

WebIR 2018

segunda entrega

Bruno Di Bello - 4.887.674
Juan Roquero - 4.473.116
Maximiliano Barnech - 4.505.184
Pablo Martinez - 5.170.824

Introducción	3
Problema	3
Arquitectura	3
Componentes	4
Twitter API	4
Elastic Search	4
Backend	4
Frontend	4
Implementación	5
Backend	5
Frontend	6
Funcionalidades y uso	7
Evaluación y resultados	7
Conclusiones	7
Trabajo Futuro	8
Frontend	8
Backend	8
Referencias	9

Introducción

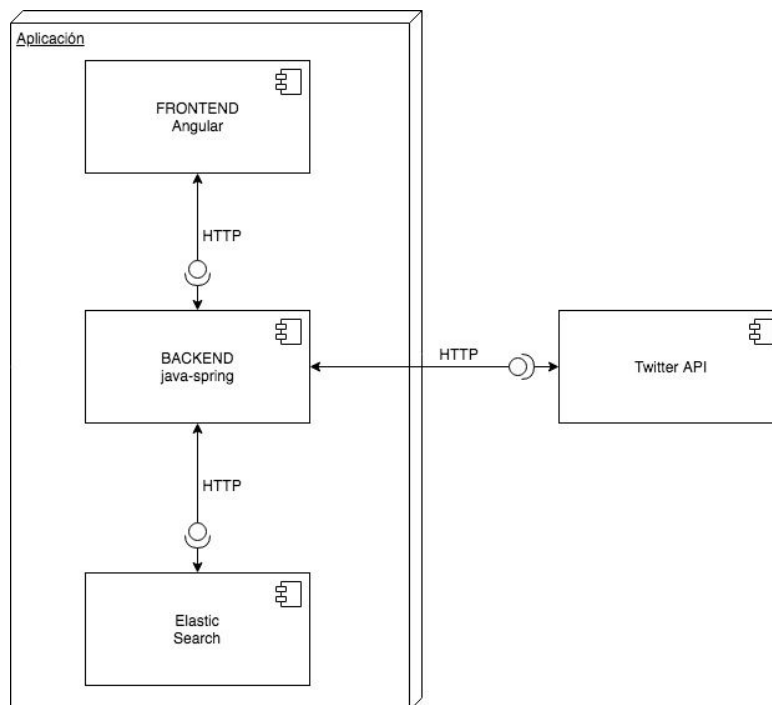
Las redes sociales día a día son más utilizadas por estadísticos. Marcan tendencia y ayudan a establecer entre otras cosas estrategias de campañas políticas.

A partir de esto el proyecto sugiere obtener las diferentes referencias políticas de los usuarios de la plataforma Twitter. Estas referencias serán obtenidas mediante los tweets publicados por los usuarios de la red social acotadas a ciertas fechas.

Problema

El problema a resolver es obtener las opiniones de los usuarios de twitter de distintas zonas de Uruguay para determinados intervalos de tiempo sobre los distintos agentes políticos. Interesa conocer en particular la cantidad de tweets sobre determinado tema en dicha zona e intervalo de tiempo así como también cualquier otro tipo de datos sobre los usuarios que se puedan recabar. La búsqueda deberá ser eficiente y tener filtros variables que el usuario del sistema pueda seleccionar. La interfaz gráfica deberá presentar los datos de forma clara y concisa.

Arquitectura



Componentes

Twitter API

Twitter provee una API rest de la cual se pueden obtener los tweets de determinada zona geográfica y en cierto rango de tiempo.

Elastic Search

ElasticSearch es un motor distribuido, RESTful de búsqueda y análisis capaz de resolver un número creciente de casos de uso. Éste motor será utilizado para realizar la búsqueda de identificadores (nombre políticos y relacionados) dentro del contenido de los tweets.

Backend

El backend es el encargado de obtener los tweets utilizando la la API rest de Twitter y de ingresarlos a ElasticSearch para ser indexados. Además provee una interfaz que consumirá el Frontend para obtener las consultas sobre los tweets.

Frontend

El frontend es el encargado de desplegar la cantidad de apariciones de cierta figura política dentro del conjunto de tweets ordenadas en el tiempo y agrupadas por día.

Implementación

Backend

El backend del sistema consiste de una aplicación implementada utilizando Java 8 con el framework Spring y además un nodo de Elastic Search en el cual se almacenan los tweets.

El flujo es el siguiente:

- Se inicia el nodo de Elastic Search
- Al iniciar el backend se obtienen los tweets
- Los tweets obtenidos son ingresados en el nodo de Elastic Search
- Se inicia la aplicación web que disponibiliza el endpoint utilizado por el frontend

Obtención de tweets:

Para obtener los tweets se utiliza la API Rest que expone Twitter, la cual permite obtener tweets dada cierta fecha, ubicación y ciertos criterios de búsqueda. El criterio para obtener los tweets es el siguiente:

- Se obtuvieron los tweets del último mes y medio
- Los tweets son de la zona Metropolitana
- Los criterios de búsqueda (keywords que son enviados a la API de Twitter) que fueron utilizados son los siguientes: politica, elecciones, campaña, mujica, astori, tabare, vazquez, pepe, bonomi, sendic, raul, daniel, martinez, edgardo, novic, luis, lacalle, pou, pedro, bordaberry, jorge, larrañaga, fa, frente amplio, pc, partido colorado, pi, partido independiente, mpp, partido nacional, pn.

