archivo `charts.js` procesa los datos recibidos y dibuja los gráficos del dashboard para completar la página a la que el **usuario ya tiene acceso**.
    - Inicialmente no se aplica ningún filtro y se muestran los datos de todo el contenido de la base de datos según corresponda.
    - El usuario puede aplicar un **filtro de rango de fechas** ingresando una fecha de inicio y una fecha de fin en el formulario para este propósito.

        5. 1. Cuando se envía el formulario, se realiza una nueva solicitud GET a la ruta `/` con los parámetros `start_date` y `end_date`.

        5. 2. En el backend, la ruta `/` recibe estos parámetros en `routes.py` desde la solicitud.

            - Se convierten las fechas de inicio y fin de los parámetros en objetos `datetime`.
            - Se ejecutan consultas en la base de datos para filtrar los vídeos y comentarios publicados entre las fechas especificadas usando `find` y `aggregate` de PyMongo para obtener los documentos que resulten de la filtración.
            - Se hace el conteo de los sentimientos asociados a los comentarios filtrados para redibujar el diagrama circular.

5. 3. Se organiza la información de los vídeos y comentarios filtrados en un formato adecuado para ser enviada al frontend.

5. 4. Los datos procesados se pasan a `index.html` mediante el método `render_template`.

- La plantilla recibe los datos de vídeos, comentarios y sentimientos para actualizar la página principal.
- El frontend recibe el HTML renderizado desde el backend y realiza solicitudes `fetch` a los endpoints `/data` y `/top_words` desde `charts.js`.
- En el backend estos endpoints responden con los datos correspondientes en formato JSON.
- El archivo JavaScript procesa los datos recibidos para dibujar los gráficos de sentimientos y palabras más frecuentes usando Chart.js.
- Todos **los gráficos se actualizan dinámicamente** con los datos recibidos.

5. 5. El usuario puede ver los vídeos y comentarios filtrados junto con los demás gráficos que representan los datos del dashboard.

6. La interfaz de usuario permite aplicar nuevos filtros de fechas para actualizar los datos y gráficos mostrados, lo que le llevaría de nuevo al paso 5. 1.

