Recuperación de Información y Recomendaciones en la Web

2023

Informe final

Grupo 9

Maximiliano Gutiérrez 5.321.011-3

Bruno Rienzi 2.914.248-3

Introducción

Muchas veces nos encontramos buscando noticias sobre un tema en un sitio Web o buscador y luego queremos ampliar esa información a través de comentarios de los usuarios o de videos sobre esas noticias. En muchos casos, lo que hacemos para ampliar la información es repetir, en otra fuente, parte del texto de la noticia o del texto de búsqueda, como ser Youtube, X, Reddit, otro portal de noticias etc.

Una de las formas más comunes de complementar los artículos de noticias es mediante videos relacionados y comentarios de usuarios. Por este motivo, varios sitios suelen incluir junto con el artículo escrito un reproductor de video embebido y en algunos casos también permiten que los usuarios agreguen comentarios.

Si bien algunos sitios permiten complementar los artículos de noticias de esa manera, existen sitios especializados que manejan mejor un determinado tipo de información y en mayor volumen (ej. Reddit, Youtube, etc.). De esta forma, la búsqueda de información complementaria se realiza en muchos casos utilizando las funcionalidades de búsqueda de cada sitio.

A su vez, en años recientes se ha generado una tendencia a la creación y difusión de noticias falsas (*fake news*) o publicaciones en donde es difícil determinar si se manejan datos objetivos a afirmaciones subjetivas. Por ejemplo, un periodista puede citar la declaración de un gobernante atribuyéndole un determinado tono, que sin embargo no se corresponde con el tono que percibimos cuando podemos ver el video. Esto hace que la necesidad de consultar varias fuentes y en diferentes formatos sea aún más relevante.

Problema

El problema que se plantea consiste en investigar la posibilidad de mostrarle al usuario información integrada que proviene de diferentes fuentes y en diferentes formatos. De esta forma el usuario recibe tanto el texto de la noticia como los videos del hecho y comentarios de los usuarios, presentando eventualmente diferentes puntos de vista para la misma noticia, permitiendo así, que el usuario pueda ver una imagen más diversa y menos sesgada. La integración de la información debe hacerse en forma automática (no existe un administrador que integra la información manualmente) y los errores de integración de información (ej. el video no corresponde al hecho de la noticia) deben minimizarse para mantener un nivel de confiabilidad razonable.

Enfoque de solución

Nuestra propuesta consiste en un sitio Web en donde el usuario ingresa un texto de búsqueda y el sistema le muestra en forma en forma organizada las noticias que concuerdan con su búsqueda, junto con los comentarios y videos relacionados, obtenidos de Reddit y YouTube respectivamente.

Las principales características de la solución son las siguientes:

- Las noticias se obtienen mediante la News API [1] que permite configurar las fuentes de noticias (ej. BBC, Google News, etc.), el idioma, rango de fechas, y ordenarlas por relevancia, popularidad, etc. Existen más de 100 fuentes de noticias que son accesibles mediante esta API.

- Utilizando el título de la noticia, se obtienen los videos de la YouTube API y los comentarios de la Reddit API buscando en todos los subreddits.

- El orden de los resultados es por relevancia. Cada API la calcula en base a su propio algoritmo, considerando palabras clave, popularidad, etc. Por ejemplo la YouTube API considera métricas como la retención (porcentaje de tiempo que los espectadores pasan viendo un video sobre su duración total) o involucramiento (ej. likes, comentarios, etc.) para establecer la relevancia de un video.

Las consultas que realizamos en Elasticsearch tienen las siguientes características:

- Utilizamos una *bool query* que permite combinar condiciones mediante operadores lógicos.

- Utilizamos el operador lógico *should* para indicar que al menos un criterio debe coincidir para que se devuelva un documento, con el parámetro *minimum_should_match=1* para que al menos un criterio deba cumplirse.

- Buscamos en los campos título y descripción, multiplicando por 2 la relevancia cuando la coincidencia se da en el título, mediante el parámetro *boost=2*.

- Utilizamos consultas con *match_phrase_prefix* y un *slop=2* para permitir hasta 2 términos distintos entre las palabras de la frase para que se considere una coincidencia. Por ejemplo, los textos “guerra Ucrania” y “guerra en Ucrania” tienen un slop de 1, por lo que se consideran equivalentes en nuestro sistema. Un valor de slop mayor a 2 podría mejorar el recall pero perjudicar la precisión.

Diseño e Implementación

En la Fig. 1 mostramos la arquitectura propuesta, la cual contiene los siguientes componentes:

- El Extractor realiza la recuperación de datos offline, consultando las fuentes y correlacionando los resultados.