Endpoint en backend:

Dentro del backend exponemos un único endpoint el cual permite obtener las la cantidad de tweets (documentos dentro de Elastic Search) que coinciden con cierto criterio de búsqueda, en nuestro caso son los nombres de los diferentes políticos de nuestro país. Para realizar la búsqueda dentro de Elastic Search utilizamos Fuzzy Search dentro del texto de los tweets almacenados ya que los usuarios de Twitter pueden cometer errores al ingresar los nombres de los políticos y este tipo de búsqueda nos permite obtener resultados por aproximación.

El endpoint que disponibilizamos es el siguiente:

```
get_tweets_count?search=<nombre>
```

Cuya respuesta es un json con el siguiente formato:

```
{
  "search": "nombre a buscar",
  "total_count": 20,
  "results": {
    "08/10/2018": 4,
    "07/10/2018": 5,
    "10/10/2018": 2,
    "12/10/2018": 6,
    "15/10/2018": 3
  }
}
```

Donde podemos encontrar:

- *search* indica el nombre que se buscó
- *total_count* es la cantidad total de documentos que encontró Elastic Search con dicha búsqueda
- dentro de *results* tenemos la cantidad de resultados agrupados por fecha.

Frontend

El frontend del sistema consiste de una aplicación angular, implementada en Angular 4.

Se decidió utilizar esta tecnología por la familiaridad de los estudiantes con la misma.

Fue implementado de una manera sencilla para cumplir con el cometido de mostrar resultados y facilitar el uso del sistema.

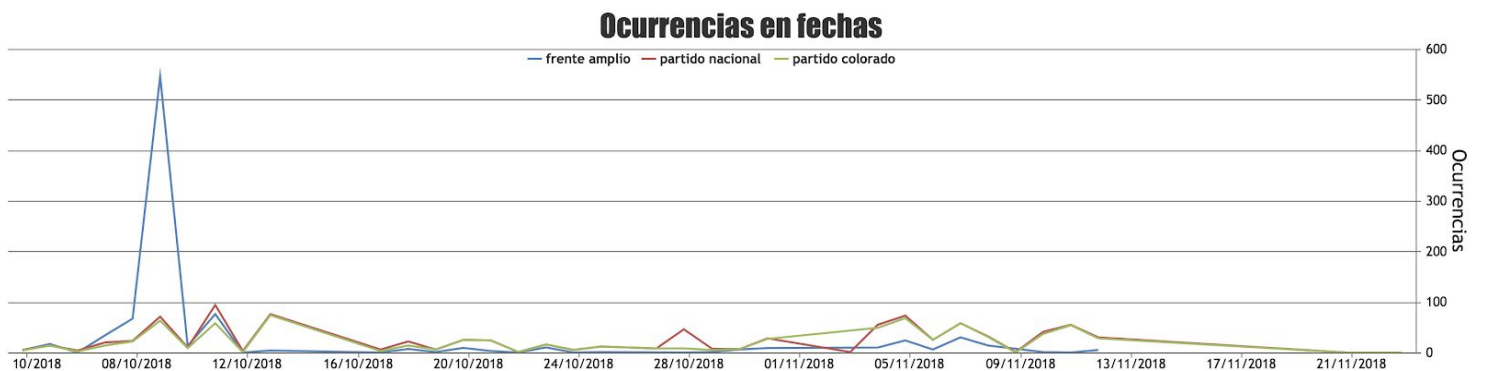
Se comunica con el backend mediante web services expuestos por el mismo. En términos generales la comunicación consiste en lo siguiente: el frontend una vez recibe como input del usuario el nombre del político que se desea buscar, este nombre se envía al backend y en respuesta el frontend recibe la cantidad de tweets que contienen al nombre del político y en qué fechas. Este conjunto de datos de ocurrencias y fechas son graficados y mostrados al usuario utilizando una biblioteca llamada "CanvasJS".

