



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IDENTIFICACIÓN DE CONTENIDO MULTIMEDIA RELEVANTE A PARTIR DE  
EVENTOS UTILIZANDO SU INFORMACIÓN SOCIAL

TESIS PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

MAURICIO DANIEL QUEZADA VEAS

PROFESORA GUÍA:  
BÁRBARA POBLETE LABRA

MIEMBROS DE LA COMISIÓN:  
SERGIO OCHOA DELORENZI  
MAURICIO MARÍN CAIHUAN

SANTIAGO DE CHILE  
DICIEMBRE 2012



# Resumen

asdf



*Jason Funk disipa patitos*

# Agradecimientos

asdf

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	2
1.2. Motivación . . . . .	3
1.3. Contribuciones . . . . .	3
1.4. Alternativas analizadas . . . . .	3
1.5. Objetivos . . . . .	3
1.5.1. Objetivo general . . . . .	3
1.5.2. Objetivos específicos . . . . .	3
1.6. Descripción general de la solución . . . . .	3
1.7. Resultados obtenidos . . . . .	3
<b>2. Antecedentes</b>	<b>4</b>
2.1. Marco teórico . . . . .	4
2.2. Soluciones existentes . . . . .	4
<b>3. Especificación del Problema</b>	<b>5</b>
3.1. Descripción detallada . . . . .	5
3.2. Relevancia de una solución . . . . .	5
3.3. Características de calidad . . . . .	5
3.4. Criterios de aceptación . . . . .	5
<b>4. Descripción de la Solución</b>	<b>6</b>
4.1. Desafíos técnicos . . . . .	6
4.2. Metodología de desarrollo . . . . .	6
4.3. Casos de estudio . . . . .	6
4.4. Validación . . . . .	6
<b>5. Conclusiones</b>	<b>7</b>
5.1. Resumen del trabajo realizado . . . . .	7
5.2. Objetivos alcanzados . . . . .	7
5.3. Relevancia del trabajo realizado . . . . .	7
5.4. Trabajo futuro . . . . .	7
<b>6. Unclassified D:</b>	<b>8</b>
6.1. Twitter . . . . .	8
6.2. Metodología de obtencion del dataset . . . . .	9
6.2.1. Recolección de eventos . . . . .	9

6.2.2.	Enriquecimiento de eventos . . . . .	10
6.2.3.	Identificación de documentos a partir de tweets . . . . .	10
6.3.	Performance . . . . .	11
6.4.	Restricciones de la API de Twitter . . . . .	11



# Capítulo 1

## Introducción

Al igual que en el buffet de un restaurante, por mucho que se quisieran comer todos los platos favoritos, es imposible comer todo lo que uno quisiera por razones obvias. Una posibilidad es probar un poco de cada comida, para así saber qué es lo más delicioso y comer hasta hartarse.

Pero, ¿qué hacer si hay demasiados platos y no se conocen todos? de alguna manera hay que saber cuáles hay que probar, si el objetivo es comer lo mejor posible. Un amigo puede recomendar una u otra comida, lo cual puede servir para orientarse. Entonces se pueden escoger pequeñas muestras de acuerdo a las recomendaciones.

Pasando a un contexto diferente, supóngase que este gran buffet es la Web y los distintos platos corresponden a contenido publicado en ella. Por lo tanto, dada la gran cantidad de información disponible, se hace necesario poder encontrar lo más atractivo de acuerdo a la preferencia del usuario o de los usuarios. Nótese que se está haciendo otra suposición importante con esta analogía, y es que se está considerando que la información es íntegramente para ser *consumida*, y no, por ejemplo, para generar más contenido, conocimiento, o para ser utilizada por máquinas, etc. Dentro de este contexto se plantea la pregunta de cómo seleccionar el contenido más atractivo dentro de todo lo que hay disponible en un momento dado.

Siguiendo el razonamiento de la analogía, una manera de poder seleccionar sólo el contenido más “atractivo” (de acuerdo a las preferencias del usuario), es probar un poco de cada uno. Diversos esfuerzos han sido hechos para este propósito, entre ellos está el de generar resúmenes automáticos<sup>1</sup> a partir de uno o múltiples documentos. Luego, utilizando las recomendaciones de otros usuarios es posible ordenar estos resultados de acuerdo a la relevancia que éstos les dan.

Este trabajo consistió en el desarrollo de un sistema que permite generar resúmenes automáticos de contenido general (*multimedios*: texto, imágenes, vídeos, sonidos) a partir de eventos bien definidos. El contenido es filtrado o seleccionado de acuerdo a indicadores sociales: los objetos más comentados en la Web son considerados más importantes.

