Captura de datos de Internet

Gonzalo Rivero

Big Data en investigación social y opinión pública

22 a 26 de julio, 2019

Captura de datos de Internet

- REST API
  - El modo más sencillo
  - Alguien ha decidido ofrecer los datos para ser consumidos por una aplicación
- Captura de web (web scrapping, web harvesting)
  - Situación más habitual
  - Extraeremos el contenido de la página y recuperaremos las piezas que nos interesen
- Streaming
  - Tarea especializada
  - Nos conectaremos a una fuente que emite los datos en tiempo real

Una introducción a REST API

- REST es uno de los principios arquitectonicos de la Web.
- Define un modo de interactuar entre clientes (navegadores) y servidores sin que los agentes sepan nada sobre el otro. La unica restricción es que se comunican sobre HTTP.
- El servidor provee al cliente de la localización de un recurso y de los campos que son necesarios.

Caracteristicas de una REST API

- Los recursos tienen un identificador único
- Los recursos se manipulan con verbos
  - `GET`: Obtener datos
  - `POST`: Crear datos/Pedir ejecución
  - `DELETE`: Borrar datos
  - `PUT`: Actualizar datos
- El recurso viene devuelto en XML o JSON

Ejemplo de un recurso

- Un servicio para solicitar libros
- Los datos están organizados en clientes, que contienen determinadas peticiones que a su vez incluyen libros. Por ejemplo,

```
http://server.test:8080/order_api/{customer_id}/{order_id}/{book_id}
```

- Enviar un `DELETE` podría eliminar un libro de una orden
- Enviar un `GET` recuperaría los detalles de un libro (como el título)

Contenido del encabezado

- Reservado para parámetros que definen la transacción
- No debe usarse para pasar parámetros o datos
- Era convención que los campos no-estándar empiecen con `X-`

Campo	Descripción	Ejemplo
`Accept`	Tipo de contenido	`text/plain`
`Accept-Charset`	Grupo de caracteres	`utf-8`
`Accept-Encoding`	Codificación	`gzip`
`Cookie`	Cookie	
`Content-Type`	MIME del cuerpo	`application/json`
`Date`	Fecha del envío	
`Authorization`	Credenciales	

Respuesta de una petición

- Respuesta incluye un código y la carga (*payload*) en JSON o XML
- La carga es específica a la aplicación (datos y a veces metadatos)

Código	Mensaje	Descripción
200	OK	Éxito
400	Bad Request	Error en la petición
401	Unauthorized	Credenciales son incorrectas
403	Forbidden	Límite de API o datos privados
404	Not Found	URL está mal
500	Server Error	El servidor ha fallado
503	Service Unavailable	El servidor está caído

Un ejemplo de JSON

```
{
  "primerNombre": "Gonzalo",
  "edad": 36,
  "parientes": [ "Julia", "Jose" ]
}
```

Tres campos identificados con comillas dobles. Valor de campo puede ser texto, número, array, constante lógica, o `null`.

Un ejemplo de XML

```
<persona>
    <primerNombre>Gonzalo</primerNombre>
    <edad>36</edad>
    <parientes>
        <pariente>Jose</pariente>
        <pariente>Julia</pariente>
    </parientes>
</persona>
```

Tres campos identificados cerrados con etiquetas de apertura y cierre. Un array es un nuevo objeto anidado.

Introducción a HTML

- Es el lenguaje que los navegadores interpretan para mostrarnos una página web.
- HTML describe la estructura de la página
- Cada elemento consiste de *tags* de apertura y cierra y contenido. Generalmente vienen en pares, pero no siempre. Pueden no tener contenido.
- Los tags de apertura pueden contener atributos:
  - Los atributos da información adicional sobre el elemento.
  - Los atributos tienen un nombre y un valor
  - El valor generalmente viene entre comillas

