

RECUPERACIÓN DE INFORMACIÓN Y RECOMENDACIONES EN LA WEB

Entrega Final

Juan Domínguez C.I.: 3.911.571-3
Mario Saul C.I. 4.545.715-5
Carolina Alvarez C.I 4.564.322-9
Malvina Betarte C.I. 4.817.127-3

ÍNDICE

ÍNDICE	2
Introducción	3
Problema	3
Enfoque de la solución	4
Diseño e Implementación	4
Conclusiones	9
Trabajo futuro	9
Referencias	10

1.Introducción

Hoy en día existen grupos abiertos en Facebook para publicar información de mascotas perdidas y encontradas. Generalmente se comparte una foto, la zona en la cual se perdió o fue encontrada y un teléfono de contacto.

Dada la gran cantidad de grupos existentes resulta útil disponer de la información centralizada en un único lugar, teniendo la posibilidad de buscar por aquellos filtros que uno considere relevantes, por ejemplo: animal (perro, gato), color, entre otros.

2.Problema

El extravío de mascotas es un problema común en nuestra sociedad. Las personas han acudido a las redes sociales para realizar publicaciones de mascotas perdidas o encontradas.

El trabajo consiste en centralizar los datos relevantes de publicaciones en grupos de Facebook de esta temática, facilitando de esta manera el acceso a dicha información a los usuarios interesados.

Para ello se decidió realizar un sistema con una interfaz web que permita la búsqueda, utilizando herramientas que mejoren la rapidez en la que se sirven los datos.

3.Enfoque de la solución

El enfoque de la solución que se plantea es recuperar información de determinados grupos abiertos de Facebook y publicarlos en una web junto a su imagen, lugar y fecha en la que se encontró o perdió la mascota, y el mensaje del post.

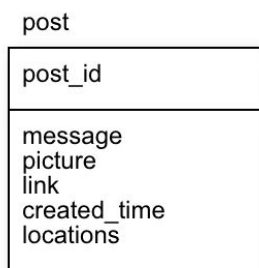
Se requieren búsquedas según determinados filtros que sean satisfechas rápidamente.

Vemos que hay una gran importancia en tener una variada fuente de entrada de datos, es decir tomar información de distintos grupos de Facebook, de distintos lugares del país.

4. Diseño e Implementación

El sistema cuenta con varios componentes:

- Base de datos MySQL donde se persisten los datos que consideramos relevantes de los posts.



- Elasticsearch¹ como motor de búsqueda de texto completo, distribuido y con capacidad de Stemming y Fuzzy matching para mejorar la precisión de los filtros sobre los posts con errores léxicos.
Elasticsearch es un servidor de búsqueda basado en Lucene. Provee un motor de búsqueda de texto completo, distribuido y con capacidad de multi-tenencia con una interfaz web RESTful y con documentos JSON. Elasticsearch está desarrollado en Java y está publicado como código abierto bajo las condiciones de la licencia Apache.
- Script en Ruby
Se tiene un script escrito en código Ruby, que realiza la conexión a la base de datos MySQL y crea la tabla de posts, obtiene las publicaciones de varios grupos de Facebook y los persiste en la base de datos. También se encarga de indexar los mismos en elasticsearch.
En este script se utilizan:
 - API de Facebook² que permite acceder directamente a los posts de grupos públicos.
 - ActiveRecord³ es el object relational mapper de Rails. Es utilizado para crear la base de datos, insertar los posts y mapear los datos a un modelo.
 - Koala facilita la conexión con la API de Facebook.

- Elasticsearch/model⁴: es una librería construída encima de la librería de elasticsearch. Su objetivo es simplificar la integración entre las clases de Ruby con el motor de búsqueda de Elasticsearch. En este script se realizó la carga del índice en elasticsearch.
- Grails⁵

Grails es un framework para aplicaciones web libre desarrollado sobre el lenguaje de programación Groovy (el cual a su vez se basa en la Java platform). Grails pretende ser un marco de trabajo altamente productivo siguiendo paradigmas tales como convención sobre configuración o no te repitas (DRY), proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador.

Con grails desarrollamos nuestra aplicación frontend, la cual cuenta con los filtros predeterminados: Lugar, Animal, Color , Fecha Desde y Hasta.

Lugar: seteamos a modo de ejemplo Montevideo, Canelones y Pando

Animal: opciones de gato y perro

Color: Negro, blanco y marrón

Y con la fecha obtenemos todos los posts que fueron publicados entre las dos fechas seleccionadas.

Esta aplicación se conecta con el motor de búsqueda de ElasticSearch para obtener la información de todos los posts que cumplen con los filtros seleccionados.



