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

VISUALIZACIÓN ESPACIO/TEMPORAL DE EVENTOS NOTICIOSOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

CAMILO PABLO ANTONIO PALMA PRADENA

PROFESOR GUÍA:
BÁRBARA POBLETE LABRA

MIEMBROS DE LA COMISIÓN:
JOSÉ PINO URTUBIA
ANDRÉS MUÑOZ ÓRDENES

SANTIAGO DE CHILE
ABRIL 2013

Resumen Ejecutivo

Todos los días y a toda hora ocurren eventos en el mundo. Los eventos importantes pasan a ser noticia y generan contenido en la Web. La motivación de este trabajo es visualizar este tipo de datos, mediante la construcción de una aplicación, para saber qué ocurre en un determinado intervalo de tiempo y a su vez utilizar esta herramienta para investigación de datos estructurados de noticias.

Para ello fue necesario investigar cómo visualizar eventos, para entregar de manera efectiva la información de estas noticias. También, analizar las distintas fuentes de datos de noticias disponibles en la web, como Google News y Feedzilla, para poder recolectar datos y validar una solución. Además, se modeló un diseño de base de datos que permite unificar la estructura de los datos de distintas fuentes.

Se desarrolló una aplicación web llamada Eventsvis capaz de visualizar datos estructurados de noticias, en la que se muestra la relevancia, ubicación, categoría y fecha en la que ocurren estas noticias. También, se desarrollaron dos módulos capaces de recolectar noticias tanto de la API de Google News como de la API de Feedzilla. Además, esta información se enriqueció con información adicional que no era entregada por las fuentes, como la ubicación de las noticias y keywords del contenido de estas. Finalmente se pudo automatizar la recolección de noticias mediante un cron.

Como resultado de la solución desarrollada, se pudo obtener información adicional de los conjuntos de datos solamente observando la visualización. Se determinó que ciertas categorías son más difíciles de georeferenciar que otras, también que en ciertos intervalos de tiempo no se recolectaban datos y que ciertas ubicaciones poseen más noticias que otras. Al mismo tiempo, se construye una herramienta capaz de mostrar los eventos ocurridos en un intervalo de tiempo.

Finalmente, se discute cómo sería posible extender este trabajo o reutilizar este trabajo para utilizar otras fuentes de datos.

A mis padres y amigos.

Agradecimientos

Me gustaría comenzar agradeciendo a mis padres, gracias a ellos fue posible comenzar a estudiar en la Universidad de Chile y finalmente poder terminar aquí con este trabajo.

También me gustaría agradecer a mis amigos, tanto de la universidad como de la vida, quienes hicieron mi paso por la universidad asombroso e inolvidable. Gracias también por apoyarme y estar al tanto de este trabajo.

Finalmente, a mi profesora guía Bárbara Poblete, quien siempre tuvo tiempo para mí y me motivó a realizar esta memoria sobre visualización de datos.

Tabla de Contenido

Índice de tablas

Índice de figuras

Introducción

Día a día, ocurren distintos eventos en el mundo. Estos eventos pueden ser de tópicos muy variados, por ejemplo, el concierto de una banda en alguna ciudad, las medidas económicas adoptadas por un país, el lanzamiento de un nuevo smartphone, etc. Cada uno de estos eventos genera contenido en Internet, como videos, fotos, audio, noticias, reseñas en blogs. Parte de la información anterior, o al menos la considerada más relevante como las noticias, puede ser obtenida de forma estructurada por medio de *APIs* (Application Programming Interfaces). Las APIs proveen mensajes que pueden ser interpretados por una aplicación web, siendo estos mensajes contestados con información estructurada.

¿Cómo se podría inspeccionar este tipo de datos? ¿Cómo se podría navegar a través de estos datos? Una visualización de datos permite lo anterior, y trae consigo varios desafíos debido a las características de los datos. Los datos de eventos noticiosos poseen varios atributos como fecha, ubicación, categoría, relevancia, contenido, imágenes y más. El desafío está en representar todos los atributos sin tener una visualización sobrecodificada, es decir, representar la mayor cantidad de atributos posible sin que la visualización se haga inentendible.

Por otro lado, también está el desafío de representar ubicación geográfica y ubicación temporal al mismo tiempo. Además, se quiere visualizar la mayor cantidad de noticias sin que la cantidad de estas genere una visualización sobrecargada que no aporta información, porque si bien la cantidad de datos no genera problemas para que estos sean procesados, la cantidad de datos es grande para ser visualizada. Por último, la usabilidad en el área de visualización de datos es un problema que se considera difícil, por la dificultad para generar interacciones efectivas.

El objetivo de este trabajo es desarrollar una herramienta de visualización que sirva para visualizar datos estructurados sobre noticias, de manera tal de poder obtener información de los datos a primera vista y poder navegar entre ellos. De esta forma, se propone una manera de visualizar estos datos basada en investigación que entrega cómo mostrar de la mejor forma estos datos, también se propone un modelo de datos que contiene los datos necesarios para la visualización, como ubicación geográfica, fecha de publicación, categoría entre otros. Este modelo se construye con el objetivo de que sirva para cualquier fuente de datos. Otro objetivo deseable de este trabajo es que la herramienta de visualización sirva para saber que ocurre en un momento determinado en el mundo y obtener detalles de estos eventos.

Para lo anterior, se presenta el diseño y la implementación de una aplicación capaz de recolectar y visualizar noticias. Se construye un prototipo realizado utilizando el lenguaje de programación *python* y el *framework* para desarrollar aplicaciones web con python *Django*.

La validación de la visualización como solución a los problemas de inspeccionar noticias y saber qué ocurre en un intervalo de tiempo se evaluará utilizando diferentes conjuntos de datos, encontrando información relevante solamente observando la visualización. Un ejemplo de lo anterior es que mediante la visualización se observó que hubo problemas en la recolección de datos. También mediante la visualización se descubrió que noticias de ciertas categorías son más difíciles de georeferenciar que otras.

Actualmente se encuentran disponibles en Internet dos versiones de desarrollo de esta aplicación, una utilizando datos de Google News[1] y otra que utiliza datos de Feedzilla[2]. Estas aplicaciones se encuentran en desarrollo, por lo que su disponibilidad no está asegurada.

Finalmente, se presenta cómo seguir extendiendo este trabajo tanto por el lado de mejorar la calidad de los datos como por el lado de mejorar la visualización y sus interacciones.

[1] `http://eventsvis.herokuapp.com/visualize/`
[2] `http://eventsvis-feedzilla.herokuapp.com/visualize/`

Capítulo 1

Antecedentes

En este capítulo se revisarán conceptos necesarios para entender el desarrollo del trabajo, tales como conceptos de visualización de datos y minería de datos. Además, se revisará el trabajo relacionado que aportó información relevante al área de estudio. Entre estos trabajos se encuentran aplicaciones que identifican eventos y los visualizan como TwitInfo y Trendistic. También se encuentra una herramienta de visualización de datos llamada Trendalyzer y una de análisis de eventos llamada Twitris. Finalmente se presenta Newsmap, aplicación que más se acerca al área de estudio dado que también visualiza noticias.

1.1. Conceptos Involucrados

1.1.1. Minería de Datos

En este trabajo se trata de visualizar *atributos* de *clusters* extraídos de una *API Web* que conforman un *dataset*. Para comprender lo anterior pueden ser necesarias algunas definiciones.

Clustering es el proceso en el que se asignan objetos a grupos según la similitud que tengan estos objetos. Los grupos formados se llaman *clusters*.

Un *dataset* es un conjunto de datos usualmente presentado como una tabla donde cada columna es un *atributo* (o variable) y cada fila es un miembro del conjunto.

Los atributos pueden ser de distintos tipos: nominal, ordinal y cuantitativo entre otros. Un atributo nominal es como una etiqueta o pertenecer a una clase, por ejemplo, color de ojos o en el caso de las noticias su categoría. Los atributos ordinales son los que tienen una relación de orden, como por ejemplo la altura: alto, mediano o bajo. Finalmente los atributos cuantitativos son los que poseen un valor numérico preciso.

Una *API* (Application Programming Interface) en el contexto de la Web es un conjunto de requests HTTP que hacen posible la comunicación entre dos sistemas. Por ejemplo, utilizando la API de Google News, es posible a través de un request GET con ciertos parámetros obtener

las últimas noticias que están ocurriendo en Estados Unidos.

1.1.2. Visualización de Datos

Construir una visualización podría ser un proceso muy subjetivo, sin embargo, existen trabajos anteriores que le dan un sustento teórico a una propuesta de visualización. Jock Mackinlay en [9] propone una aplicación capaz de automatizar la presentación de información relacional. Para esto utiliza el concepto de efectividad (*effectiveness*). La efectividad de un lenguaje gráfico se refiere a su capacidad de aprovechar el sistema visual humano. A su vez, un lenguaje gráfico se refiere a los elementos utilizados en una visualización (como ubicación, formas, colores, texturas). De esta forma, dos visualización podrían ser comparadas según su efectividad para expresar los datos.

Mackinlay elaboró un ranking con los elementos gráficos más efectivos para representar ciertos tipos de atributos, apreciado en la Tabla 1.1. Este ranking es una extensión al ranking elaborado en [7] (Figura 1.1), que solamente sirve para datos cuantitativos. Ambos rankings se basan en resultados psicofísicos, donde una definición de psicofísica es: "*Disciplina que estudia las relaciones entre la magnitud de los estímulos físicos y la intensidad de las sensaciones que producen.*".

Quantitative	Ordinal	Nominal
Position	Position	Position
Length	Density	Hue
Angle	Saturation	Texture
Slope	Hue	Connection
Area	Texture	Containment
Volume	Connection	Density
Density	Containment	Saturation
Saturation	Length	Shape
Hue	Angle	Length
Texture	Slope	Angle
Connection	Area	Slope
Containment	Volume	Area
Shape	Shape	Volume

Tabla 1.1: Ranking de Mackinlay

De esta forma, según el ranking de Mackinlay, siempre el mejor elemento para representar cualquier atributo es la posición. Luego el mejor elemento que describe un atributo cuantitativo es el tamaño (o largo). Para un atributo nominal el color es el elemento que representa mejor, y así.

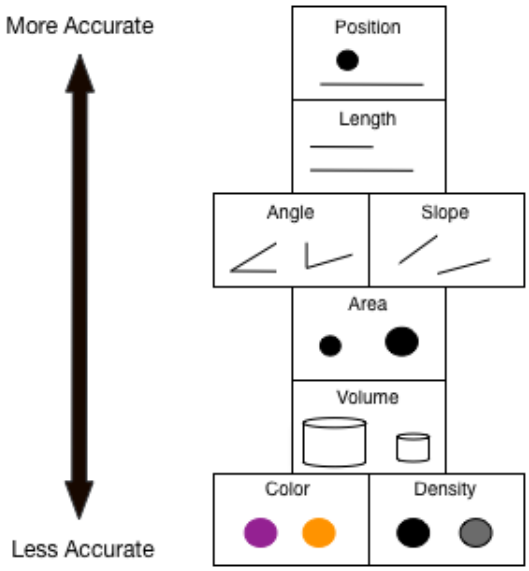

Figura 1.1: Ranking de Cleveland y McHill

1.2. Trabajo Relacionado

1.2.1. Twitris

En [5], se presenta Twitris[1] (Figura 1.2), la cual es una aplicación desarrollada para analizar eventos y entregar distintas métricas a través de redes sociales, wikipedia, noticias o otros recursos disponibles en Internet.

Twitris entrega varias métricas, entre ellas, Trending Topics de Twitter más utilizados en algunos lugares. También es capaz de mostrar un análisis de sentimiento donde se caracterizan los tweets de manera positiva o negativa. Además es capaz de mostrar un grafo de interacción de los usuarios más relevantes. En el caso de las Elecciones de Estados Unidos, se muestra el grafo de interacciones de Twitter de Obama y Romney. Por último, la aplicación es capaz de recolectar información multimedia desde distintos medios asociada a los eventos.

Twitris está orientado a analizar eventos específicos, en particular, 2012 U.S. Presidential Election, Hurricane Sandy, India Against Corruption, Occupy Wall Stret Protest. Dado esto, no es posible visualizar otros eventos y tampoco es posible visualizar varios eventos a la vez.

1.2.2. TwitInfo

TwitInfo[2] [4], fue desarrollada por investigadores del MIT con el fin de detectar, visualizar y explorar eventos. Estos eventos son detectados mediante la actividad en Twitter y son

[1] `http://twitris.knoesis.org/`
[2] `http://twitinfo.csail.mit.edu/`

Figura 1.2: Twitris: Keywords de las elecciones en Estados Unidos

visualizados en una linea de tiempo (Figura 1.3), donde el eje x representa el tiempo y el eje y el volumen de tweets.