Un ejemplo de HTML

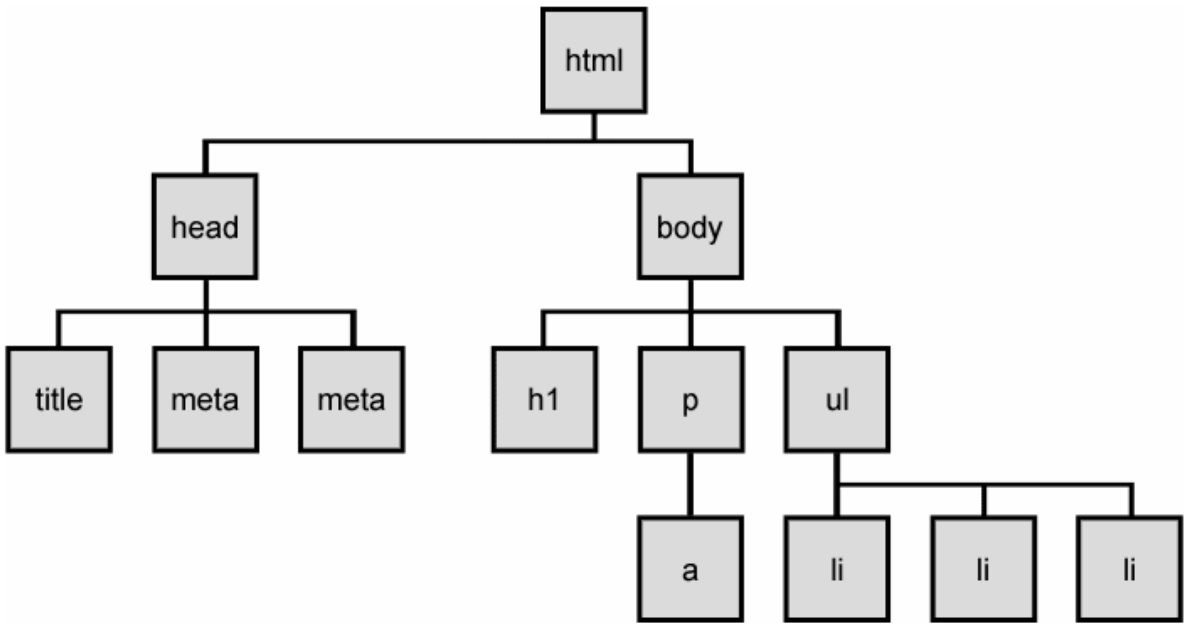
```
<p class="fruta">manzana</p>
```

El contenido es `manzana` dentro de una etiqueta `p` (párrafo) cerrado con `</p>` que tiene atributo `class` con valor `fruta`.

El Document Objet Model

- Define el modelo con la jerarquía de elementos que existen en la página actual
- El elemento superior del modelo siempre es el `document` que representa la página en su totalidad
- Cada hijo representa otros elementos en la página y van anidados
- El DOM provee de métodos para atravesar la página (no los veremos)

Un ejemplo de DOM



La página contiene un encabezamiento (`head`) con metadatos y un cuerpo (`body`) con la parte que veremos en el navegador. El cuerpo contiene un encabezado (`h1`), un párrafo que contiene un enlace (`a`), y una lista desordenada (`ul`) con tres elementos (`il`).

Un ejemplo de página web

```
<!DOCTYPE html>
  <html>
    <head>
      <title>Titulo de la pagina</title>
        <link rel='stylesheet' href='css/style.css' />
    </head>
  <body>

  <section id='page1'>
    <h1>Un titulo</h1>
     <img src='images/fruta.png' id='regalo' />
     <section id='mesa'></section>
    </section>
  </body>
</html>
```

Un ejemplo de una lista

```
<html>
  <body>
    <div id='page'>
      <h1 id='header'>Lista</h1>
        <ul>
          <li id='uno' class='fruta'><em>manzana</em></li>
          <li id='dos' class='fruta'>pera</li>
          <li id='tres' class='higiene'>peine</li>
        </ul>
    </div>
  </body>
</html>
```

Un ejemplo de una tabla

```
<table style="width:100%">
  <tr>
    <th>Nombre</th>
    <th>Apellido</th>
    <th>Edad</th>
  </tr>
  <tr>
    <td>Gonzalo</td>
    <td>Rivero</td>
    <td>36</td>
  </tr>
  <tr>
    <td>Antonio</td>
    <td>Pérez</td>
    <td>29</td>
  </tr>
</table>
```