Se tomaron las siguientes decisiones en cuanto a la implementación:

Dado que la cantidad de grupos en Facebook de esta temática es muy amplio, se eligieron determinados grupos de los cuales obtener las publicaciones:

- [Animales Perdidos, Encontrados Y En Adopción -Uruguay](#)
- [Animales Perdidos en Montevideo](#)
- [Animales Perdidos Y Encontrados URUGUAY - APYEU Comunidad](#)
- [Animales Pando](#)
- [Animales abandonados](#)

En la aplicación web se cuenta con un filtro de lugar. Este lugar se hardcodea en el script ruby al momento de la carga de datos. Es decir, al saber en qué lugar geográfico opera el grupo de Facebook, fácilmente se puede saber el país, departamento y/o localidad.

BUSCANDO A MI MASCOTA

Buscar

Lugar

- ✓ Montevideo
- Canelones
- Pando

Perro

Color

Negro

Desde

28 noviembre 2017

Hasta

28 noviembre 2017

Buscar

A su vez, la aplicación web se comunica con el motor de búsqueda Elasticsearch para realizar las consultas según los filtros seleccionados.

Esto se implementa mediante una llamada POST a la dirección http://localhost:9200/_search, con el JSON de la consulta deseada.

Por ejemplo si se desea obtener los gatos de color negro que se perdieron o encontraron dentro de Montevideo para la fecha 28/11/2017 la consulta en elasticsearch sería la siguiente:

```
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "message": "negro"
          }
        }
      ],
    }
  }
}
```


```

{
  "match": {
    "locations": "montevideo"
  }
},
{
  "match": {
    "message": "gato"
  }
},
],
"filter": {
  "range": {"created_time": {
    "gte": "2017-11-28T00:00:00",
    "lte": "2017-11-28T23:59:59",
    "format": "yyyy-MM-dd HH:mm:ss"
  }}
}
}
}
}
}

```

La consulta es una combinación de una consulta Bool Query con Range Query. En la consulta se buscó por los campos message, locations y created_time. Para el campo message se busca que exista los strings “gato” y “negro”; para locations se busca que exista el string “montevideo” y para el caso del campo created_time se busca que las fechas esté entre las dos fechas establecidas. Elasticsearch retorna todos los posts que coincidan con la consulta.

A continuación mostramos una imagen del sistema al realizar la consulta de búsqueda de gatos negros en Montevideo en determinado rango de fechas:



BUSCANDO A MI MASCOTA

Buscar

Lugar

Animal

Color

Desde

Hasta

Gata negra encontrada en Juan Carlos Gómez entre Cerrito y 25 de Mayo. Ahora está en transitorio. Agradezco difusión para encontrar a su filia! Tiene un collar rosa, con moñita y cascabel.

Se busca urgente (MONTEVIDEO) Zona cordón, se rompió un vidrio y se escapó por ahí. Es semi salvaje rescatada de una colonia de gatos. Si bien se ha domesticado bastante y en casa tranquila se deja acariciar, no se deja agarrar y es muy asustadiza. No sabe estar en la calle. Tiene el collar de la foto rosado y bnco y un collar antipulgas. Esta castrada y tiene 11 meses. Por favor compartir!

5. Conclusiones

Las redes sociales son la herramienta más utilizada a la hora de buscar una mascota perdida o encontrarle hogar a una mascota encontrada. En particular, los grupos abiertos de Facebook son los preferidos. Observamos que si existe la necesidad de centralizar esta información y que es posible hacerlo de una manera simple.

La recuperación de la información de grupos de Facebook se puede hacer de una manera sencilla utilizando las APIs que provee.

Elasticsearch es una herramienta muy potente, que facilita y agiliza las búsquedas. Tiene una curva de aprendizaje, sobre todo en la integración con otros sistemas.

6. Trabajo futuro

Como trabajo a futuro observamos que sería útil contar con un módulo de procesamiento de lenguaje natural para que dado el mensaje se puedan obtener

datos más específicos del post. Por ejemplo, el lugar en donde se perdió o encontró (dirección, número de puerta, barrio, ciudad).

Dado que se toman posts de varios grupos de Facebook, puede suceder que éstos se repitan. Es decir que una misma persona publique un post en varios grupos. Por lo tanto sería deseable tener un mecanismo para unificar estos posts y así no tener información repetida.

Una mejora en el sistema sería facilitar el agregado de nuevos grupos de Facebook de los cuales obtener publicaciones.

Dado que es solo un prototipo la carga de datos se hace manual ejecutando scripts. Esto se podría automatizar utilizando alguna herramienta como ser Cron⁶, encargada de ejecutar cada un día por ejemplo los scripts que actualicen los datos (en caso de que estos cambien) y agreguen nuevos posts a la base de datos y motor de búsqueda.

7. Referencias

[1] Elasticsearch - <https://www.elastic.co/products/elasticsearch>

[2] API Facebook -

<https://developers.facebook.com/docs/graph-api/reference/v2.10/group/feed>

[3] ActiveRecord - <https://github.com/rails/rails/tree/master/activerecord>

[4] Elasticsearch/model -

<https://github.com/elastic/elasticsearch-rails/tree/master/elasticsearch-model>

[5] Grails - <https://grails.org/>

[6] Cron - <https://github.com/javan/whenever>