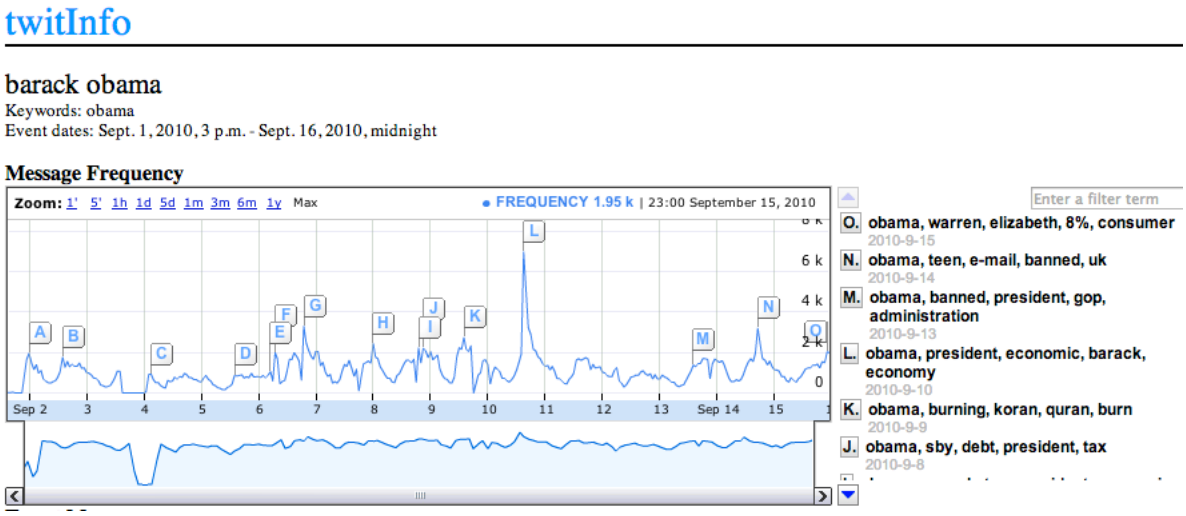

Figura 1.3: TwitInfo: Eventos identificados con el keyword “obama”

Esta aplicación es capaz de identificar eventos y asociarlos a un conjunto de keywords a partir de una consulta. También es capaz de georefenciar los tweets asociados a un evento identificado. Además se puede visualizar una consulta de keywords en un intervalo de tiempo de 1 minuto, 5 minutos, 1 hora, 1 día, 5 días, 1 mes, 3 meses, 6 meses o 1 año. Finalmente es capaz de graficar el sentimiento promedio de los tweets asociados a un evento, es decir, grafica si el evento es positivo o negativo.

Algunos de los problemas que tiene esta aplicación es que los keywords que se pueden ingresar para ser analizados están fijos y desactualizados. Además la información no cambia en tiempo real, es decir, se debe refrescar la página para poder ver los cambios en los eventos.

1.2.3. Newsmap

Newsmap[3] [11], como su creador lo define, es una aplicación que refleja el cambio constante en las noticias de Google News. Esta aplicación utiliza un algoritmo de visualización llamado Treemap [6], el cual tiene como objetivos: utilizar eficientemente el espacio de una pantalla, asegurar interacción con los contenidos, asegurar comprensión de la información de manera de facilitar la extracción de esta y finalmente entregar un diseño estético para la presentación de los datos. En la Figura 1.4 se muestra Newsmap y su visualización utilizando Treemap.

Figura 1.4: Newsmap y su visualización Treemap

En Newsmap se pueden filtrar las noticias por categorías. Estas categorías son: Mundo, Nacional, Negocios, Tecnología, Deportes, Entretenimiento y Salud. Se puede visualizar cualquier combinacion de estas categorías.

Google News posee cerca de 70 ediciones diferentes, las cuales están dadas por regiones geográficas. Cada edición muestra noticias diferentes y que no necesariamente hablan de su región. Por ejemplo, es posible ver la edición chilena de Google News, la cual tiene mayoritariamente noticias chilenas. Sin embargo, en esta edición también pueden aparecer noticias de Estados Unidos que son publicadas en Chile. Newsmap tiene la capacidad de filtrar algunas de estas ediciones. Las ediciones que posee Newsmap son: Argentina, Australia, Austria, Brasil, Canada, Francia, Alemania, India, Italia, México, Holanda, Nueva Zelanda, España, Reino Unido, Estados Unidos.

Como se puede apreciar en la Figura 1.4, Newsmap muestra las noticias más relevantes con un tamaño mayor. Es decir, se está mostrando un atributo cuantitativo con su segundo

[3] `http://newsmap.jp/`

mejor descriptor gráfico que es el tamaño (o largo) . También las categorías se muestran con colores (atributo nominal usando color). Finalmente se ocupa densidad para mostrar noticias que ocurrieron hace menos de 10 minutos, más de 10 minutos, más de una hora. En este caso el atributo es ordinal y nuevamente se ocupa la segunda mejor opción según Mackinlay, que es la densidad. Probablemente ninguno ocupa la posición debido a las características del algoritmo de Treemap, el cual ocupa la posición solo para distribuir los elementos.

Una de las desventajas de esta visualización es que no es posible visualizar un orden de tiempo entre las noticias. Y mucho menos, no es posible navegar entre las noticias para, por ejemplo, encontrar noticias más antiguas. Otra de las críticas a Newsmap, es que no se refresca automáticamente para mostrar las ultimas noticias que están ocurriendo, es decir, para refrescar el contenido de la página se debe entrar nuevamente al sitio.

1.2.4. Trendistic

Trendistic (Figura 1.5) fue desarrollada por una empresa llamada Indextank, que fue adquirida hace poco por LinkedIn. El objetivo de esta aplicación era identificar eventos por medio de la actividad social de Twitter. Es posible ingresar un keyword en la aplicación y revisar la actividad de este keyword en un rango de 24 horas, 7 días, 30 días o 90 días. Además, al ingresar un keyword se muestran los tweets relacionados. En particular, en el gráfico de la Figura 1.5 se aprecian los eventos relacionados el keyword “earthquake”. Lamentablemente, Trendistic ya no se encuentra disponible en Internet.

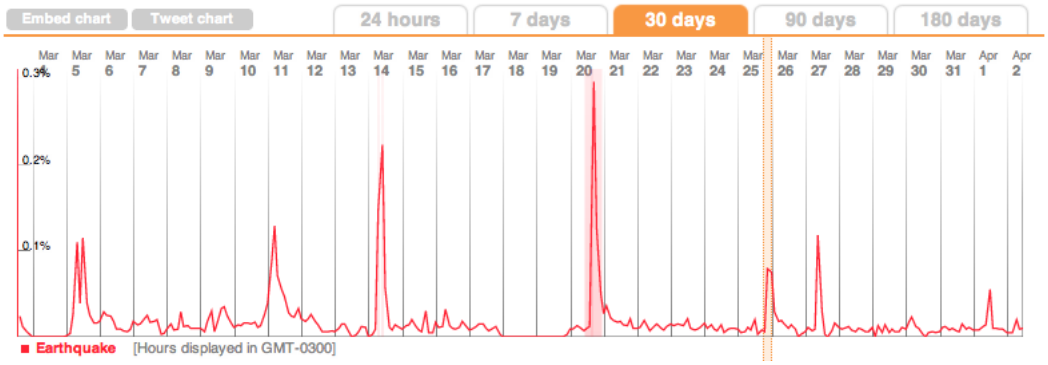

Figura 1.5: Trendistic: Eventos identificados con el keyword “earthquake”

Si bien los eventos en Trendistic se pueden apreciar de manera clara, no es posible obtener una georeferencia de estos eventos. Además, no es posible comparar distintos eventos, es decir, ingresar varios keywords. Por último, los intervalos para navegar en el tiempo están fijos, por lo que tampoco es posible ir a una fecha específica a revisar el comportamiento de un keyword.

1.2.5. Trendalyzer

Hans Rosling[4], creó una herramienta de visualización llamada Trendalyzer, la cual usó para mostrar datos de la población mundial y su cambio en el tiempo. En particular, en la charla TED citada en el pie de página, muestra como la esperanza de vida de algunos países varía junto a la cantidad de hijos per capita en el tiempo en un gráfico parecido al de la Figura 1.6.

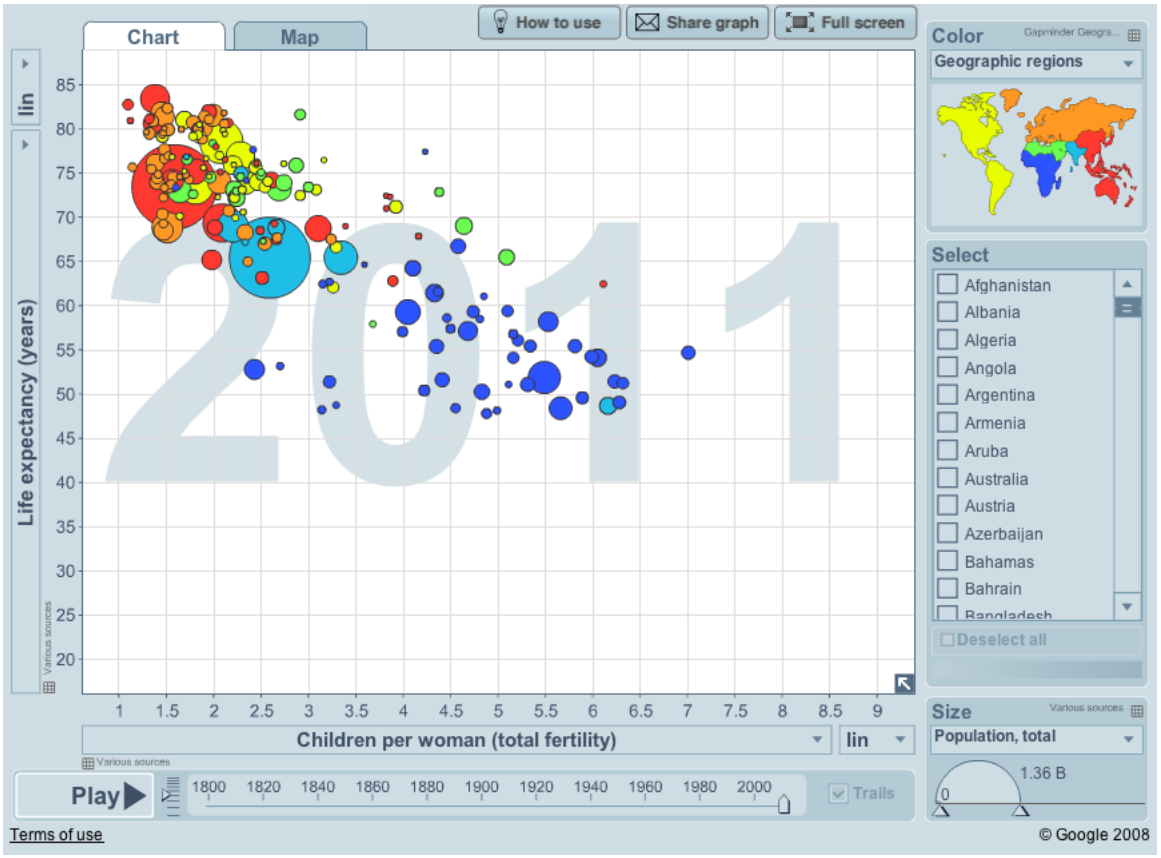

Figura 1.6: Trendalyzer: Esperanza de vida vs. Fertilidad Mujeres

Se puede notar que la visualización propuesta por Hans Rosling posee un lenguaje gráfico acorde al ranking de Mackinlay. Por ejemplo, la fertilidad y la esperanza de vida que son elementos cuantitativos se reflejan utilizando posición. Por otra parte, un atributo nominal como el continente se ve reflejado por el color. Finalmente el tamaño de los círculos se utiliza para un atributo cuantitativo como lo es la población del país.

Lo bueno de esta visualización es que es capaz de mostrar cerca de 5 atributos sin que la visualización se vuelva inentendible.

[4]http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen.html

Capítulo 2

Especificación del Problema

El problema que se quiere abordar es el de visualizar eventos noticiosos con dos fines: saber qué ocurre en el mundo en un intervalo de tiempo específico y como herramienta de investigación para inspeccionar datos estructurados como noticias, encontrando información relevante a primera vista.

Para lograr lo anterior, se separa el problema en dos partes: Recolección y Visualización de Datos. Si bien este trabajo está más orientado a la visualización, fue necesario recolectar datos para poder validar la solución.

2.1. Recolección de Datos

La construcción de una visualización de eventos noticiosos con los fines descritos anteriormente posee distintos desafíos. En primer lugar es necesario armar un dataset con los datos disponibles en Internet. Para esto existen distintas APIs que proveen datos de noticias. Entre las APIs más conocidas se encuentran las de Google News, Feedzilla y News Search de Yahoo!.