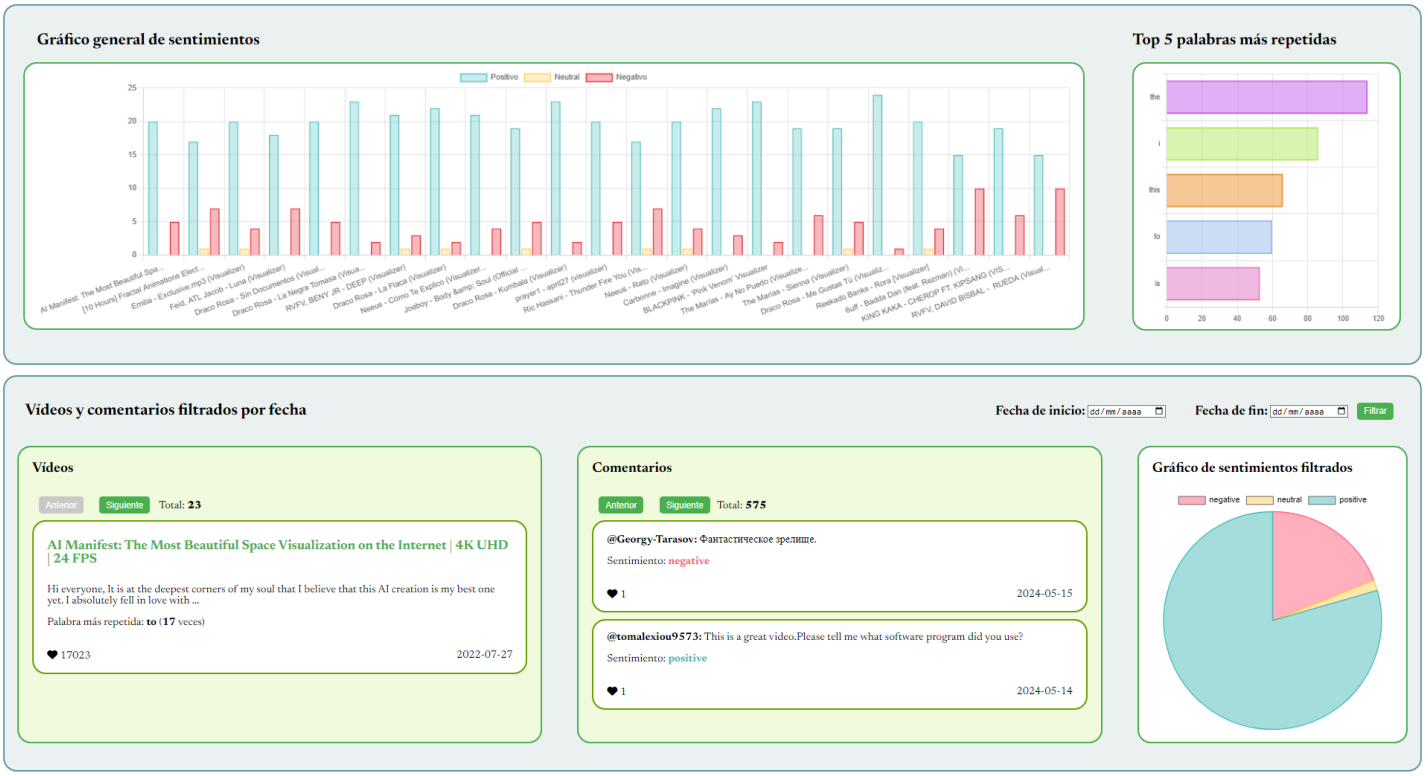


Figura 3. 7.- Ejemplo de ejecución usando la palabra clave “visualizer”

En resumen, en la aplicación la interacción entre el frontend y el backend se maneja principalmente a través de solicitudes HTTP, donde el frontend envía solicitudes para responder a las peticiones del usuario y el backend responde con los datos adecuados de la base de datos tal y como se ha explicado. Este proceso se facilita mediante el uso de JavaScript para realizar solicitudes y actualizar la interfaz de usuario sin necesidad de recargar la página.

(*) Proyecto ubicado en: https://github.com/pilarmedina00/TFG

4. Caso de estudio y resultados

En esta sección, se aborda el estudio de un caso específico para poner en práctica la aplicación y probar la efectividad del sistema desarrollado sobre el análisis del discurso en línea en YouTube. Al centrarnos en un tema particular, se obtiene una comprensión más profunda de los patrones de comunicación, las dinámicas de sentimiento y los temas predominantes en los comentarios relacionados con el tópico en cuestión.

4. 1. Análisis del discurso en línea: machismo en YouTube

Como caso de estudio, se ha seleccionado como tópico el *machismo* [30], un término que describe un carácter fuerte o agresivo de los hombres respecto de las mujeres, que a menudo se traduce en actitudes y comportamientos sociales que discriminan a las mujeres. Este tema controvertido proporciona un rico conjunto de datos para el análisis, ya que provoca fuertes reacciones y revela opiniones diversas de la comunidad en línea.

Para llevar este análisis a cabo se recogen los vídeos de YouTube de la API de la plataforma que contienen la palabra clave "machismo" y que, por tanto, abordan temas relacionados con él, incluyendo su impacto en la sociedad, experiencias personales, implicaciones culturales más amplias, igualdad de género o similares. El análisis se centra en una colección de veinticinco vídeos como máximo (se obvian los que tienen los comentarios deshabilitados), que discuten aspectos del tópico. De cada uno de los vídeos, se toman hasta veinticinco comentarios que permiten observar la tendencia en los sentimientos del mismo.

El objetivo es entender cómo los usuarios interactúan con este tema sensible e identificar las emociones y los temas dominantes presentes en sus discusiones. Al analizar los sentimientos expresados en los comentarios relacionados con los vídeos que discuten sobre este tópico, buscamos descubrir las respuestas emocionales subyacentes y la prevalencia de ciertas palabras clave asociadas con este discurso. Este examen detallado resalta las capacidades del sistema y

proporciona información valiosa sobre la naturaleza de las interacciones de los usuarios en el contexto de temas controvertidos o significativos.