Un ejemplo de un enlace

```
<a href="http://www.griverorz.net">Visita mi web</a>
```

Introducción a CSS

- El CSS define las reglas que definen cómo se muestran los elementos en el navegador.
- HMTL es la estructura, CSS es la estética
- Cada regla tiene un selector y un bloque de declaración
- El selector indica a qué elementos se aplica y la declaración indica cómo se aplican los elementos
- Nos interesa por que a veces identifica los elementos que queremos

Un ejemplo de CSS

```
<p class="fruta">manzana</p>
```

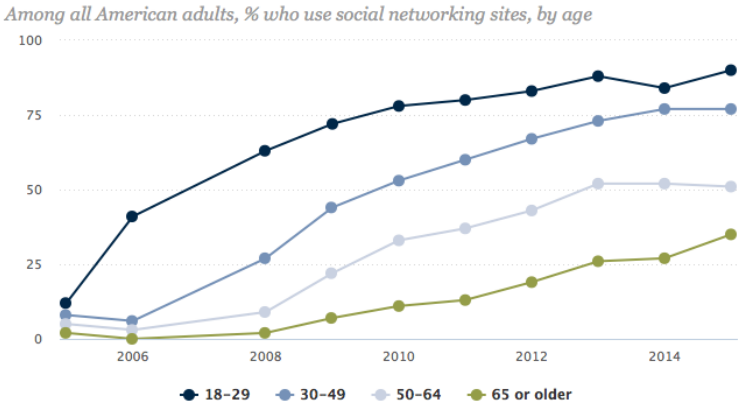
Las características del atributo `fruta` se modifican en el CSS

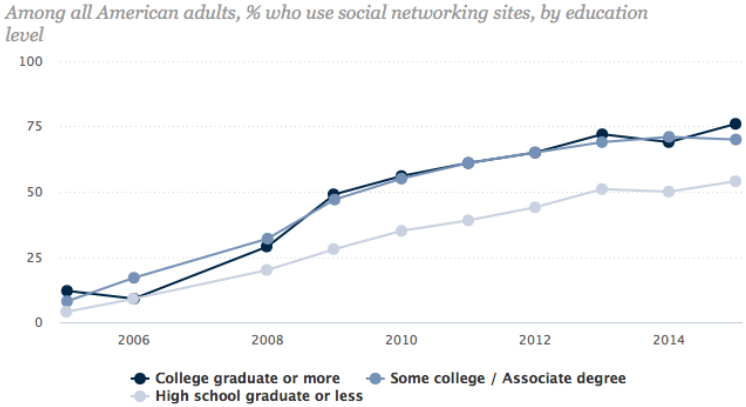
```
.fruta{color: verde;}
```

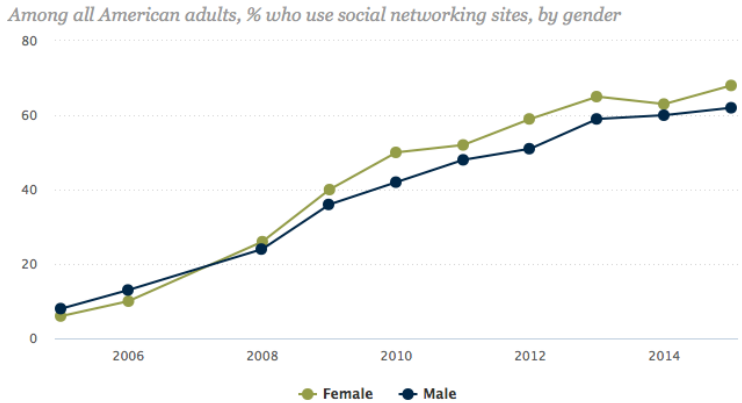
Retos de la investigación en redes sociales