En particular, Google News y Feedzilla son capaces de entregar las noticias que han ocurrido en los últimos minutos en ciertas categorías. Esto quiere decir que sus respectivas APIs proveen la funcionalidad de rescatar las ultimas noticias ocurridas a partir del momento en que se realiza la consulta sin realizar una consulta de "keywords". Por otro lado, las noticias de Yahoo! por su parte solo entregan noticias relacionadas con una consulta, es decir, entregan un conjunto de noticias dada una búsqueda "keywords" y no es posible obtener las ultimas noticias sin tener que realizar una consulta de palabras.

Habiendo obtenido un conjunto de noticias, el siguiente desafío consiste en limpiar los datos, es decir, descartar los atributos que no son necesarios ni interesantes para la visualización. Además, puede pasar que los datos no proporcionen toda la información necesaria para que estos puedan ser visualizados. De hecho, si se quisiera utilizar los datos de Google News sería necesario calcular de alguna forma la ubicación geográfica de las noticias, puesto

que este atributo no viene dado por la API. Otro escenario podría ser el de hacer clustering de noticias en el caso de que la API no retorne un conjunto de noticias relacionadas.

También se quiere automatizar el proceso de recolección y limpieza, para así poder recolectar datos diariamente o cada cierto periodo. Esto involucra la programación de un cron recolector de noticias que cada cierto tiempo pueda obtener, limpiar y almacenar noticias.

2.2. Visualización de Datos

Teniendo ya datos más refinados y estructurados, se vuelve al problema de visualización de eventos noticiosos. Existen varios desafíos en la construcción de una visualización.

Uno de estos desafíos es utilizar apropiadamente el espacio limitado para un conjunto grande de datos: este problema tiene relación con las limitaciones físicas de una pantalla cualquiera y con la cantidad de información cómoda de visualizar para una persona. Aún utilizando un pixel para visualizar un elemento, en una pantalla de 1280x800 pixeles solo se podrían visualizar 1,024,000 datos, que si bien no es poco, es una cota superior. Además, una persona no sería capaz de distinguir tal información. Es necesario encontrar una forma de agrupar la información y mostrar una cantidad cómoda para la visualización sin dejar fuera la información relevante.

También, mostrar la mayor cantidad de atributos posibles: en una visualización puede ser posible codificar muchos atributos, el problema es que la visualización se vuelve inentendible. Lo anterior se llama sobrecodificación y concretamente es que si bien se pueden utilizar todos los recursos gráficos mostrados en la Tabla 1.1 como color, largo, posición, textura, ángulo, saturación, etc. para describir múltiples atributos, abusar de esto lleva a que la visualización se sobrecargue y no pueda ser comprendida. Es necesario acotar el número de atributos que se quieren mostrar, además de representarlos de la mejor manera gráfica según el tipo de sus atributos (ordinal, cuantitativo y nominal).

Visualizar tiempo y ubicación al mismo tiempo: se debe encontrar una manera de visualizar ambos atributos al mismo tiempo. Es muy común cuando se visualizan datos georeferenciados utilizar un mapa para ubicar los elementos, como lo hace la funcionalidad para graficar los tweets en tiempo real de Twitris [5]. Cuando se hace lo anterior, el tiempo se visualiza como una animación en vez de un elemento gráfico explícito, perdiendo la posibilidad de navegar en él.

Utilizar interacciones efectivas en la visualización: uno de los problemas grandes es visualización científica es el de las interacciones [8]. Aún se investiga como hacer visualizaciones efectivas e intuitivas. En particular para la visualización que se quiere construir, sería encontrar una forma para navegar en los datos como por ejemplo filtros y vistas que muestren más atributos. También este tema se relaciona con la codificación que deben tener los atributos para que estos puedan ser comprendidos de mejor manera.

Si bien ya existen algunas herramientas para visualizar eventos como las expuestas en la Sección 1.2, se quiere construir una visualización en base a los descriptores visuales de

Mackinlay para asegurar efectividad. También se quiere aprovechar las falencias de las aplicaciones revisadas en la Sección 1.2, como por ejemplo no poder navegar en el tiempo, sets fijos de datos, representación de múltiples atributos a la vez como tiempo y ubicación.

Capítulo 3

Descripción de la Solución

La solución representada por la visualización de eventos está compuesta por varias partes. Primero se presenta la arquitectura del sistema, donde se muestra el diseño de software de la aplicación más las interacciones con sistemas externos. Después se presenta el modelo de datos, el cual fue diseñado como la cantidad mínima de atributos para que la aplicación visualice los datos. La idea es adaptar los datos de distintas fuentes a este modelo de datos. Luego se detalla cómo se realiza la recolección de los datos y la interacción con sistemas externos como las APIs de noticias, georeferencia de texto y extracción de keywords. Finalmente se muestra la construcción de la visualización más los detalles de implementación de toda la aplicación.

3.1. Arquitectura del Sistema

La arquitectura de la solución implementada se muestra en la Figura 3.1. En ella se puede apreciar tanto los servicios externos como la arquitectura interna de Eventsvis.

La aplicación está compuesta por varios servicios externos, entre los que se encuentran la API de Google News, la API de Feedzilla, Yahoo! Placemaker y Content Analysis de Yahoo!. Como ya se ha mencionado, Google News y Feedzilla proveen los datos de los dos prototipos construídos. Yahoo! Placemaker es el servicio utilizado para georeferenciar texto y Content Analysis, que también es de Yahoo! sirve para extraer "keywords" o palabras clave de un texto.

También se muestran los módulos de Eventsvis, donde cada uno tiene una tarea en particular: Collector es el encargado de pedir datos y almacenarlos en la base de datos. Visualization se encarga de las consultas realizadas por un cliente como un Browser consultando a la base de datos según los criterios del cliente y mostrando la visualización. También se programaron distintos "Wrappers" que son capaces de consultar las distintas APIs que se utilizarán.

El flujo de información es el siguiente:

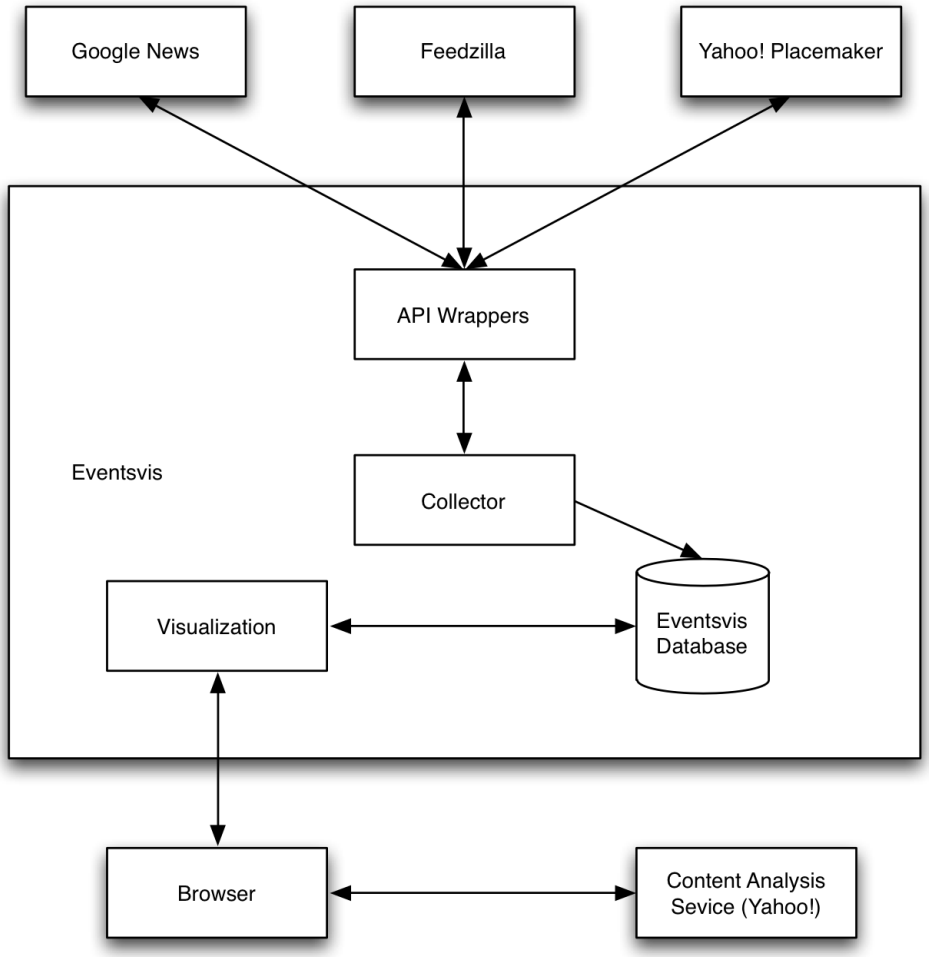

Figura 3.1: Arquitectura de la aplicación de visualización Eventsvis

1. **Collector** consulta a alguna API que tenga información de noticias (**Feedzilla o Google News**), todo esto a través del **wrapper** correspondiente a la API.
2. **Collector** consulta a **Yahoo! Placemaker** de qué ubicación son estas noticias.
3. **Collector** guarda las noticias según la estructura definida en el modelo de datos de la siguiente sección.
4. Un cliente a través de un **Browser** puede consultar los datos en la aplicación.
5. Mientras se muestran los datos en la aplicación, se consultan los keywords de una noticia usando el servicio de **Content Analysis**. Todo esto en el cliente.

El servidor utilizado para mantener Eventsvis tanto en su versión Google News como en su versión Feedzilla es Heroku[1]. En un principio, la aplicación estaba alojada en los servidores del DCC, pero la necesidad de incluir nuevas librerías cada cierto tiempo hacía incomoda la mantención y el desarrollo de la aplicación en los servidores del DCC.

El lenguaje de programación elegido para desarrollar esta aplicación es Python utilizando el framework para desarrollar aplicaciones web Django. Ambas herramientas fueron elegidas

[1] `http://www.heroku.com/`

debido a la experiencia que se tenía con ellas. La base de datos en la que se almacenan las noticias utiliza el motor Postgres.

Para el desarrollo de la vista de la visualización se utilizó HTML y Javascript. Una de las librerías más recientes utilizadas para visualizar datos en la web es D3 [10], la cual se encarga tanto del manejo de datos en formato JSON que provee la aplicación como de las interacciones y animaciones.

Finalmente, la aplicación completa se está versionando en Github[2] y queda disponible para que pueda seguir siendo extendida o para que pueda ser probada con otros conjuntos de datos.

3.2. Modelo de Datos

El modelo de datos diseñado tiene el fin de soportar múltiples fuentes de datos. La idea es que cualquier conjunto de datos que cumpla con la especificación del modelo de base de datos sea visualizable en la aplicación, por lo que esta especificación es acorde a lo que ofrecen comúnmente las APIs de noticias. De esta forma, es posible visualizar datos provenientes de Google News y Feedzilla. Más aún, si alguien procesara datos no estructurados como los de Twitter y los estructura de manera tal de que cumpla con la especificación, estos datos podrían ser visualizados en la aplicación.

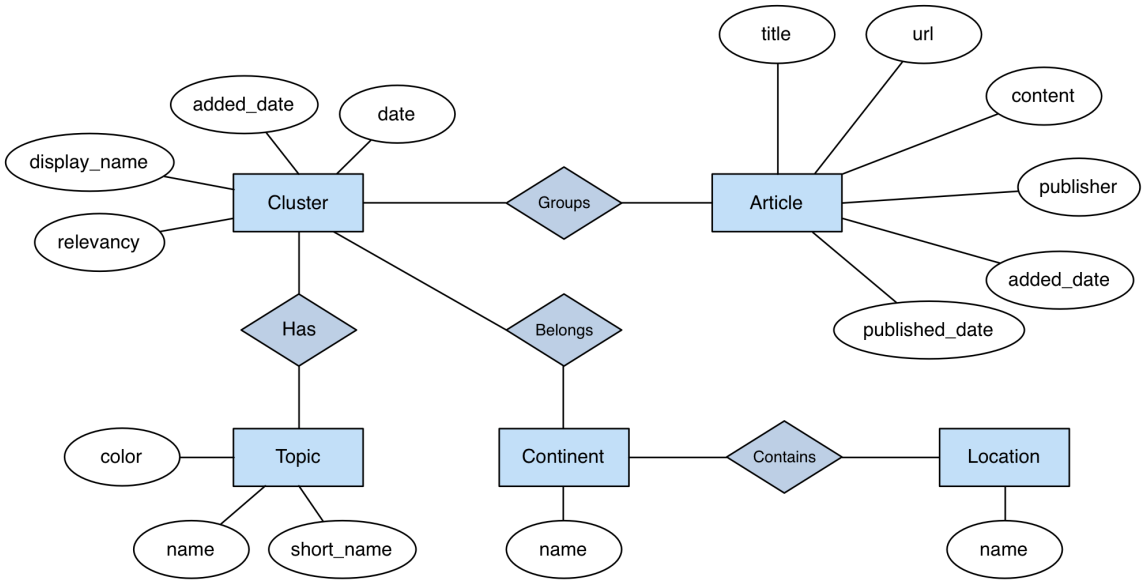

Figura 3.2: Diagrama Entidad-Relación de Eventsvis

La Figura 3.2 muestra el modelo de datos diseñado para la aplicación Eventsvis. Se pueden apreciar 5 entidades: Cluster, Article, Topic, Continent, Location. A continuación se detalla cada una de estas entidades:

[2] `http://github.com/campalma/memoria`

- **Cluster**: Conjunto de noticias. Estos son los elementos que serán visualizados en la aplicación.
- **Article**: Una noticia, que contiene título, url, contenido, editor, fecha de publicación y fecha en que se agrega a la base de datos.
- **Topic**: Categoría de la que trata una noticia. Algunos ejemplos de categorías son: Deportes, Negocios, Salud. Una noticia solo puede tener una categoría.
- **Continent**: Continente del cual habla de un conjunto de noticias. Un cluster de noticias puede corresponder a más de un continente.
- **Location**: Lugar específico del que habla una noticia. Puede ser una ciudad o un país. Un cluster de noticias puede corresponder a muchos lugares.