Los resultados del análisis se visualizaron utilizando el dashboard interactivo desarrollado, que permite explorar los datos de manera dinámica.

4. 2. Resultados

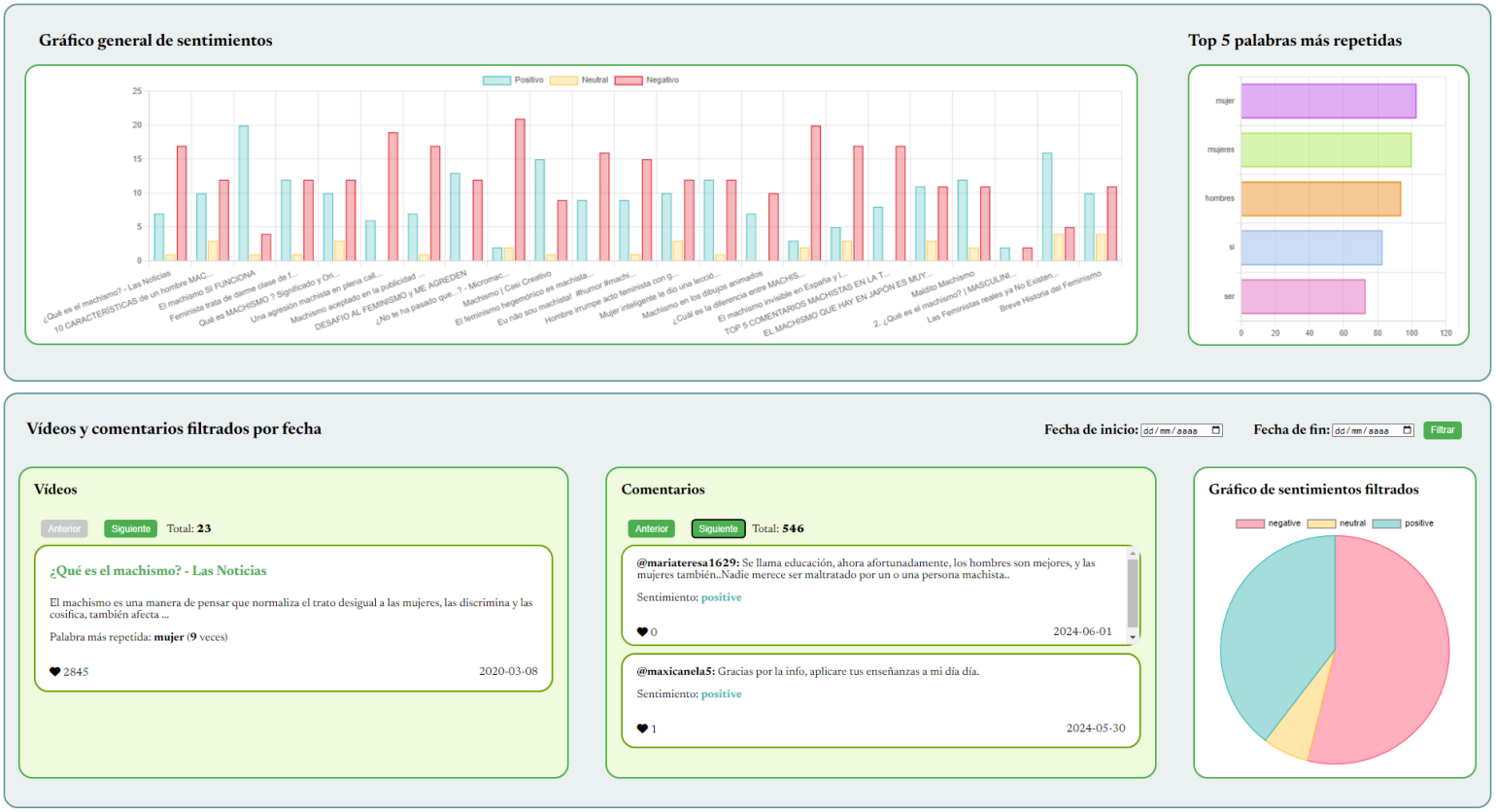


Figura 4. 1.- Dashboard de resultados

Como investigación previa a la obtención de los resultados del análisis de sentimientos del tópico escogido, se hacen varias pruebas para observar cómo varía la distribución de sentimientos según los diferentes vídeos y temas, reflejando la naturaleza diversa de las discusiones en YouTube. Por ejemplo, los vídeos relacionados con entretenimiento y estilo de vida como *vlogs*, vídeos de viajes o *storytimes*, tienden a tener muchos más comentarios positivos (Figura 3. 5. y Figura 3. 6.), mientras que aquellos relacionados con temas controversiales como problemas sociales o políticos, tienen una proporción baja o incluso nula de comentarios neutrales, mostrando que el discurso de los usuarios en línea sobre estos temas está mucho más polarizado (Figura 4. 1.).

Si analizamos la Figura 4.1. dedicada al tópico del caso de estudio detenidamente, se observa esta tendencia de división de los sentimientos. En ella, los vídeos tienen una distribución

de comentarios en la que predominan los sentimientos negativos, seguidos por los positivos. En su mayoría, el número de estos dos en cada vídeo concreto suele estar bastante igualado, siendo los comentarios neutros muy escasos, lo que sugiere que el tema del machismo genera reacciones intensas entre los espectadores.

La tendencia en este tópico son títulos llamativos o provocativos como "10 CARACTERÍSTICAS de un hombre MACHISTA" y "DESAFÍO AL FEMINISMO y ME AGREDEN", donde se observan más comentarios negativos. Otros vídeos de una naturaleza más expositiva, como los de noticias ("¿No te ha pasado que...? - Micromachismos | eldiario.es") o explicativos ("¿Cuál es la diferencia entre MACHISMO, FEMINISMO y HEMBRISMO?") tienen, sin embargo, una proporción mayor de comentarios positivos.