Funcionalidades y uso

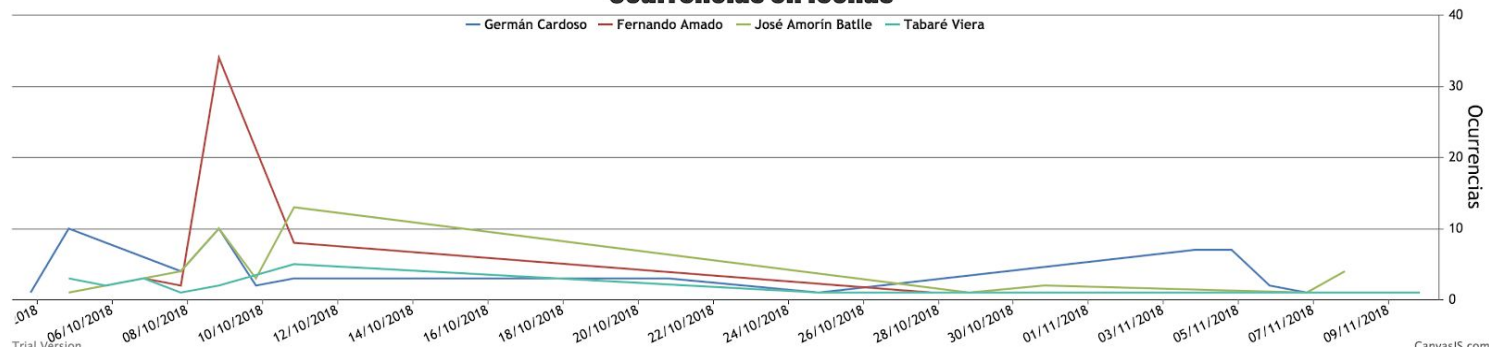
Se utilizará el sistema para obtener datos estadísticos así como también descubrir tendencias de los usuarios relativas a un partido político o persona política.

Otra de las funciones podría ser su utilización para campañas políticas ya que en base a los datos recabados por el sistema se facilitan varios aspectos del análisis de una situación política para un cierto lugar específico.

A continuación se presentan los resultados obtenidos al realizar consultas sobre las ocurrencias de los partidos tradicionales y sus precandidatos:



Ocurrencias en fechas



Evaluación y resultados

Resulta notorio cuando cierto tema político, como por ejemplo alguna noticia reveladora, ya que se observan más comentarios en twitter por medio de la gente relacionado a ese tema en particular que a los otros.

Un ejemplo de esto podría verse tras las elecciones de brasil. Al buscar la palabra Bolsonaro, se observa un gran crecimiento de comentarios relacionados al mismo en días previos y post elecciones. Esto da un indicativo de cuáles son los temas tendencia y cuales comienzan a ser temas más olvidados.

Un comportamiento similar surge al observar los resultados de Sendic o Mujica, donde se encuentran claras asociaciones con noticias que los relacionan o eventos que suceden de los cuales ellos forman parte.

Conclusiones

Como podemos observar en los resultados presentados en la sección anterior, con solo obtener las cantidades de menciones de cierta figura política dentro de una red social y compararlo contra otras pueden obtenerse resultados intención sobre la población, los cuales pueden no ser muy precisos ya que son datos crudos sin procesamiento de intencionalidad dentro del tweet, pero obtenemos una mera idea de lo que la población está hablando en las redes (en nuestro caso Twitter).

Estos resultados, por más básicos que sean, pueden ser utilizados como un indicador para las campañas electorales, tanto como para detectar la presencia de alguna figura política como para saber dentro de qué contexto se está hablando (seguridad por citar un caso), por parte de equipos de campaña para conocer qué puntos reforzar o atacar para mejorar la presencia del mismo en las redes.

Trabajo Futuro

Frontend

- Uno de los componentes a ser mejorados es la interfaz de usuario. Además de mejoras estéticas, se deben hacer mejoras relacionadas a la experiencia de usuario como por ejemplo una manera de que el usuario elimina las búsquedas realizadas sin que tenga que recargar la página.
- Poder seleccionar el rango de fechas donde realizar la consulta.

Backend

- En cuanto al backend quedan mejoras relacionadas a las búsquedas para hacer. Una posibilidad sería mostrar en una gráfica alterna, los resultados que se muestran hoy en día junto a resultados “similares” y con esto se refiere a alteraciones del objeto de búsqueda en una letra o carácter. Por ejemplo errores ortográficos, Vazquez y Vasquez se muestran en la misma gráfica.
- Mejorar los resultados y filtros de las búsquedas para que tenga en cuenta no sólo la fecha sino también la ubicación de los tweets. Es de interés saber no solo cuando se hizo referencia a cierto político sino desde donde. En cuanto a las fechas, es considerada en los resultados pero no en los filtros de búsqueda.
- Otra posible mejora es incorporar análisis semántico del contenido del tweet para poder desglosar la intención del tweet en sí y poderlo identificar como positivo, negativo o neutro.
- Acorde con el punto dos del listado de mejoras en frontend, soportar por parte de backend que se permita obtener los resultados dado un rango de fechas.

Referencias

1. Elastic search: <https://www.elastic.co/products/elasticsearch>
2. Twitter api: <https://developer.twitter.com/en/docs/tweets/search/overview/standard.html>
3. Angular: <https://angular.io/>
4. Spring: <https://spring.io/>
5. CanvasJS: <https://canvasjs.com/javascript-charts/multi-series-line-chart/>
6. Docker: <https://www.docker.com/>
7. Fuzzy search: https://en.wikipedia.org/wiki/Approximate_string_matching