3.3. Recolección de Datos

Para recolectar datos se utilizaron las APIs de Google News[2] y de Feedzilla[1]. Estas APIs proveen consultas de búsqueda según distintos parámetros. A continuación se revisan las características de ambas APIs, cómo se recolectan datos con cada una de estas APIs, sus ventajas y desventajas para conformar un dataset.

3.3.1. Google News

La API de Google News provee argumentos para restringir la búsqueda de noticias. Los más importantes de estos se muestran en la Tabla 3.1.

Argumento	Descripción
q	Consulta o expresión de búsqueda
ned	Edición de noticias a consultar
rsz	Cantidad de resultados a obtener
scoring	Orden de noticias, puede ser por fecha o relevancia
start	Índice para recorrer resultado de noticias
topic	Categoría de las noticias

Tabla 3.1: Argumentos URL importantes de Google News Search API

El parámetro *q* es obligatorio si es que no se especifica el parámetro *topic*. Al no especificar el parámetro *q*, se puede realizar una búsqueda que retorne las últimas noticias ocurridas en alguna categoría sin la necesidad de escribir una palabra o "keyword". El parámetro *ned* sirve para consultar en alguna de las ediciones de Google News. Estas ediciones corresponden a las regiones que soporta Google News, como por ejemplo Estados Unidos, Australia, Chile y en total son cerca de 70.

La respuesta de una consulta a Google News, Figura 3.3, incluye título de la noticia, noticias relacionadas, url de la noticia, url del cluster de noticias, imágenes, fuente, fecha de publicación entre otros atributos. Según la documentación de la API, una consulta debiese

ser capaz de retornar la ubicación de una noticia, pero en la práctica no es así, por lo que fue necesario calcular este atributo utilizando el servicio de Yahoo! Placemaker.

```
{
  "responseData":{
    "results":[
      {
        "GsearchResultClass": "..."
           "clusterUrl": "..."
           "content": "...";
           "unescapedUrl": "..."
           "url": "..."
           "title": "..."
           "titleNoFormatting": "..."
           "location": "..."
           "publisher": "..."
           "publishedDate": "..."
           "signedRedirectUrl": "..."
           "language": "..."
           "image": "..."
            "url": "..."
            "tbUrl": "..."
            "originalContextUrl": "..."
            "publisher": "..."
            "tbWidth": "..."
            "tbHeight": "..."
           },
           "relatedStories": [
              {...}
            ]
      }
      {...}
    ]
  }
}
```

Figura 3.3: Respuesta de consulta a Google News

Dado lo anterior, el algoritmo utilizado para recolectar noticias es el siguiente:

```
for r in regions:
  for t in topics:
    for p in pages:
      for a in articles:
        for rel in related:
          rel.saveArticle()
        a.saveCluster()
```

Figura 3.4: Algoritmo de recolección utilizando Google News

Una de las ventajas de utilizar Google News es la estructuración que posee la respuesta de la API. Esto quiere decir que no se necesita procesar mucho los datos para obtener los datos necesarios para la visualización, por lo que esta estructuración es fácil de adaptar a la estructura del modelo de datos.

Además, las noticias obtenidas con esta API son claramente más relevantes que las entregadas por Feedzilla. Esto se expondrá en detalle en la sección 3.3.2.

Otra ventaja, es que es posible recolectar noticias de 70 regiones, lo que permitiría conformar un dataset con cerca de 140000 noticias por día. El número anterior se estima dado que se logró recolectar 2000 noticias en un día pertenecientes a Estados Unidos.

De esta forma, con Google News fue posible recolectar 11638 artículos, correspondientes a 2577 clusters de 2273 lugares. Estas noticias fueron recolectadas durante 11 días. Esto fue realizado recolectado las noticias de todas las categorías disponibles en Google News una vez al día solamente de la edición de Estados Unidos. Esto se hizo utilizando un cron que ejecutaba el script de recolección una vez al día a la misma hora.

Se intentó recolectar noticias de diferentes ediciones para obtener más resultados. El problema es que la API empezó a retornar errores debido a que se realizaban muchos requests. Por esto no fue posible recorrer las 70 regiones provistas por Google News para recolectar noticias y solamente era posible recorrer 2 regiones antes de que la API retornara errores. Así es como se busca otra fuente de datos que pueda ser utilizada sin restricciones como la de Feedzilla.

3.3.2. Feedzilla

La API de Feedzilla es mucho más simple, pero por esta razón posee más limitaciones. En la Tabla 3.2 se exponen los argumentos que pueden ser enviados en una consulta a la API de Feedzilla.

Argumento	Descripción
topic	Categoría de las noticias
since	Fecha mínima para una noticia
count	Cantidad de resultados a obtener

Tabla 3.2: Argumentos URL importantes de Feedzilla Newsfeed API


```
{
   "articles":[
      {
         "url": "http://proxy.feedzilla.com/r/35423",
         "title": "Feedzilla: News edition has changed! Please read more!",
         "summary": "Please read the article when possible, we are likely to
             get used to it!",
         "publish_date": "Thu, 01 Jul 2010 14:14:38 +0000",
         "author": "Feedzilla Team",
         "source": "CNN"
      },
   ],
   "syndication_url": "http://feeds.feedzilla.com/en_us/news/business.rss"
}
```


Figura 3.5: Respuesta de consulta a Feedzilla

Así, la forma de recolectar noticias utilizando esta API se muestra en la Figura 3.6.

```
for t in topics:
  for a in articles:
    a.saveArticle()
    a.saveCluster()
```

Figura 3.6: Algoritmo de recolección utilizando Feedzilla

La ventaja de Feedzilla por sobre Google News es que sus términos de uso no son restrictivos. Es posible almacenar los datos obtenidos a través de la API, es posible cambiar el orden de los resultados y también es posible automatizar la obtención de información siempre que no supere cierto límite de requests.

La calidad de las noticias no es muy buena. A simple vista se nota que las noticias obtenidas no son tan relevantes como las de Google News. Esto se puede observar en las categorías en las que Feedzilla permite buscar, por ejemplo, Oddly Enough y Fun Stuff. También se observa porque las noticias son más bien locales y probablemente no de interés mundial.

Por otro lado, la API no responde con un cluster de noticias como lo hace Google News, por lo que habría que procesar los datos obtenidos para obtener un cluster lo que es inviable debido al alcance de este trabajo. De todas formas, es posible adaptar esto al modelo de datos almacenando una noticia como un cluster de tamaño uno.

3.4. Georeferencia de los Datos

Los datos obtenidos desde Google News y Feedzilla no poseían ubicación geográfica, por lo que para determinarla fue necesario utilizar otra API, llamada Yahoo! Placemaker[3]. Esta API es capaz de determinar los lugares geográficos de los que habla un texto o una URL (Tabla 3.3).

Argumento	Descripción
documentContent	Texto a georeferenciar
documentURL	URL a georeferenciar (si es que no se especifica documentContent)
documentType	Tipo del texto
outputType	Formato de salida (json, xml)

Tabla 3.3: Argumentos URL importantes de Yahoo! Placemaker

La respuesta de la API de Placemaker contiene información que depende de la confianza de la respuesta. Por ejemplo, si la API está segura de la ubicación de cierto texto, entrega atributos como país, ciudad, coordenadas, continente, etc. Sino, la respuesta es más ambigua y cambia la estructura del json, por ejemplo, podría entregar solamente continente. En otras ocasiones, la respuesta es incluso ambigua, porque entregaba el país pero no el continente.

Dado este problema, y que solamente el nombre de las ubicaciones era el requerido, se programa un parser json capaz de extraer solamente los nombres de la respuesta de Placemaker.

Luego de esto se utiliza una librería de Python llamada "countryutils", que dado un país es capaz de responder el continente de dicho país. Con esto se obtienen ubicaciones genéricas que pueden ser países, ciudades, continentes, regiones, etc., además de continentes extraídos de las ubicaciones anteriores.

Resuelto lo anterior, para obtener la ubicación de un conjunto de noticias, se tomaron varios enfoques. Dado que es posible consultar por texto a la API de Placemaker y las noticias recolectadas ya sean de Google News o de Feedzilla entregan un resumen, es posible enviar este texto para determinar la ubicación.

También las APIs de noticias entregan la URL de estas, por lo que en teoría es posible encontrar la ubicación mediante esta vía. Con lo anterior ocurren dos problemas: Feedzilla no entrega directamente la URL de la noticia, entrega una URL que redirige a la URL original haciendo que Placemaker no encuentre lugares precisos. El otro problema es que la URL entrega peores resultados que vía texto, esto probablemente porque intenta ubicar lugares sobre todo el texto de la página, que muchas veces puede tener otras noticias no relacionadas con la original o publicidad.

Finalmente se optó por enviar el resumen de la noticia a Placemaker, pudiendo georeferenciar el continente de un cluster más del 65 % de los casos. Esta medición fue realizada georeferenciando los datos recolectados utilizando Google News descritos en la sección 3.3.1

3.5. Construcción de la Visualización

Utilizando los conceptos de visualización de datos entregados por Mackinlay, inicialmente se propuso la visualización de eventos noticiosos de la Figura 3.7. En esta visualización, el eje X era utilizado para graficar el tiempo, el eje Y era utilizado solamente para distribuir la información. El color representaba el continente del cual hablaba la noticia y el borde representaba si la noticia solamente era discutida dentro del país. El tamaño de los círculos representa la relevancia de la noticia.

El tiempo es un atributo cuantitativo y está siento representado por su mejor descriptor visual el cual es la posición. El continente visto solamente como un nombre es un atributo nominal. Si el continente hubiese sido descrito con coordenadas sería un atributo cuantitativo. Dado que no es el caso y se trata de un atributo nominal, el mejor descriptor es el color.

La localidad de una noticia, es decir, si se trata de una noticia de la cual se habla solo en el país del cual habla o en todo el mundo, es un atributo nominal. Es un atributo nominal pues solo se trata de encasillar en las etiquetas "local" o "global". Dado esto, se grafica utilizando el tercer descriptor gráfico que representa mejor un atributo nominal que es la textura, o en este caso un borde.

Adicionalmente, esta visualización debía ser capaz de expandir cada círculo para obtener mayor información sobre los eventos. También, se debía poder navegar entre las noticias de un mismo lugar. En la Figura 3.7 se muestra como se permite navegar en más noticias de Argentina.