- Los datos integrados se agregan a un índice en Elasticsearch.

- El Front-end realiza la búsqueda online sobre los resultados integrados usando la API RI, que consulta a ElasticSearch.

- Los videos se obtienen directamente de YouTube mediante el ID del video.

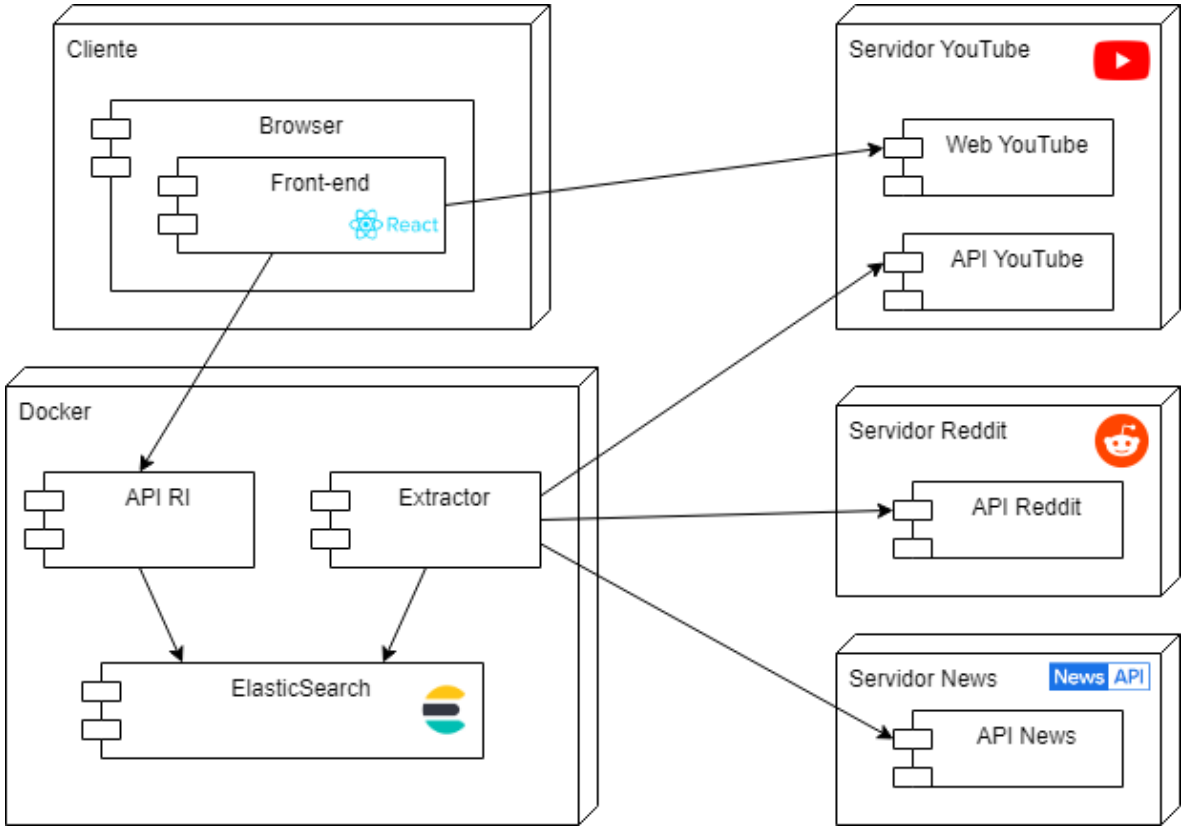


Figura 1 - Arquitectura de la solución propuesta

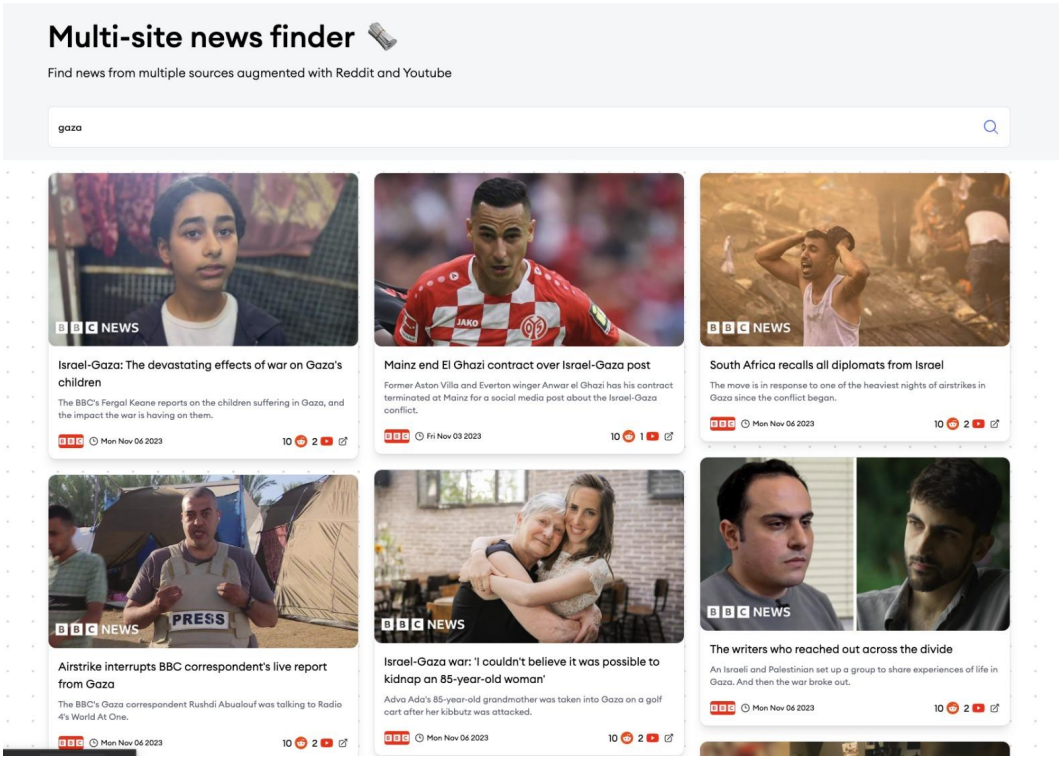
Para probar la factibilidad técnica de la solución se desarrolló un prototipo. El backend está implementado