El análisis de frecuencia de palabras reveló patrones interesantes en el lenguaje utilizado por los usuarios de YouTube. Las palabras comunes incluidas en el tema analizado fueron sustantivos como "mujeres", "mujer" y "hombres"que aparecen con alta frecuencia, lo cual es lógico dado que el tema involucra cuestiones de género, reflejando que las discusiones sobre machismo se basan fundamentalmente en comparaciones y referencias a ambos géneros. La palabra "si", por otra parte, sugiere que las afirmaciones o consensos también son parte de las conversaciones de los comentarios. También destaca el verbo "ser" que sugiere que los comentarios se centran en definir qué es ser machista, describiendo comportamientos o actitudes de este tipo para emitir juicios y opiniones del tema. Este hecho revela que hay ciertas palabras clave estrechamente asociadas con cada tema específico, indicando la relevancia que tiene estudiarlas para comprender las conversaciones en curso a través de estas tendencias.

Utilizando los filtros para analizar diferentes rangos de fechas se proporciona información sobre cómo evolucionan a lo largo del tiempo las discusiones sobre el machismo. De esta forma, se observa que picos de mayor cantidad de comentarios (años 2023 y 2024) o de vídeos (años 2020 y 2023) que a menudo corresponden con eventos significativos, lanzamiento de nuevo contenido o resurgimiento de la popularidad de este tópico. Por ejemplo, un importante anuncio político o la ocurrencia de un delito mediático relacionado con este tema podrían resultar tanto en un mayor volumen de vídeos como de interacciones en los comentarios.

El listado de vídeos que se muestra por fecha de publicación, cuenta con información detallada de cada uno, incluyendo los *likes*, la palabra más repetida y el número de veces que aparece, que muestran el interés de los usuarios y en qué se basa el discurso de los comentarios en el mismo. De la misma forma se puede revisar cada comentario, ver el sentimiento que la clasificación le ha otorgado y los *likes* que ha recibido por parte de otros usuarios. En este apartado se observan desde comentarios de lo más concisos hasta expresiones de opiniones complejas y controversiales, por ejemplo, invitando a discusiones algo comprometidas sobre temas religiosos o usando expresiones verbales algo violentas, siendo un reflejo de cómo el tema puede desatar discusiones acaloradas donde los sentimientos están bastante marcados.