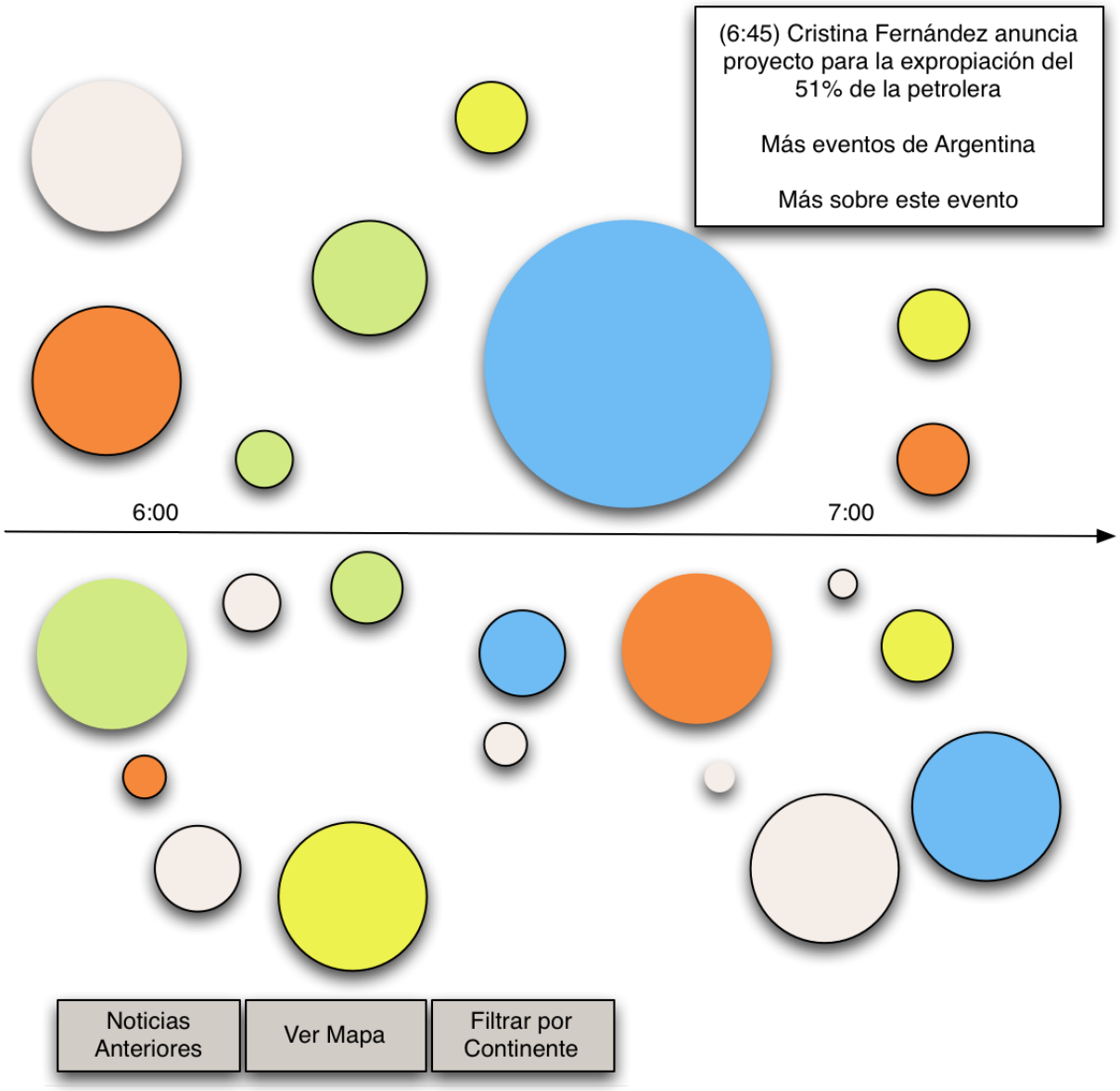

Figura 3.7: Propuesta Inicial de Visualización

Además se pensó en tener dos vistas: una como la de la Figura 3.7 y otra con un mapa como el de Twitris (Figura 1.2) donde se perdía la visualización del tiempo pero se ganaba una visualización más clara de la ubicación geográfica. Finalmente se quería entregar la posibilidad de navegar por el tiempo y de filtrar por continente.

El problema de la propuesta anterior es que dejaba fuera el atributo de la categoría. Además no se estaba aprovechando el eje Y para graficar algún atributo de los recolectados y según Mackinlay, la posición es la mejor opción para graficar cualquier atributo. Por otra parte, no se conocía el valor de visualizar si una noticia solo ocurre a nivel local o a nivel global.

3.6. Implementación

3.6.1. Prototipo Inicial

Dada la especificación del problema y el diseño de la solución, se construyó el primer prototipo de la aplicación Eventsvis apreciado en la Figura 3.8.

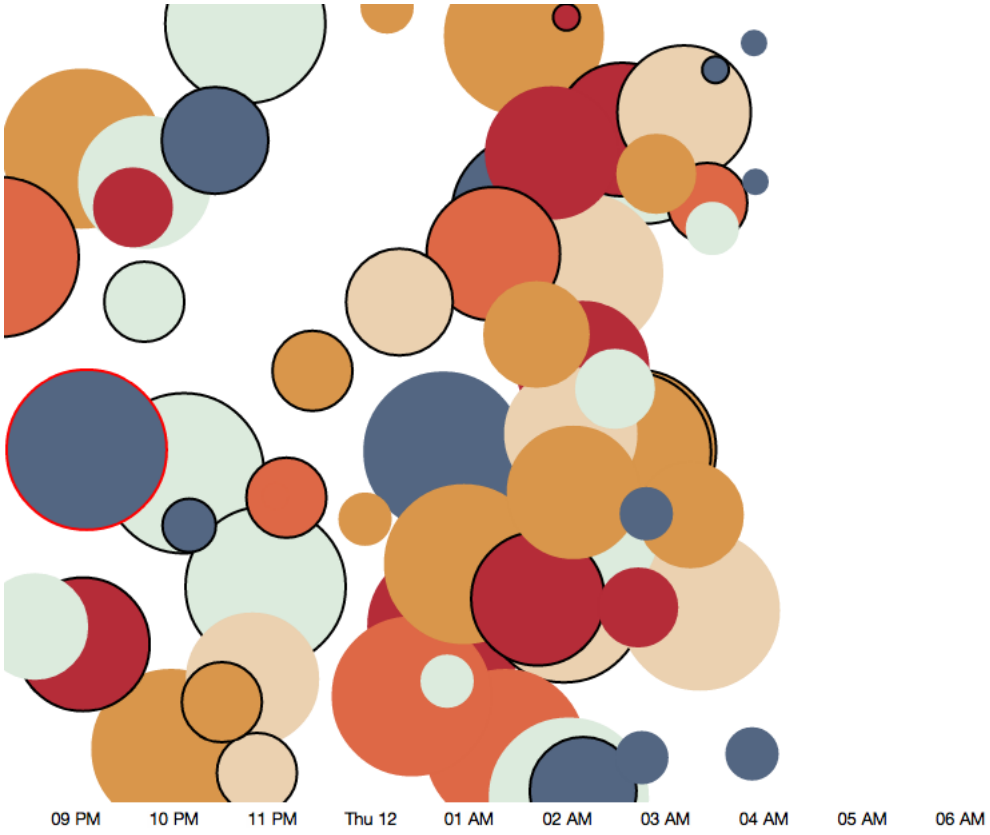

Figura 3.8: Primer prototipo construido de Eventsvis: Visualización

2012-07-12T06:40:36 Dozens Killed in Nigeria Fuel Truck Fire [Voice of America]
2012-07-12T06:25:41 Nigeria fuel tanker fire kills 95 [Telegraph.co.uk]
2012-07-12T06:25:19 Nigeria Fuel Tanker Fire Leaves at Least 95 People Dead [Businessweek]
2012-07-12T06:07:18 Official: 95 killed in Nigeria fuel tanker fire [Fox News]
2012-07-12T05:47:11 Over 100 people dead after Nigerian tanker fire [ABC Online]
2012-07-12T05:03:07 Nigerians die in fuel tanker fire [BBC News]
2012-07-12T04:25:00 Nigeria oil tanker fire kills more than 100 [AFP]

Figura 3.9: Primer prototipo construido de Eventsvis: Contenido de cluster seleccionado

A la fecha de construcción de este prototipo no se tenían todos los datos necesarios para la visualización, por lo que en un comienzo se asumió que se tenían todos los datos. En particular en este prototipo inicial se habían recolectado datos de Google News cuya ubicación

era desconocida y no se habían georeferenciado, por lo que los colores que se pueden apreciar en la Figura 3.8 son aleatorios.

Por otro lado, se está graficando en el borde de los círculos si una noticia ocurre a nivel local o global. Este dato tampoco se había calculado en estos momentos e incluso fue desechado para las siguientes versiones de la aplicación dado que los beneficios de visualizarlo eran pocos considerando su complejidad para calcularlo.

Al hacer click en un círculo, se muestra el contenido del cluster de noticias (Figura 3.9). El contenido de un cluster son sus noticias relacionadas con su fecha de publicación y fuente. Al hacer click en una de estas noticias se abre una nueva pestaña en el navegador con la noticia original.

3.6.2. Eventsvis Google News

Iterando sobre el prototipo inicial, se obtiene como resultado Eventsvis Google News. Esta aplicación se encuentra disponible en Internet en los servidores de Heroku[3] con un conjunto de datos correspondientes a 11 días.

Google News, como se detalló en la sección 3.3.1 entrega varios atributos. En la figura 3.10 se aprecian los clusters de noticias, donde cada círculo es un conjunto de noticias relacionadas. El tamaño de estos círculos está dado por la cantidad de noticias relacionadas y por la ubicación en el resultado de la consulta a Google News.

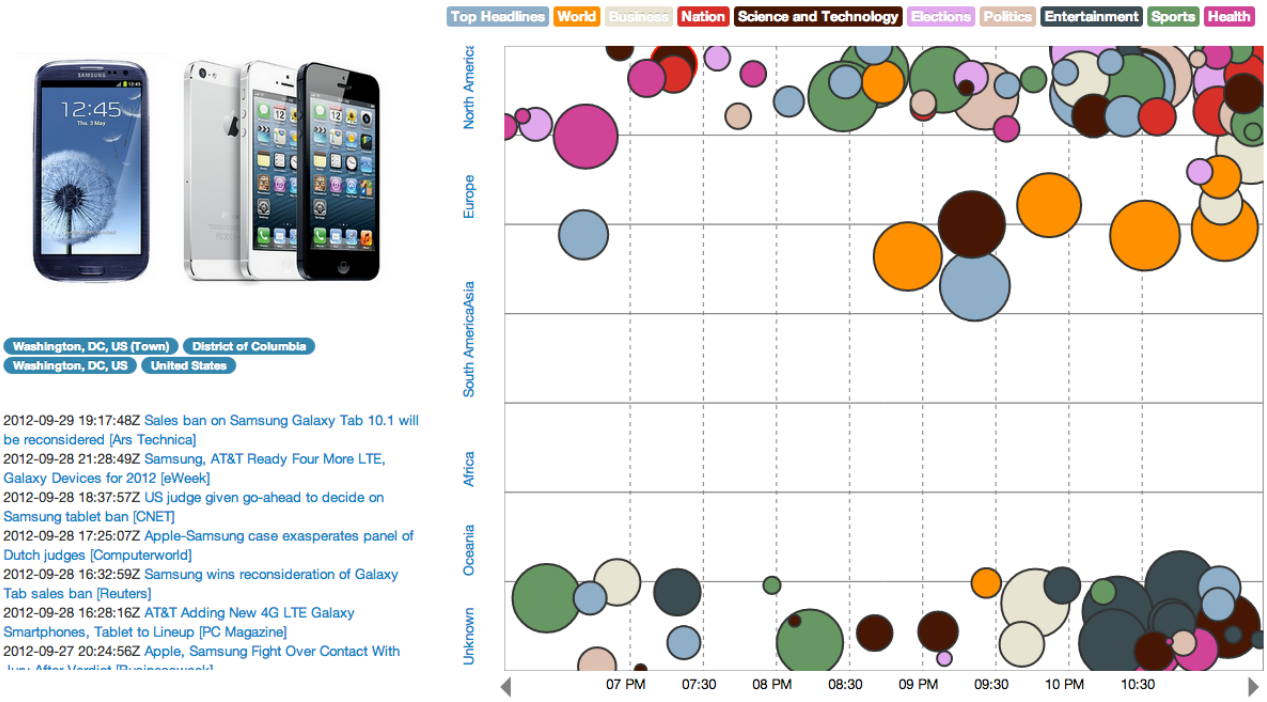

Figura 3.10: Estado actual del prototipo Eventsvis utilizando datos de Google News

[3] `http://eventsvis.herokuapp.com/visualize`

Para esta versión se utilizó el eje y para graficar la ubicación de las noticias. Anteriormente, las categorías no se estaban utilizando y ahora pasan a ser el color de los clusters y también pasan a mostrarse en la parte superior de la visualización donde toman dos papeles: servir como simbología para la visualización y también como filtros en las noticias contenidas en la visualización.

A la izquierda de la Figura 3.10 se muestra: una foto del cluster seleccionado en la visualización, los lugares detectados utilizando Yahoo! Placemaker y links a todas las noticias que componen en cluster, que además contienen la fecha de publicación y la fuente.

En esta versión de la aplicación también es posible filtrar por continentes haciendo click en un continente. Si se hace click en Europe por ejemplo se mostrarán todas las noticias referentes a Europa manteniéndose el filtro de las categorías.

También se puede navegar en el tiempo haciendo click en las flechas que se encuentran en el eje del tiempo. Al iniciarse la visualización se muestran las 100 noticias más recientes que se encuentran en la base de datos. Al hacer click en una flecha, por ejemplo en la izquierda, lo que se hace es mostrar las 100 noticias que anteceden a las que se estaban visualizando en tiempo. De esta forma, la escala de tiempo es automática y depende del intervalo de tiempo al que corresponden estas 100 noticias.

De la implementación de Eventsvis Google News surge la interrogante de si la base de datos es lo suficientemente rápida para manejar consultas con una gran cantidad de noticias. Al menos con las noticias correspondientes a 11 días de recolección si es posible filtrar por categorías, tiempo y ubicación.