en Python ya que este lenguaje posee librerías actualizadas que permiten interactuar con las APIs necesarias sin escribir código de bajo nivel. Las librerías utilizadas son newsapi para utilizar la News API, googleapiclient.discovery para utilizar la YouTube API [2] y praw para utilizar la Reddit API [3] .

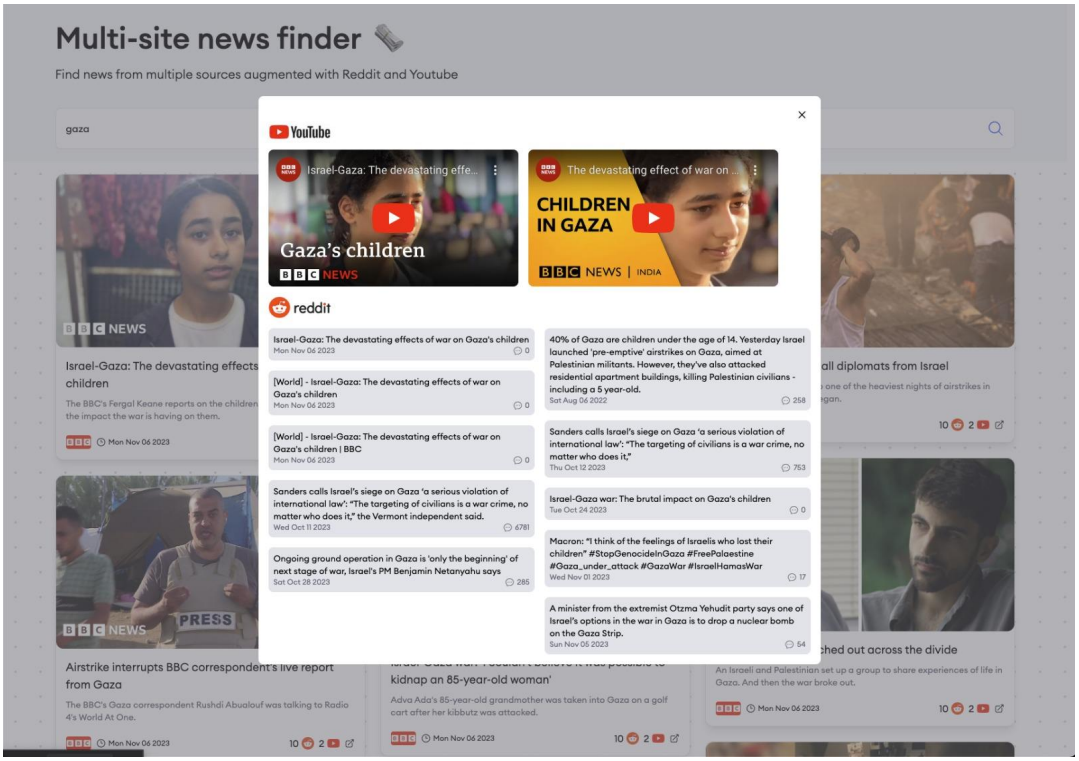
El frontend está implementado en React, lo que brinda la posibilidad de tener un interfaz de usuario moderna, escalable y fácilmente mantenible.