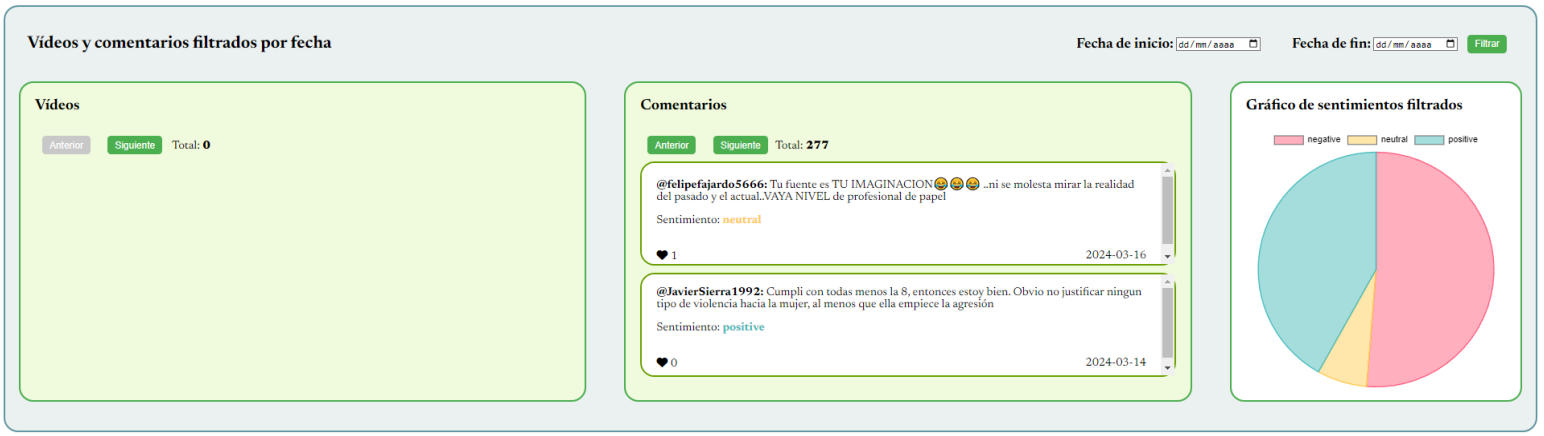


Figura 4. 2.- Contenido publicado en el año 2024

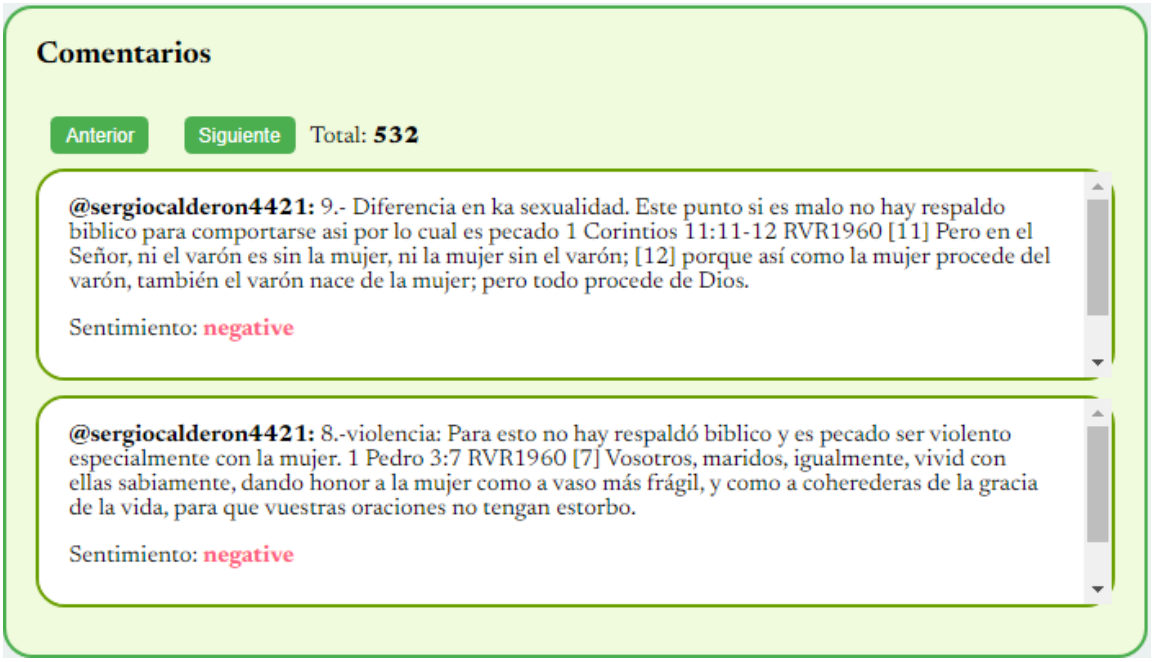


Figura 4. 3.- Ejemplo de comentarios más complejos

En resumen, los gráficos y datos del dashboard sobre este tópico demuestran que el machismo es un tema polarizador, generando polémica y exponiendo la sensibilidad del mismo. Las palabras más repetidas indican que las discusiones sobre este tópico están centradas en las relaciones y diferencias entre ser mujer y ser hombre. Por otro lado, la capacidad de filtrar por fechas permite entender cómo los eventos de actualidad a lo largo del tiempo pueden impactar en las discusiones sobre este tema. Como conclusión del uso aplicado de esta herramienta vemos que el análisis que proporciona ofrece una visión comprensiva y detallada de cómo se percibe y discute el tema del machismo en YouTube, resaltando la utilidad de herramientas de esta naturaleza para comprender las dinámicas sociales en las discusiones en línea.

5. Conclusiones y vías futuras

Este apartado expone las conclusiones principales del estudio llevado a cabo y se discuten posibles vías futuras para la mejora y expansión del proyecto, destacando tanto la importancia del análisis realizado como su relevancia en diversos contextos.

5. 1. Conclusiones

Durante el desarrollo de este proyecto se observa la importancia de entender cómo los usuarios se relacionan en la red, expresan sus opiniones e interactúan entre sí, proporcionando información valiosa sobre las tendencias sociales, los sentimientos que los usuarios quieren hacer públicos y los temas de interés emergentes, concretamente en YouTube.

El proceso de desarrollo de este proyecto ha proporcionado esta herramienta software que puede ser de gran importancia por varias razones:

- Conforma un sistema robusto de recopilación de datos capaz de recuperar vídeos relevantes sobre un tema concreto junto a sus comentarios.
- Permite clasificar el texto de los comentarios indicando su polaridad y recabar las palabras que aparecen con mayor frecuencia en los comentarios dentro del tema escogido.