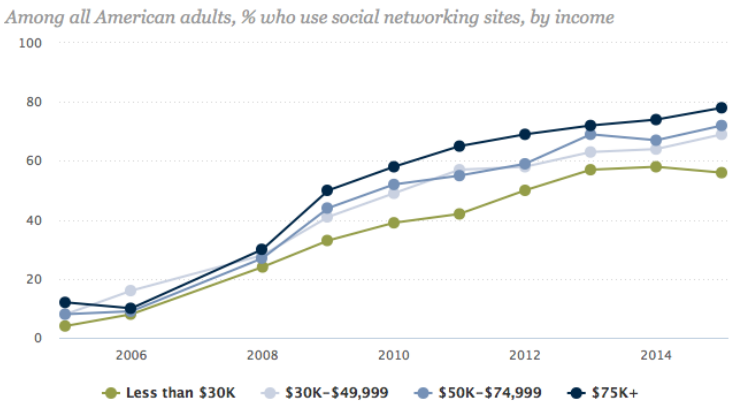
- Es recolección no intrusiva de datos generados por los usuarios
- Naturaleza de los datos se refiere a interacciones entre usuarios
- Enorme crecimiento de base de usuarios y más actividades en redes sociales

Evolución del uso









Crecimiento sostenido con mayor diferencia en variación por edad.

Complicaciones en el uso de los datos

- El uso no siempre es gratuito
- Representatividad únicamente de la plataforma sabiéndose observados
- Validez puede no extenderse al mundo fuera de linea
- No hay consentimiento de investigación

Diurnal and Seasonal Mood Patterns

SHARE

0

0

REPORT

Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures

Scott A. Golder*, Michael W. Macy
+ See all authors and affiliations

Science 30 Sep 2011:
Vol. 333, Issue 6051, pp. 1878-1881
DOI: 10.1126/science.1202775

Article | Figures & Data | Info & Metrics | eLetters | PDF

You are currently viewing the abstract.

View Full Text

Abstract

We identified individual-level diurnal and seasonal mood rhythms in cultures across the globe, using data from millions of public Twitter messages. We found that individuals awaken in a good mood that deteriorates as the day progresses—which is consistent with the effects of sleep and circadian rhythm—and that seasonal change in baseline positive affect varies with change in daylength. People are happier on weekends, but the morning peak in positive affect is delayed by 2 hours, which suggests that people awaken later on weekends.

Diurnal and Seasonal Mood Patterns

- Pregunta:
  - Cómo cambia el mood a lo largo del día, estaciones y culturas.
- Datos:
  - Twiter API
  - 509 millones de mensajes de 2.4 millones de personas
  - Febrero 2008 a Enero 2010
- Métodos:
  - Comparación de medias

A 61-million-person Experiment

A 61-million-person experiment in social influence and political mobilization

Robert M. Bond, Christopher J. Fariss, Jason J. Jones, Adam D. I. Kramer, Cameron Marlow, Jaime E. Settle & James H. Fowler

Affiliations | **Contributions** | **Corresponding author**

Nature **489**, 295–298 (13 September 2012) | doi:10.1038/nature11421
Received 17 April 2012 | Accepted 18 July 2012 | Published online 12 September 2012

Human behaviour is thought to spread through face-to-face social networks, but it is difficult to identify social influence effects in observational studies[9, 10, 11, 12, 13], and it is unknown whether online social networks operate in the same way[14–19]. Here we report results from a randomized controlled trial of political mobilization messages delivered to 61 million Facebook users during the 2010 US congressional elections. The results show that the messages directly influenced political self-expression, information seeking and real-world voting behaviour of millions of people. Furthermore, the messages not only influenced the users who received them but also the users' friends, and friends of friends. The effect of social transmission on real-world voting was greater than the direct effect of the messages themselves, and nearly all the transmission occurred between 'close friends' who were more likely to have a face-to-face relationship. These results suggest that strong ties are instrumental for spreading both online and real-world behaviour in human social networks.

A 61-million-person Experiment

- Pregunta:
  - ¿Ver que un amigo ha votado afecta a mi probabilidad de votar?
- Datos:
  - Experimento sobre los muros de Facebook
  - 61 millones de usuarios
  - Elecciones al Congreso de 2010
- Métodos:
  - Tratamiento-control
  - Comparación de medias