Evaluación y resultados

A continuación se muestran capturas de la interfaz de usuario de la aplicación. En la sección superior se encuentra el buscador, donde por cada modificación del término de

búsqueda, se realiza la consulta al API RI, y se muestran los resultados luego de obtenida la respuesta.





En cuanto a las limitaciones del prototipo implementado, es importante destacar que nuestra base de datos se encuentra restringida debido a las condiciones de la clave gratuita de la News API, que nos ofrece un conjunto limitado de 100 noticias como máximo. Ese máximo

es diario, por lo que en un enfoque incremental se cargarían 100 nuevas noticias todos los días, y en caso de precisarse más, se recurriría a una licencia de uso comercial. La base de datos en sí puede manejar grandes volúmenes de información.

Además, en lo que respecta a la obtención de contenido asociado a estas noticias, nos encontramos con ciertas limitaciones en la cantidad de comentarios y videos que podemos obtener. Actualmente, nuestro enfoque se centra en capturar los 10 primeros comentarios vinculados a cada noticia. Asimismo, en lo que respecta a los videos, nuestra implementación se limita a los 2 primeros por cada noticia. Esta decisión se fundamenta en la relevancia y la importancia percibida de los primeros comentarios y videos en comparación con los últimos. Al aumentar la cantidad de comentarios y videos recuperados, encontramos que la calidad y la relevancia de los últimos disminuyen significativamente en comparación con los primeros. Esta estrategia nos permite priorizar la calidad sobre la cantidad, asegurando que los usuarios accedan a la información más pertinente y valiosa en el contexto de cada noticia.

Conclusiones y Trabajos a Futuro

La conclusión principal a la que llegamos es que el prototipo implementado muestra que la solución propuesta al problema es factible. Si bien se trata de un prototipo, la correspondencia entre los datos de las diversas fuentes es alta, por lo que la mayoría de los resultados que nos brindan las búsquedas son coherentes y corresponden a la información buscada. Esto muestra que soluciones de este tipo pueden ser desarrolladas con un ciclo corto de desarrollo ya que las tecnologías utilizadas (APIs, librerías, ElasticSearch, etc.) ya poseen un nivel de madurez y documentación que hacen que esto sea posible.

En cuanto a la carga e integración offline, esto permite mejorar los tiempos de respuesta y aprovechar las capacidades de ElasticSearch ya que la integración se realiza en el backend. En un sistema de este tipo que fuera puesto en producción, sería necesario que la carga se realizará en forma periódica y con algún criterio que permitiera manejar prioridades para la carga de noticias (ej. los resultados de partidos, elecciones, etc. deberían estar disponibles en forma inmediata o casi inmediata).

Como trabajos a futuro, plantemos los siguientes. En primer lugar, uno de nuestros objetivos es optimizar el proceso de búsqueda de noticias. Para lograr esto, planeamos implementar estrategias avanzadas de preprocesamiento del título de las noticias antes de realizar consultas en nuestras diversas APIs. Esto podría implicar la evaluación y posible eliminación de palabras irrelevantes o comunes, también conocidas como 'stop words', con el fin de mejorar la precisión y relevancia de los resultados obtenidos.

Además, queremos otorgar a los usuarios un mayor control sobre sus búsquedas, brindando configuraciones personalizadas que les permitan ajustar los parámetros de búsqueda para obtener resultados más precisos y específicos. Estas configuraciones permitirán modificar los valores predeterminados que actualmente utilizamos en nuestras

búsquedas a través de Elasticsearch, ofreciendo a los usuarios la capacidad de adaptar la búsqueda según sus preferencias y necesidades individuales.

Otro aspecto interesante consiste en la integración de filtros personalizados. Esto incluirá la implementación de técnicas como sinónimos y stemming, entre otras, con el objetivo de refinar aún más la precisión de los resultados de búsqueda.

Finalmente, nos planteamos cómo aprovechar la geolocalización del usuario para potenciar la relevancia de los resultados. Esta iniciativa implica utilizar la información de ubicación del usuario para influir en la clasificación y presentación de los resultados, mejorando así la pertinencia de la información que se muestra en base a lo que está ocurriendo en su ciudad, país, etc.

Referencias

[1] *News API*: https://newsapi.org

[2] *YouTube API*: https://developers.google.com/youtube/v3

[3] *Reddit API*: https://www.reddit.com/dev/api/

[4] *Elastic Search*: https://www.elastic.co/guide/index.html