---

<sup>1</sup>??

El sistema implementado consideró dos tipos de eventos: noticias y conciertos musicales. Para obtenerlos, se utilizó el servicio de Google News<sup>2</sup> y Last.fm<sup>3</sup>. Para medir la relevancia de los documentos y obtener los mismos se utilizó la red social Twitter<sup>4</sup>, que provee una *Application Programming Interface* (o API) para realizar búsquedas y obtener información sobre los *tweets* o mensajes cortos que publican los usuarios del servicio.

## 1.1. Contexto

La tasa de crecimiento de la cantidad de datos en la red, y en particular, de las *redes sociales online* (OSN, *Online Social Networks*), es de tal magnitud que se vuelve necesario encontrar formas de filtrar y buscar sólo la información relevante dentro de todas las fuentes que hablan del mismo tópico o tema.

En el contexto de las redes sociales online, cada día se publican millones de *actualizaciones de estado* (mensajes breves sobre el estado actual del usuario) con respecto a distintos tópicos, ya sean conversacionales, personales o sobre algún evento en particular<sup>5</sup>. Además, el auge de los teléfonos inteligentes o *smartphones* con mayor capacidad de procesamiento e integrados con todo tipo de sensores (cámaras fotográficas, de vídeo, acelerómetro, barómetro, osciloscopio, etc.), hace posible el generar aun más información y en tiempo real con respecto a eventos que suceden en el mundo, o bien con respecto al estado particular de los usuarios de estas redes y dispositivos.

Este aumento de la generación de datos no sólo influye en la riqueza en la variedad de éstos, sino también en el comportamiento de los usuarios a lo largo del tiempo. Actualmente, una gran parte de los usuarios valora más el contenido multimedia en las redes sociales online.

---

<sup>2</sup><http://news.google.com>

<sup>3</sup><http://last.fm>

<sup>4</sup><http://twitter.com>

<sup>5</sup>Pear Analytics. Twitter Study TODO!!!!

**1.2. Motivación**

**1.3. Contribuciones**

**1.4. Alternativas analizadas**

**1.5. Objetivos**

**1.5.1. Objetivo general**

**1.5.2. Objetivos específicos**

**1.6. Descripción general de la solución**

**1.7. Resultados obtenidos**

# Capítulo 2

## Antecedentes

2.1. Marco teórico

2.2. Soluciones existentes

## Capítulo 3

### Especificación del Problema

- 3.1. Descripción detallada
- 3.2. Relevancia de una solución
- 3.3. Características de calidad
- 3.4. Criterios de aceptación

# Capítulo 4

## Descripción de la Solución

- 4.1. Desafíos técnicos
- 4.2. Metodología de desarrollo
- 4.3. Casos de estudio
- 4.4. Validación

# Capítulo 5

## Conclusiones

5.1. Resumen del trabajo realizado

5.2. Objetivos alcanzados

5.3. Relevancia del trabajo realizado

5.4. Trabajo futuro

# Capítulo 6

## Unclassified D:

### 6.1. Twitter

Twitter es una red social online que permite conectar a personas mediante la comunicación de mensajes cortos, rápidos y frecuentes. Estos mensajes son publicados en el perfil del usuario que los emite, pueden ser vistos directamente por los seguidores de este usuario o ser vistos directamente en el perfil o buscándolos mediante una funcionalidad que provee el servicio. Además, un usuario puede *seguir* a otros para poder ver en su *timeline* los mensajes de todos a quienes sigue.

Estos mensajes, o *tweets*, pueden además *mencionar* a otros usuarios, mediante la convención “@usuario” que indica que se está mencionando a la persona con el nombre “usuario”. Adicionalmente, existen varias convenciones o costumbres que han surgido a lo largo del tiempo en esta red:

- Respuestas o *replies*: son mensajes del tipo @usuario [texto]..., que ocurren usualmente en una conversación entre dos usuarios.
- Menciones o *mentions*: un poco más general a una respuesta, el nombre del usuario mencionado puede estar en cualquier parte del mensaje. La diferencia semántica es que no se le habla “directamente” al usuario mencionado, como en una respuesta, sino que sólo es mencionado por si el mensajes es de su interés o no.
- *Retweets*: son mensajes del tipo RT @usuario: [texto]. Ocurren cuando se quiere compartir el mensaje de otro usuario, o citarlo para mencionarlo en el mismo mensaje.
- *Hashtags*: son palabras precedidas por el caracter #, que indican un identificador a cierto evento o suceso dentro o fuera de la red. Suelen usarse para categorizar de cierta forma un tópico, pero son libres de usarse como los usuarios quieran.
- Mensaje simple: un mensaje sin menciones ni hashtags.