- Los datos del caso de estudio se disponen ofreciendo una buena experiencia de usuario, facilitando la exploración de la dinámica social sobre el tema seleccionado.
- La visualización detallada de los datos escogiendo diferentes horquillas de fechas puede ser de interés para la investigación de momentos auge del tópico, permitiendo conocer las tendencias de los usuarios y las causas del mismo.
- Ofrece datos empíricos para estudios sociológicos y de comunicación, lo que puede ser crucial para investigadores, educadores y profesionales del marketing que buscan entender mejor las actitudes y comportamientos del público en esta plataforma.

- Aprovecha el creciente desarrollo de modelos de análisis de datos y de técnicas de PLN para intentar ofrecer una mayor precisión en la detección de sentimientos
- Ofrece la posibilidad de adaptarse a las necesidades de diferentes investigaciones, personalizando parámetros para realizar el filtrado de datos según interés y pudiendo elegir la cantidad de datos que se tienen en cuenta, tal y como se indica en la sección siguiente.

En términos de experiencia personal, este proyecto ha sido enormemente enriquecedor. Su desarrollo me ha permitido realizar una investigación exhaustiva sobre las diferentes herramientas propuestas que existen para el desarrollo de cada módulo. Una de las grandes ventajas en este sentido ha sido el descubrimiento de un gran número de posibilidades y tecnologías nuevas para mí, ampliando considerablemente mis conocimientos en áreas como: (1) plataformas y técnicas de PLN, aprendiendo sobre modelos de lenguaje, sus capacidades y sus limitaciones, y familiarizándome con bibliotecas populares como Hugging Face Transformers o NLTK; (2) desarrollo de aplicaciones web utilizando frameworks como Flask; (3) importancia de la integración de APIs para la recolección de datos en tiempo real; y (4) empleo de herramientas como Charts.js para crear visualizaciones interactivas y dashboards que faciliten la comprensión y exploración de los datos de manera intuitiva para obtener insights significativos.