3.6.3. Eventsvis Feedzilla

Dada la problemática de utilizar los datos de Google News, que se detalla en la Sección 3.3.1, se construye una nueva rama de la aplicación que utiliza datos de Feedzilla también disponible en Heroku[4].

Esta aplicación también se crea con el objetivo de probar que el modelo de datos sirve para almacenar noticias de distintas fuentes y que estas pueden ser visualizadas de igual forma por la aplicación Eventsvis. El resultado se muestra en la Figura 3.11.

Esta visualización es igual a la mostrada en la Sección 3.6.2, con la diferencia de que se ocupa toda la pantalla visualizando las noticias ocultando las noticias relacionadas y las ubicaciones asociadas. Al hacer click en un círculo se oscurece la pantalla y sobresale el artículo que se quiere visualizar, tal como se muestra en la Figura 3.12. En este recuadro se muestra el título de la noticia, el contenido, los lugares de los que habla y el conjunto de "keywords" identificados por Yahoo!

Si bien hubo que hacer cambios en la aplicación y en el modelo de base de datos para poder visualizar las noticias de Feedzilla, el esfuerzo realizado fue mínimo. En menos de una

[4] `http://eventsvis-feedzilla.herokuapp.com/visualize`

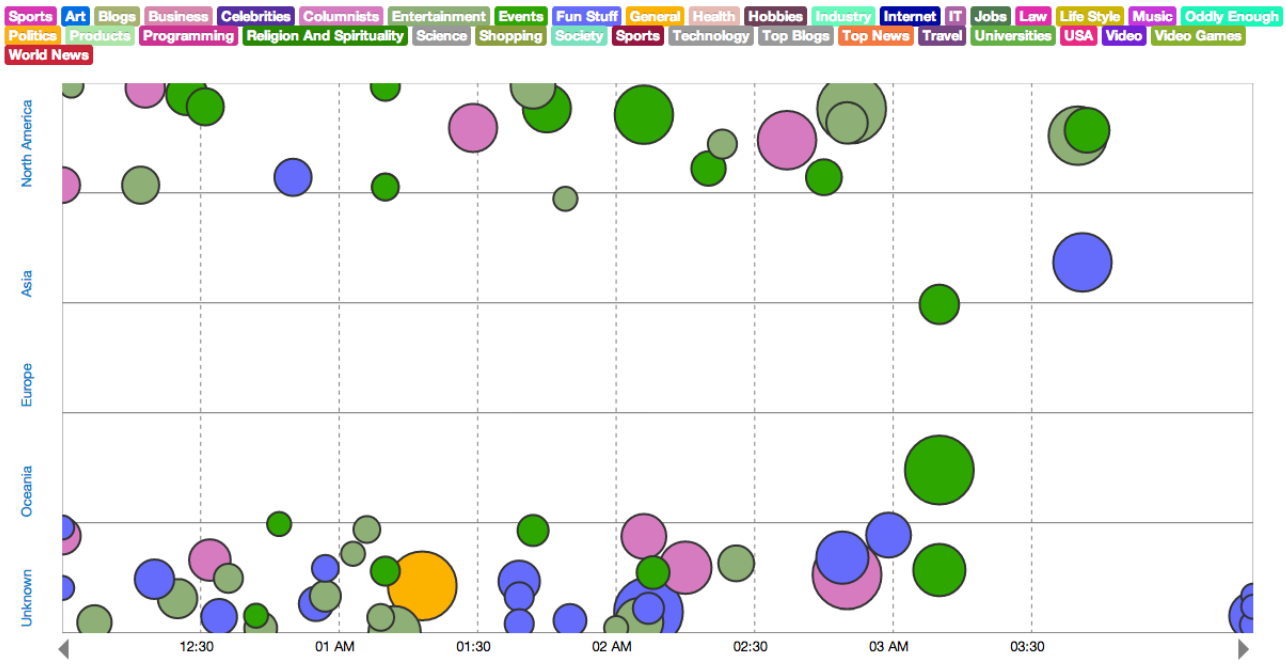

Figura 3.11: Estado actual del prototipo Eventsvis utilizando datos de Feedzilla

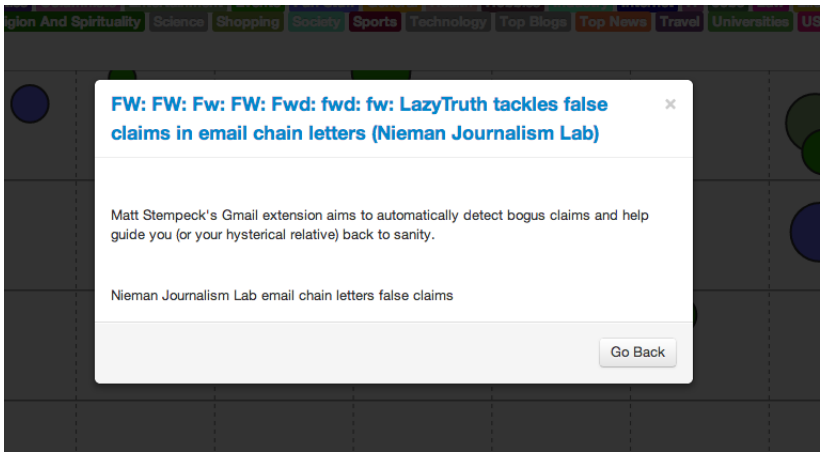

Figura 3.12: Lightbox al hacer click en un círculo

semana de trabajo fue posible cambiar la aplicación que visualizaba noticias de Google News a otra que visualizaba las noticias de Feedzilla. De esta forma se prueba que la aplicación y el modelo de datos son los suficientemente flexibles para aceptar otra fuente de datos sin complicaciones.

3.6.4. Caso de Uso

Para ilustrar el potencial que posee la aplicación, se muestra el siguiente caso de uso.

Suponiendo que se desea saber cuáles fueron los eventos más importantes en América del Norte pertenecientes a la categoría de Elecciones durante el 19 de Septiembre se deben

realizar los siguientes pasos:

1. Entrar a la aplicación.
2. Deseleccionar todas las categorías, excepto Elecciones.
3. Seleccionar América del Norte.
4. Navegar hasta el 19 de Septiembre.
5. Hacer click en las noticias pertenecientes al intervalo del 19 de Septiembre.

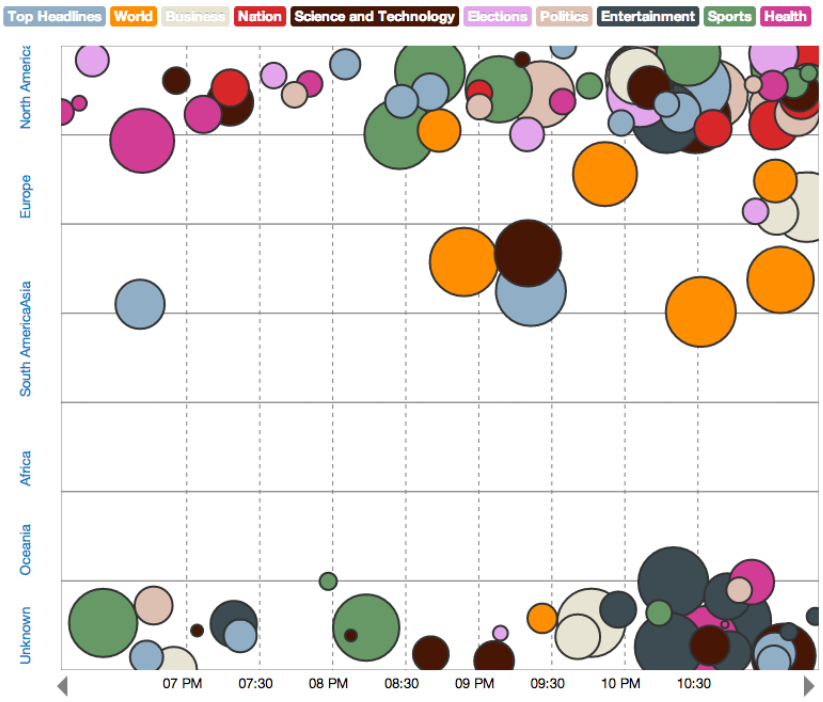

Figura 3.13: Paso 1: Entrar a la Aplicación

Al revisar las noticias de ese día se puede observar que ocurrieron eventos tales como: el inicio de la inscripción electrónica para las elecciones de Estados Unidos en California y reacción de los senadores Estadounidenses frente a declaraciones del candidato a la presidencia Mitt Romney.