Ejemplos:

- Mensaje simple: Jason Funk disipa patitos;



- Respuesta: @jason estoy de acuerdo con lo que dices;
- Mención: creo que @jason es una cumbre de sabiduría;
- Retweet: RT @jason: Jason Funk disipa patitos; y
- Hashtag: Estoy escribiendo mi memoria #dcc #summarization

## 6.2. Metodología de obtencion del dataset

Se describe a continuación el proceso diseñado para la obtención de datos para alimentar al sistema implementado.

Las etapas de generación del Dataset son las siguientes:

- Recolección de eventos (noticias y conciertos);
- Enriquecimiento de los eventos existentes mediante tweets; e
- Identificación de documentos a partir de los tweets por cada evento.

Se recolectaron datos (eventos y tweets) desde el 19 de noviembre de 2012 hasta XXXXXXXXXXXX todos los días desde la medianoche hasta que el procedimiento terminaba exitosamente.

### 6.2.1. Recolección de eventos

Se consideraron dos tipos de eventos para el sistema: noticias y conciertos musicales. Los conciertos incluyen festivales de varios artistas.

- Noticias Para obtener las noticias, se utilizó el servicio de Google News<sup>1</sup>. Existe una API (en proceso de obsolescencia, pero funcional a la fecha de este trabajo) que permite obtener no sólo los titulares y breve descripción de cada noticia, sino también un conjunto de entre 4-10 noticias relacionadas de otras fuentes. Esto sirvió para alimentar los términos de búsqueda para la etapa siguiente. Se guardaron los siguientes datos de una noticia:
  - Título,
  - Descripción,
  - URL de la fuente, y
  - Titulares de las noticias relacionadas.
- Conciertos Utilizando el servicio de Last.fm para obtener los conciertos y festivales de una ubicación en particular<sup>2</sup>, se obtuvieron los conciertos y festivales de las siguientes ubicaciones:
  - Santiago, Chile;
  - Londres, Inglaterra;

---

<sup>1</sup><http://news.google.com>

<sup>2</sup><http://www.lastfm.es/api/show/geo.getEvents>

- Glastonbury, Inglaterra;
- Las Vegas, Nevada, EE.UU.; y
- Estocolmo, Suecia.
- Título del evento (concierto o festival);
- Artistas que participan; y
- Fechas de inicio y término (esta última no siempre está como dato).

Además de otros datos descriptivos, como la ubicación, descripción breve, sitio web de la banda o festival, etc.

Cada vez que se obtienen los eventos se vuelven a obtener los conciertos, pero sólo agregando los nuevos. Las noticias siempre son nuevas, aun así por implementación no se consideraron los repetidos.

### 6.2.2. Enriquecimiento de eventos

Se obtuvieron tweets utilizando el servicio de búsqueda que provee Twitter en su API<sup>3</sup>. El objetivo es enriquecer los eventos con la información social que hay en la Web sobre éstos.

Para cada uno de los eventos obtenidos en la fase anterior, se utilizaron los términos de búsqueda asociados a ellos: los titulares de las noticias relacionadas y los nombres de los artistas para los eventos noticiosos y musicales, respectivamente.

- Para las noticias, se hace una búsqueda en Twitter de los titulares al mismo tiempo en que se obtienen de Google News, y nuevamente al día siguiente, es decir, 2 búsquedas por cada titular de un evento. Se quitan las tildes y caracteres no ASCII y las stopwords, para evitar problemas con la implementación y no hacer calce de stopwords en la búsqueda de Twitter, respectivamente.
- Para los conciertos y festivales, se utilizaron los nombres de los artistas y del evento como términos de búsqueda. De acuerdo a la información asociada al evento, se busca por una mayor cantidad de días:
  - Se busca desde un día antes de inicio del evento;
  - Si está presente la fecha de término del evento, se busca cada día dentro del intervalo “fecha de inicio” a “fecha de término” hasta tres días terminado el evento.
  - Si no está presente la fecha de término (por ejemplo, un concierto o un festival de un día), se busca hasta tres días pasada la fecha de inicio.