En este proyecto, he comprendido la importancia de la flexibilidad y la adaptabilidad en el desarrollo de software, aprendiendo a evaluar y seleccionar las mejores herramientas y técnicas para cada tarea específica, asegurando que el proyecto no solo sea funcional, sino también eficiente y escalable. Además, demuestra que no es necesario contar con un gran presupuesto o un equipo numeroso para desarrollar un sistema completo que cumple con los objetivos planteados, ya que se han usado únicamente herramientas y recursos disponibles de forma gratuita, lo cual resalta la importancia de la accesibilidad en las tecnologías de inteligencia artificial y su potencial para ser aplicadas en una amplia gama de contextos, incluso con recursos limitados.

En conclusión, el sistema desarrollado no solo facilita la identificación de tendencias, sentimientos y temas clave dentro del contenido generado por los usuarios de YouTube, sino que también permite un examen detallado de cómo estos elementos evolucionan con el tiempo, llevando a la total comprensión de la dinámica social de esta red social. La experiencia personal obtenida a lo largo de este proyecto ha sido invaluable, proporcionando una profunda comprensión de las tecnologías y metodologías utilizadas. Estoy segura de que las habilidades y conocimientos adquiridos durante el desarrollo de este proyecto serán muy beneficiosos en mi futuro como profesional.

5. 2. Vías futuras

Aunque el sistema desarrollado cumpla con los objetivos propuestos del desarrollo del mismo, puede ser objeto de futuras mejoras que lo hagan más versátil y avanzado.

El software actual puede adaptarse a diferentes temas para observar la variabilidad según la naturaleza del análisis con tan solo cambiar el tópico a la hora de realizar la búsqueda de datos. Podría incluso usarse para investigaciones multidisciplinares que impliquen analizar varios temas, buscando por varias palabras clave y almacenando a cuál de ellas corresponden los vídeos recuperados para visualizar los resultados. Otra de las ventajas del software en este sentido es que tiene gran escalabilidad, permitiendo escoger el número de vídeos y comentarios a analizar según la cantidad de datos que se quieran barajar en cada investigación.

El trabajo futuro podría centrarse en optimizar el sistema para manejar flujos de datos en tiempo real de manera más eficiente. Esto podría implicar el uso de marcos de trabajo de computación distribuida, que permiten procesar y analizar grandes volúmenes de datos en tiempo real. La implementación de arquitecturas de canalización de datos más sofisticadas y el aprovechamiento de soluciones basadas en la nube para la escalabilidad también podrían mejorar significativamente el rendimiento del sistema y su capacidad para proporcionar información oportuna.

Otras mejoras futuras podrían poner el foco en la precisión y la sensibilidad del análisis de sentimientos, incorporando técnicas más avanzadas como la detección de sarcasmo y la comprensión de contexto, usando para ello modelos entrenados en conjuntos de datos específicos del dominio, lo cual podría marcar la diferencia para temas donde las opiniones pueden ser matizadas y no fácilmente clasificables.

Otro aspecto a tener en cuenta es que el sistema actual está limitado a YouTube, pero la metodología se puede extender o replicar en otras plataformas como X (antiguo Twitter), Facebook o Instagram, permitiendo obtener una visión más completa de la dinámica social en redes y pudiendo llegar a hacer análisis cruzados entre diferentes comunidades y plataformas.

La interfaz de usuario del dashboard podría ser mejorada incluyendo características interactivas más avanzadas y otras opciones de personalización para explorar los datos, donde incluir la representación de otros elementos que pudieran ser de interés o habilitar más opciones de navegabilidad. La retroalimentación de los usuarios y las pruebas de usabilidad podrían guiar estas mejoras en el futuro, asegurando que el tablero satisfaga las necesidades y expectativas de sus usuarios.