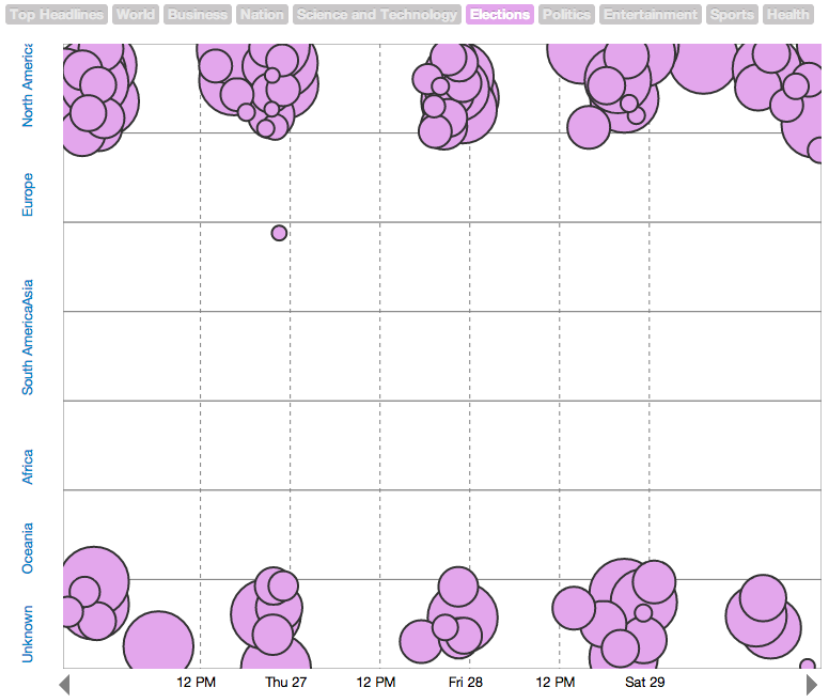

Figura 3.14: Paso 2: Categoría de Elecciones Seleccionada

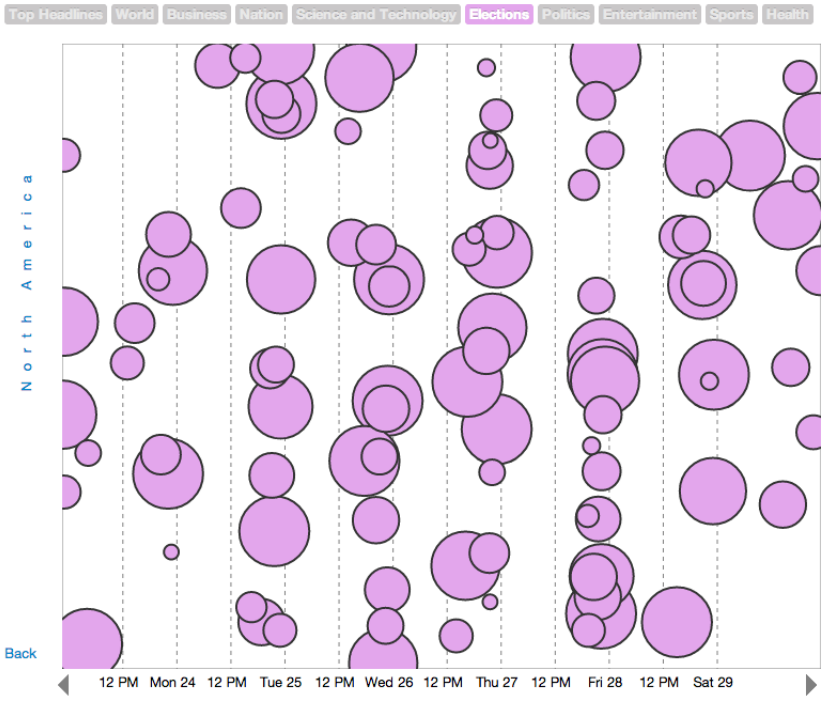

Figura 3.15: Paso 3: Filtro por Continente: América del Norte

Figura 3.16: Paso 4: Navegación en el Tiempo

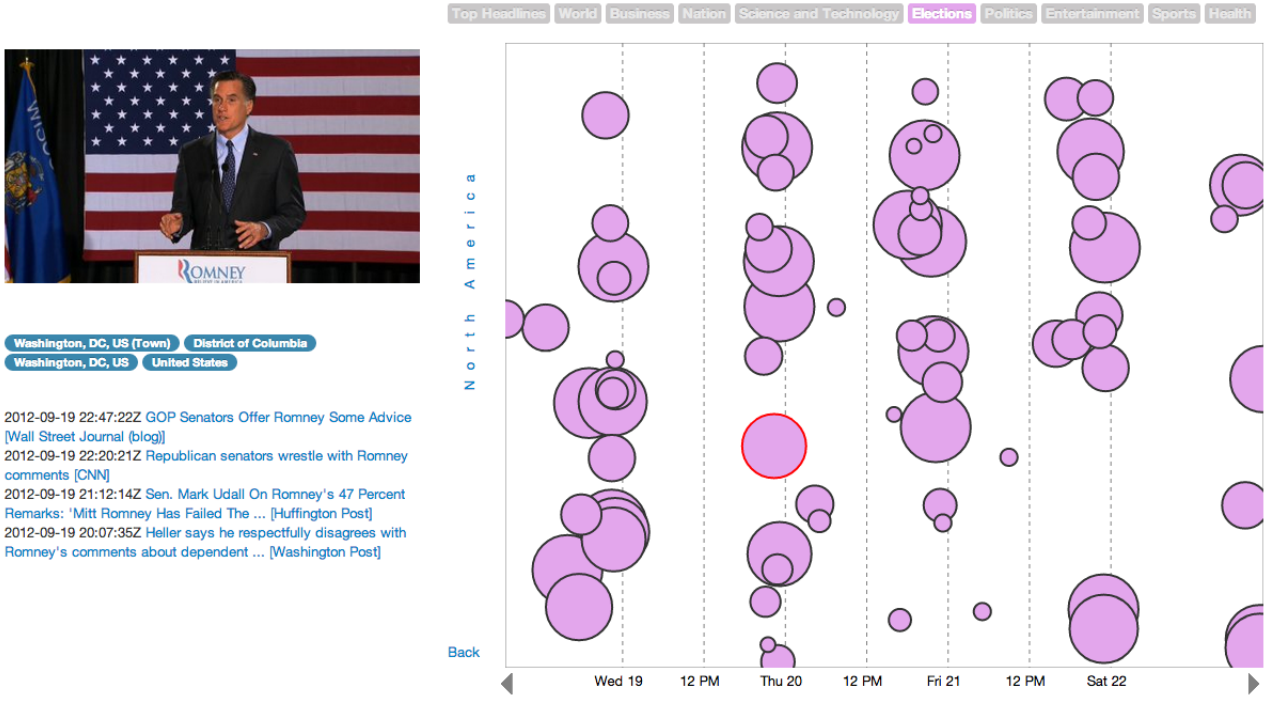

Figura 3.17: Paso 5: Selección de un Conjunto de Noticias

Capítulo 4

Discusión de la Solución

Existen varias razones para crear visualizaciones de datos. Entre ellas se encuentran: responder preguntas, descubrir preguntas, tomar decisiones, encontrar patrones, presentar argumentos, etc. Por lo que la validación de una solución que incorpora una visualización debiese hacer parte de estos objetivos. En particular, se validará que la visualización es capaz de encontrar más información sobre los datos a primera vista y que el modelo de datos utilizado en la aplicación sirve para visualizar distintos datos independiente de su fuente.

4.1. Intervalos de Tiempo sin Datos

Al programar el cron que ejecuta el script de recolección de datos, se asumió que recolectar noticias una vez al día sería suficiente para obtener las noticias de un día completo. En la figura 4.1 se observa que existen intervalos de tiempo en los que no existen noticias sobre elecciones. Notar que la Figura 4.1 está graficando las noticias de la categoría "Elections" que corresponden a tres días.

Inspeccionando un día cualquiera, también se puede notar que cerca de la hora en que se recolectan las noticias (12 A.M.) existen muchas noticias, pero pasada esa hora la cantidad de noticias recolectada disminuye considerablemente (Figura 4.2).

Que no se registren noticias en estos intervalos puede ocurrir por distintas razones: como se mencionó anteriormente, puede que la recolección de noticias una vez al día no sea suficiente para esta categoría. También puede ocurrir que la fuente de datos no proporcione datos en dichos intervalos de tiempo o que no proporcione datos de esta categoría en estos intervalos de tiempo.

Lo importante es que la visualización fue capaz de descubrir la pregunta: "¿Por qué no se recolectan noticias todo el tiempo?". Lo anterior mediante el reconocimiento del patrón de que periódicamente en ciertos intervalos no hay noticias. Finalmente, la visualización permite entregar información para por ejemplo tomar la decisión de correr el cron de noticias 2 veces al día para así obtener noticias todo el tiempo.

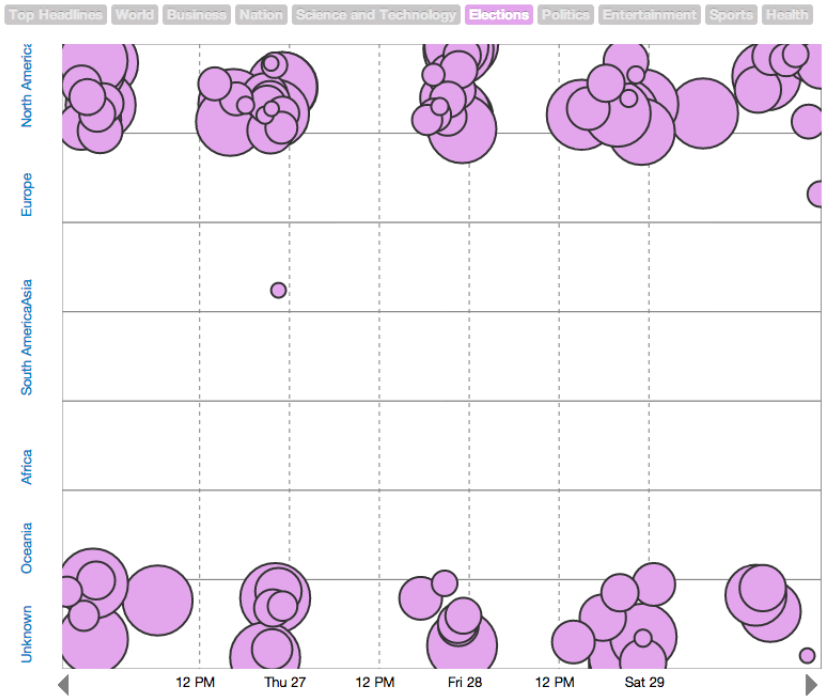

Figura 4.1: Intervalos de tiempo sin datos (3 días)

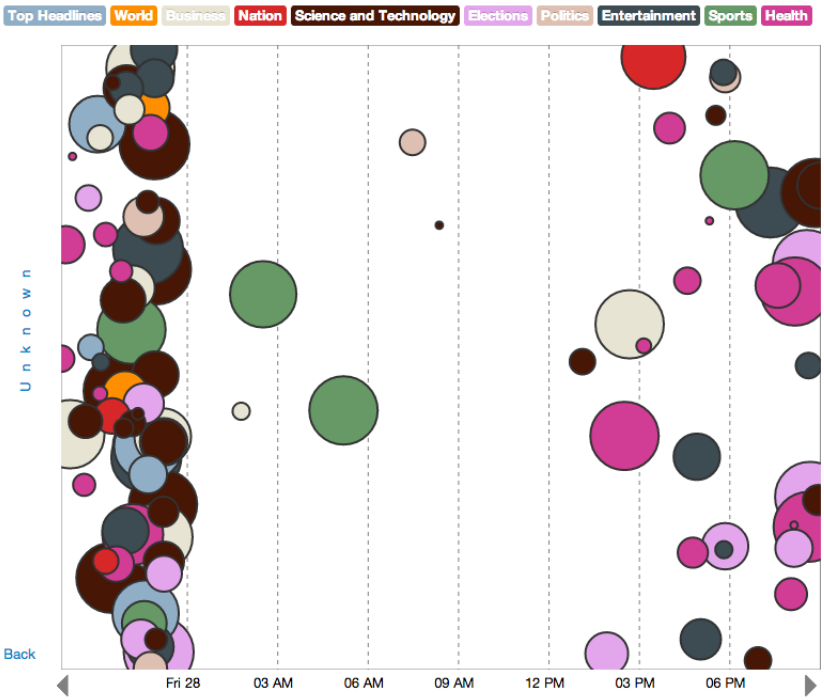

Figura 4.2: Intervalo de tiempo sin datos (1 día)

4.2. Datos del Mundo según Estados Unidos

Los datos recolectados utilizando Google News corresponden a la edición de Estados Unidos. Por lo mismo, la georeferencia de los datos utilizados en este documento apuntan ma-

yormente a América del Norte.

De todas formas, Google News edición Estados Unidos es capaz de entregar la categoría "World", la cual tiene que ver con las noticias que son mayoritariamente publicadas en Estados Unidos pero que hablan del resto del mundo.

Estas noticias se pueden aislar utilizando la aplicación utilizando los filtros de categorías, obteniendo el resultado de la Figura 4.3. Se muestran noticias en un intervalo de 5 días de la categoría "World" en sus respectivos continentes.

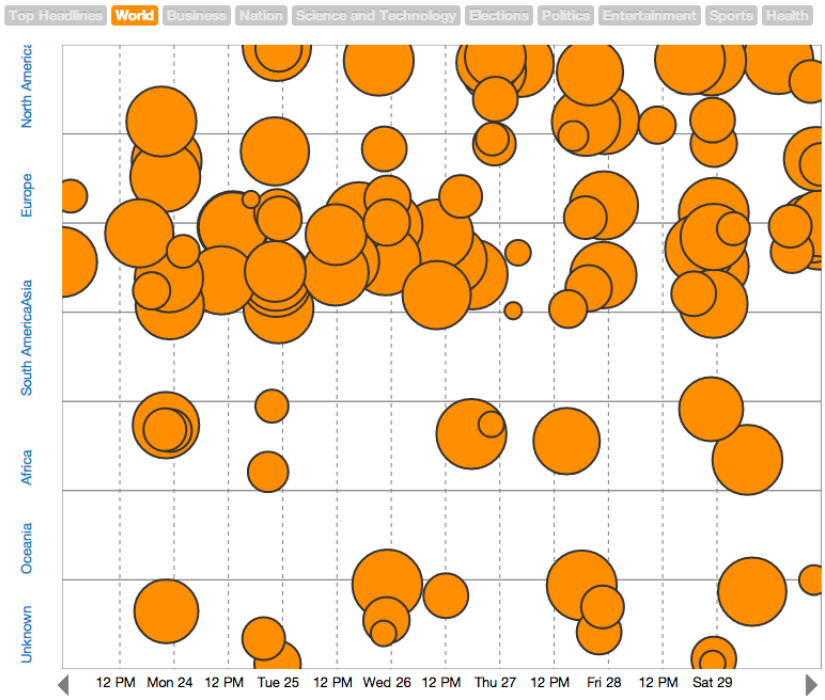

Figura 4.3: Distribución de noticias de categoría "World"

Se puede apreciar que la mayoría de las noticias se reparte entre Europa, América del Norte, Asia y África. Y que al menos en este intervalo de tiempo no se registran noticias de América del Sur y Oceanía. También que la mayoría de las noticias de "World" habla de Asia.

Lo anterior se puede deber a que ocurrió un evento en Asia en este intervalo de tiempo, o que en realidad, Asia es más relevante para Estados Unidos que América del Sur. La visualización también permite ver solamente las noticias correspondientes a la categoría World que fueron clasificadas como noticias de Asia (Figura 4.4). Inspeccionando las noticias, se aprecia que en esa semana ocurre un ataque aéreo a Siria, lo que puede explicar por qué la mayor cantidad de las noticias de World apuntan a Asia.

De cualquier manera, la visualización es capaz de mostrar este resultado que podría no ser el esperado. Se podría esperar que las noticias del mundo efectivamente hablen de todo el mundo uniformemente y no solamente de Asia.