### 6.2.3. Identificación de documentos a partir de tweets

Luego de obtener los tweets asociados a cada evento, el siguiente paso fue generar los documentos que fueron usados para la generación de los resúmenes. Nuevamente, el modelo consistió en que cada documento se modeló como un vector de palabras, donde el identificador

---

<sup>3</sup><https://dev.twitter.com/docs/api/1.1/get/search/tweets>

del documento es una URL, y sus componentes corresponden al contenido de los tweets que tienen esa URL en el texto del mensaje.

El caso en el que un tweet no tenía ninguna URL en su contenido fue abordado de la siguiente forma: la URL asociada es una tal que representa al mismo tweet (utilizando el servicio de Twitter), y el contenido de ese documento es el mismo tweet, de forma de no dejar el tweet sin ser representado.

Este proceso fue abordado recorriendo todos los eventos del dataset, observando todos los tweets asociados a cada evento, extrayendo la URL si es que hay alguna y guardando el documento con el nuevo tweet. Se marcan los tweets observados para no tener que repetir el proceso, ya que es intensivo en conexión a la red.

Dada la condición breve de los mensajes publicados en la red social, muchos de los usuarios y/o servicios que publican mensajes con una URL en su interior suelen utilizar *acortadores* (*url shorteners*) para los enlaces, y así no utilizar mucho espacio dentro de un mensaje. Otra ventaja que ofrecen es que algunos servicios como bit.ly dan estadísticas sobre los visitantes a estos enlaces (y así saber quiénes vienen de cierta red social u otra, por ejemplo). Twitter, a su vez, actualmente también ofrece acortamiento de URLs por defecto. Esto suele producir que un enlace acortado se resuelva a otro enlace también acortado, por lo que es necesario resolver la URL completa para evitar duplicados o *pseudo-duplicados* (en el caso en que dos URLs sintácticamente distintas apunten al mismo recurso). EN LA FIGURA. . . . .

#### FIGURA DE LINKS CORTOS

Por lo anterior, una vez identificada la URL del texto de un tweet, se resuelve su URL completa (que puede ya serlo de antemano), lo que consume recursos de ancho de banda y tiempo.

### 6.3. Performance

Tiempo? espacio? por evento?

### 6.4. Restricciones de la API de Twitter

La API de búsqueda de Twitter permite obtener tweets de acuerdo a un término de búsqueda. Se utilizó este servicio para enriquecer los eventos con información social utilizando como términos de búsqueda tanto los títulos de las noticias como los nombres de los artistas para las noticias y los conciertos, respectivamente.

Funciona de la siguiente forma: cada vez que se hace un request a la URL dada por el servicio, éste retorna a lo más 100 tweets por página, con un máximo de 15 páginas (indicando en el request qué página queremos consultar), dando como total hasta 1500 tweets por búsqueda. Existirán términos de búsqueda que no presenten ningún resultado (ya sea por

estar mal escritos o simplemente que no sean un tópico de discusión), o por el contrario, que se generen más tweets que los retornados por la búsqueda por cada ventana de tiempo que demore ésta (por ejemplo, un *trending topic* o tópico que sea muy mencionado en la red social).

Existe una limitación de uso de este servicio: sólo es posible hacer hasta 180 requests por cada 15 minutos, o 1 request cada 5 segundos. Además, sólo retorna tweets de hasta 7 días de antigüedad, y sus resultados no son necesariamente en tiempo real y su estabilidad varía de acuerdo a factores externos.

Los tweets retornados vienen en formato JSON (*Javascript Simple Object Notation*), e incluyen varios metadatos sobre el tweet aparte de los principales, como autor, fecha, contenido. Algunos de estos metadatos son:

- Cantidad de *retweets* hechos hasta la fecha;
- Si posee alguna URL o *hashtag* en el texto;
- Si es una *mención* a otro usuario;
- La ubicación de donde se envió el tweet;
- etc.

Además incluye datos sobre el autor, como por ejemplo:

- Si la cuenta está *verificada*;
- La cantidad de seguidores del usuario;
- Cantidad de amigos (seguidores que también lo siguen);
- Cantidad de tweets;
- Su descripción, y si incluye alguna URL, etc;
- Ubicación (dada por el mismo usuario);
- Fecha de creación de la cuenta;
- etc.

# Bibliografía

- [1] Ioannis Karatzas. and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, Berlin, 2nd edition, 2000.
- [2] Philip Protter. *Stochastic Integration and Differential Equations*. Springer, 1990.
- [3] Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*. Number 293 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin [u.a.], 3. ed edition, 1999.