En resumen, este proyecto demuestra la efectividad de las herramientas de análisis de datos en tiempo real para comprender las discusiones en línea y sienta una base sólida para futuras investigaciones y aplicaciones prácticas en diversos campos. Las mejoras propuestas y las posibles expansiones del sistema subrayan su potencial para convertirse en una herramienta aún más poderosa y útil, abriendo oportunidades para investigaciones y desarrollos futuros.

6. Bibliografía

Por orden de aparición:

- [1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* Recuperado de https://arxiv.org/abs/1810.04805
- [2] Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology Behind Search (2nd ed.)*. Addison-Wesley.
- [3] Russell, M. A. (2019). *Mining the social web: Data mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and more.* O'Reilly Media.
- [4] Google Developers. (s.f.). *YouTube Data API Overview.* Recuperado de https://developers.google.com/youtube/v3/getting-started
- [5] Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing (3rd ed.).* Pearson.
- [6] Feldman, R. (2013). *Techniques and applications for sentiment analysis. Communications of the ACM*.
- [7] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-Training.* OpenAI.
- [8] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* Recuperado de https://arxiv.org/abs/1907.11692

- [9] HARRIS, Z. S. (1954). *Distributional Structure.* Word.
- [10] MANNING, C. D., RAGHAVAN, P., & SCHÜTZE, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press.
- [11] CORTES, C., & VAPNIK, V. (1995). *Support-vector networks.* Machine Learning.
- [12] PANG, B., LEE, L., & VAITHYANATHAN, S. (2002). *Thumbs up?: Sentiment Classification using Machine Learning Techniques.* Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP)
- [13] SMITH, C. (2020). *Dash: A Python Framework for Building Analytical Web Applications*. Recuperado de https://dash.plotly.com/
- [14] TABLEAU SOFTWARE. (s.f.). *What is Tableau?* Recuperado de https://www.tableau.com/what-is-tableau
- [15] HUTTO, C. J., & GILBERT, E. (2014). *VADER: A parsimonious rule-based model for sentiment analysis of social media text.* Eighth International Conference on Weblogs and Social Media (ICWSM-14).
- [16] LORIA, S. (2014). *TextBlob: Simplified Text Processing. Technical Report, TextBlob Documentation.*
- [17] IBM WATSON. (n.d.). *Natural Language Understanding.* IBM. Recuperado de https://www.ibm.com/cloud/watson-natural-language-understanding
- [18] GOOGLE CLOUD. (2022). *Natural Language API.* Recuperado de https://cloud.google.com/natural-language
- [19] YUAN, L. (2021). *lxyuan/distilbert-base-multilingual-cased-sentiments-student [Computer software].* Hugging Face. Recuperado de https://huggingface.co/lxyuan/distilbert-base-multilingual-cased-sentiments-student
- [20] GOOGLE. (n.d.). *Google API Client Library for Python.* Google Developers. Recuperado de https://developers.google.com/api-client-library/python
- [21] GOOGLE CLOUD. (n.d.). *Authentication. Google Cloud.* Recuperado de https://cloud.google.com/docs/authentication?hl=es-419
- [22] GRINBERG, M. (2018). *Flask Web Development: Developing Web Applications with Python (2nd ed.)*. O'Reilly Media.

- [23] Dirolf, B., & Hafner, C. (2019). *PyMongo: The Python Driver for MongoDB.* Recuperado de https://pymongo.readthedocs.io/en/stable/
- [24] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O'Reilly Media.
- [25] Banker, K., & Hawkins, R. (2015). *MongoDB in Action (2nd ed.).* Manning Publications.
- [26] W3C. (2017). *HTML5.2.* Recuperado de https://www.w3.org/TR/html52/
- [27] W3C. (2023). *CSS: Cascading Style Sheets.* Recuperado de https://www.w3.org/Style/CSS/
- [28] ECMAScript. (2021). *ECMAScript® 2021 Language Specification*. Recuperado de https://www.ecma-international.org/publications-and-standards/standards/ecma-262/
- [29] Chart.js Contributors. (2021). *Chart.js Documentation.* Recuperado de https://www.chartjs.org/docs/latest/
- [30] Real Academia Española. (2024). *Machismo.* En Diccionario de la lengua española (23.ª ed.). Recuperado de https://dle.rae.es/machismo