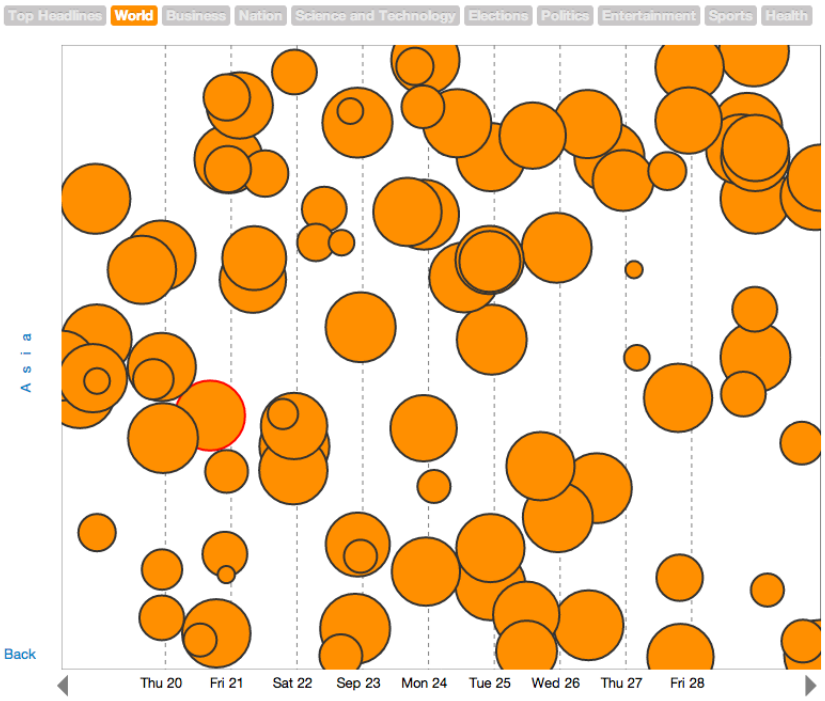

Figura 4.4: Distribución de noticias de categoría "World" georeferenciadas en Asia

4.3. Datos sin Clasificación de Continente

Anteriormente, en la Sección 3.4, se afirmó que cerca del 65 % de los datos pudo ser georeferenciado. El dato anterior, si bien no se muestra un número explícito en la visualización es coherente con lo que se observa en cualquiera de las figuras de este documento.

De todas formas, surge la inquietud de qué sucede con los datos que no pudieron ser georeferenciados. La visualización permite ver la clasificación de los clusters de cierta categoría. En particular, visualizando los datos de la categoría "Entertainment" se observa que más de la mitad de estos datos no están georeferenciados (Figura 4.5).

Nuevamente la visualización formula preguntas, como: "¿Por qué las noticias de Entertainment son más difíciles de ubicar que las de World?". En la Figura 4.3 se puede apreciar que la mayoría de las noticias de la categoría "World" se clasifican de manera correcta.

4.4. Visualización de Datos de Feedzilla

Como ya se ha mencionado a lo largo de este informe, se ha realizado un desarrollo paralelo con datos provenientes de Feedzilla. Esto se realizó con el fin de mostrar que el modelo de datos construido para la visualización no solamente es capaz de visualizar los datos de Google News, sino que también es capaz de visualizar datos provenientes de otras fuentes. Dado que fue posible construir dicha aplicación, se da por validado que la aplicación fue capaz de al menos visualizar dos conjuntos de datos de distintas fuentes.

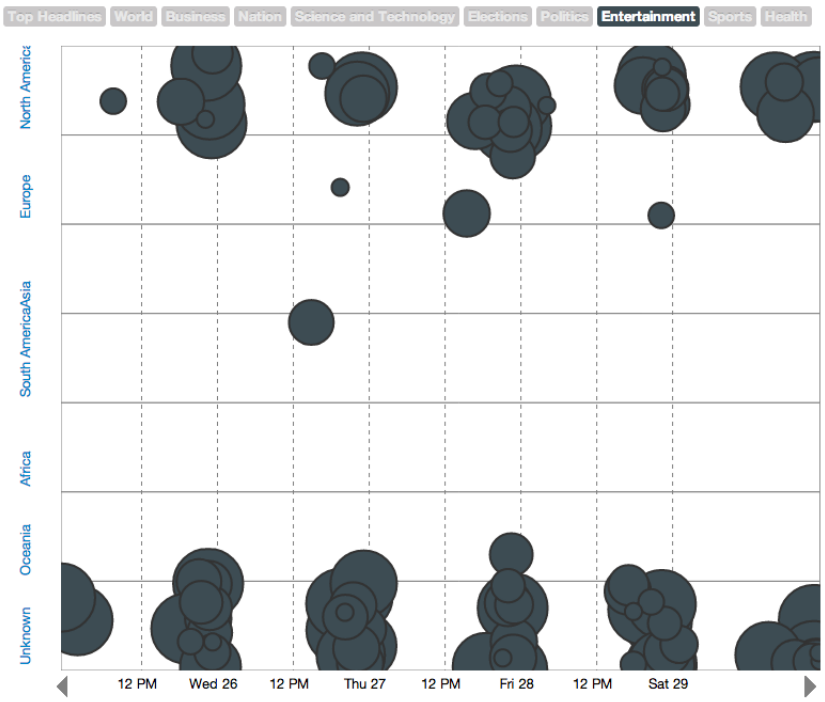

Figura 4.5: Noticias de entretenimiento

De todas formas, ¿Para qué podría servir visualizar otras fuentes de datos? Podría ser interesante visualizar datos no estructurados como los de Twitter, donde para visualizarlos con la aplicación se necesitaría un procesamiento de datos mayor que el realizado en este trabajo para que los datos calcen en el modelo de base de datos.

También podría ser interesante visualizar otro tipo de datos que posean una estructura similar a la de las noticias, como por ejemplo sismos. Los sismos poseen ubicación geográfica, tiempo en que ocurre, magnitud (que podría ser una suerte de relevancia), profundidad que podría ser representada como una categoría.

Dado que se tiene Eventsvis con dos sets de datos diferentes es posible realizar una comparación de la calidad de los datos.

4.5. Comparación entre Datos de Google News y Feedzilla

Comparando ambas visualizaciones es posible notar que los datos de Feedzilla son de menor calidad que los de Google News, tanto en sus categorías como en sus noticias.

Esto se nota revisando las categorías entregadas por Feedzilla, en las que hay algunas como Oddly Enough y Fun Stuff que entregan noticias no muy relevantes. También se nota en que las fuentes de las noticias no son de tanto renombre como las fuentes de las noticias de Google News. Incluso hay categorías duplicadas como Sports que está dos veces o IT y

Technology que pueden tener muchas noticias en común.

Probablemente se debe a que los datos de Google News posee una selección editorial a diferencia de los de Feedzilla. También porque los sitios que aparecen en los resultados de Google News hacen mucho SEO (Search Engine Optimization), entregando mucha información para aparecer en los resultados de las búsquedas en Google.

Otro de los hechos a tomar en cuenta es que la misma API de Feedzilla entrega menos información sobre las noticias que Google News haciendo que esto se vea reflejado en la inspección al visualizar.

Capítulo 5

Conclusiones

Fue posible la construcción de una aplicación de visualización capaz de recolectar datos de Google News diariamente. De esta forma, sería posible dejar la aplicación en un servidor y diariamente saber que ocurre en el mundo con la limitante de que la base de datos podría no ser lo suficientemente eficiente.

A lo largo del trabajo, se mostró que es posible visualizar diferentes sets de datos provenientes de distintas fuentes. Se construyen las aplicaciones Eventsvis Google News y Eventsvis Feedzilla que recolectan y visualizan datos provenientes de Google News y Feedzilla respectivamente. La aplicación queda disponible en Github para seguir siendo extendida ya sea en sus funcionalidades o para que se pueda probar con otros conjuntos de datos.

Simplemente con inspección visual fue posible encontrar información adicional de los conjuntos de datos. En particular, se logró determinar que ciertas categorías de noticias como Entertainment poseen un contenido más difícil de georeferenciar que el de otras categorías como World. También se lograron determinar propiedades en el conjunto de datos como por ejemplo que no se están almacenando noticias que pertenecen a ciertos intervalos de tiempo.

También solamente con inspección visual, fue posible encontrar información de los datos propiamente tal. En la sección 4.2 se muestra la distribución de continentes de las noticias de World de la versión de Estados Unidos de Google News. Se plantea la pregunta de por qué la mayoría de las noticias son de Asia y se logra identificar un evento dentro de estas noticias.

La visualización fue capaz de mostrar que un conjunto de datos es de mejor calidad que otro. En particular muestra que los datos de Google News son de mejor calidad que los datos de Feedzilla y deja abierta la pregunta de por qué pasa esto.

De lo anterior se concluye que la aplicación construida sirve para inspeccionar datos de noticias, porque se puede inferir información adicional de los datos. Y también se concluye que la aplicación sirve para saber que ocurre en cierto intervalo de tiempo en el mundo, porque es capaz de recolectar y visualizar noticias.

Capítulo 6

Trabajo Futuro

6.1. Otras Fuentes de Datos

En este trabajo se diseña un modelo de datos que tiene como objetivo almacenar los atributos usuales que tienen las noticias y servir de esquema para que la visualización pueda mostrar datos de cualquier fuente. De esta forma, si se quisiera visualizar datos de otra fuente, el trabajo estaría en adaptar estos datos al esquema de base de datos planteado en la Sección 3.2.

¿Qué otras fuentes de datos se podrían utilizar? Una de las ideas originales de este trabajo era visualizar eventos encontrados en Twitter, la cual se desechó porque el objetivo central de este trabajo es la visualización y no el proceso de datos. Procesar los datos de Twitter que no están estructurados, en comparación a procesar los datos de Google News es un esfuerzo mucho mayor.

Un acercamiento para visualizar eventos de Twitter sería cambiar la relevancia por la cantidad de Tweets que hablan del evento, las categorías de noticias podrían mantenerse en caso de que el proceso de datos de Twitter también clasifique los eventos por categoría o podría cambiarse por keywords identificados en los tweets. El eje de la fecha y el eje de la ubicación geográfica debiese mantenerse.

Otro experimento interesante sería el visualizar datos que no sean intrínsecamente noticias, pero sí eventos. Un ejemplo de esto es que podrían visualizarse datos referentes a sismos. Un primer acercamiento a la visualización de estos datos sería mantener en el eje X el tiempo y mantener en el eje Y la ubicación. Cambiar la relevancia por magnitud, es decir, el tamaño de los círculos dependería de la magnitud y cambiar la categoría por la profundidad.

De manera más abstracta, podría ser posible visualizar cualquier tipo de datos que tengan como atributos fecha, ubicación, categoría (u otra etiqueta) y relevancia.

6.2. Eficiencia en Base de Datos

Una de las interrogantes que surge al momento de implementar Eventsvis es si la base de datos es capaz de manejar una gran cantidad de noticias. Al menos se comprobó que 11638 artículos correspondientes a 11 días de noticias la base de datos era capaz de responder las consultas en un tiempo prudente.

Queda pendiente el realizar pruebas con conjuntos más grandes de datos y también con otras bases de datos SQL como MySQL o otras no SQL como Mongo o bases de datos de grafos.

6.3. Mejoras en la Visualización

Algunas funcionalidades de la aplicación quedaron incompletas o sin hacer debido a que se priorizaron otras tareas. En particular, se le dio un poco más de énfasis al proceso de los datos de lo que se hubiese deseado.

Una de las funcionalidades que quedó pendiente fue poder filtrar por una ubicación más específica y no solamente por continente. La aplicación permite hacer click en un continente y ver todas las noticias correspondientes a ese continente. Lo que queda pendiente es poder hacer click en una ubicación más específica como una ciudad, región o país y filtrar las noticias correspondientes a ese lugar.

También quedó pendiente la mejora de la navegación en el tiempo para ir a una fecha específica fácilmente. Actualmente no es posible elegir una fecha y ver qué pasó en esa fecha. Por ejemplo si se quisiera revisar las noticias que ocurrieron 5 días atrás, lo que hay que hacer es navegar a través de las noticias actuales y retroceder hasta llegar a 5 días atrás.

Otra funcionalidad pendiente es la de ajustar el intervalo de tiempo que se desea visualizar. Por ahora, el intervalo de tiempo es automático y depende del intervalo de tiempo generado por los 100 clusters que se están visualizando. Además, la cantidad de elementos que se visualizan está en el código de la aplicación y solamente es posible cambiarlos desde ahí. Esto podría haberse hecho variable y que en la visualización se eligiera cuantos clusters visualizar.

Bibliografía

[1] Feedzilla Newsfeed API. `http://www.feedzilla.com/api-overview`.

[2] Google News Search API. `https://developers.google.com/news-search/`.

[3] Yahoo! Placemaker API. `http://developer.yahoo.com/geo/placemaker/`.

[4] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, Robert C. Miller. Twitinfo: Aggregating and visualizing microblogs for event exploration. *CHI 2011*, 2011.

[5] Amit Sheth, Hermant Purohit, Ashutosh Jadhav, Pavan Kapanipathi, Lu Chen. Understanding events through analysis of social media. *PROC WWW'11*, 2011.

[6] Brian Johnson, Ben Shneiderman. Treemaps: a space-filling approach to the visualization of hierarchical information structures. *Visualization, 1991. Visualization '91, Proceedings., IEEE Conference on*, 1991.

[7] McHill R. Cleveland, W. S. Graphical perception: Theory, experimentation and application to the development of graphical methods. *J. Am. Stat. Assoc.*, 1984.

[8] Chris Johnson. Top scientific visualization research problems. *IEEE 2004*, 2004.

[9] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 1986.

[10] Michael Bostock, Vadim Ogievetsky, Jeffrey Heer. D3: Data-driven documents. *IEEE Trans. Visualization and Comp. Graphics (Proc. InfoVis)*, 2011.

[11] Marcos Weskamp. Newsmap. `http://newsmap.jp/`, 2004.