Easter eggsperiments

Author : Tim Sherratt

Tagged as : ABC Radio National, newspapers, Trove, TroveBot, WWI

Date : April 20, 2014

No, nothing to do with Easter or eggs, but it's Easter Sunday and who can resist a good opportunity for a bad pun?

This is another catch-up post, pulling together some recent experiments. If nothing else, it'll help me keep track of things I'm otherwise likely to forget.

WWI Faces

In our last instalment I was playing around with some WWI data from the State Library of South Australia. I'm really pleased to report that SLSA staff have used my experiments to help them add Trove links to more than 6,000 of their Heroes of the Great War records. Here's an example — note the 'article' link which goes straight to a digitised newspaper article in Trove. With some good data and bit of API wrangling we've now established rich linkages between an important WWI resource and Trove. Win!

I've also continued my fiddling with articles from the Adelaide *Chronicle* as I start to think about how Trove's newspapers might be used in a WWI exhibition being developed by the National Library. At the end of my last post I'd created a list of articles from the *Chronicle* that were likely to include biographical details of WWI personnel. I knew that many of these included portrait photos, so I filtered them on Trove's built-in 'illustrated' facet and saved the page images for the remaining articles. You can browse the resulting collection of pages on Dropbox. As you can see there are indeed many portraits of service people.

So the next step was to try and extract the portraits from the pages. This was rather familiar territory, as I'd already used a facial detection script to create The Real Face of White Australia. But I wasn't sure how the pattern recognition software would cope with the lower quality newspaper images. After getting all the necessary libraries installed (the hardest bit of the whole process), I pointed the script at the page images and… it worked!

From 141 pages I extracted 1,738 images, and most of them were faces. You can browse all 1,738, but be warned, I've just dumped them onto a single page and added a bit of Isotope magic — so they'll take a fair while to load and your browser might object. You'll also notice that I haven't tried to filter out photos of non-service people, I just wanted to see if it worked. And it does. Even in this rough form you can sense some of the emotive power. What's really amazing is the way that even small images of faces in group photographs were identified. All I was aiming for at this stage was a proof of concept — yes, I can extract photos of WWI service people from newspapers. Hmmm…

Trove in space

All the faces above were from one South Australian newspaper. Several years ago I worked on a project to map places of birth and enlistment of WWI service people, and while I have no interest in the national mythologies surrounding WWI, I do still wonder about the local impact of war — all those small communities sending off their sons and daughters…

So I'm wondering whether we might be able to use the digitised newspapers in Trove to navigate *from place to face*. To choose a town anywhere in Australia, and present photographs of service personnel published in nearby newspapers.

I now know I can extract the photos, but how can we navigate Trove newspapers by location? Time for a new experiment…

The Trove API provides a complete list of digitised newspaper titles. You'll notice that some of the titles include a place name as part of the summary information in brackets, while many others will include place names in their titles, for example:

- Illawarra Daily Mercury (Wollongong, NSW : 1950 – 1954)
- Hawkesbury Herald (Windsor, NSW : 1902 – 1945)
- Kiama Examiner (NSW : 1858 – 1859)
- Narromine News and Trangie Advocate (NSW : 1898 – 1955)

I haven't had much luck getting automated named entity extraction tools to work on short text strings like this, so I decided to roll my own using Geoscience Australia's Gazetteer of Australia 2012. I opened up the GML file containing all Australian places and saved the populated locations to my own Mongo database. This gave me a handy place name database, complete with geo-locations.

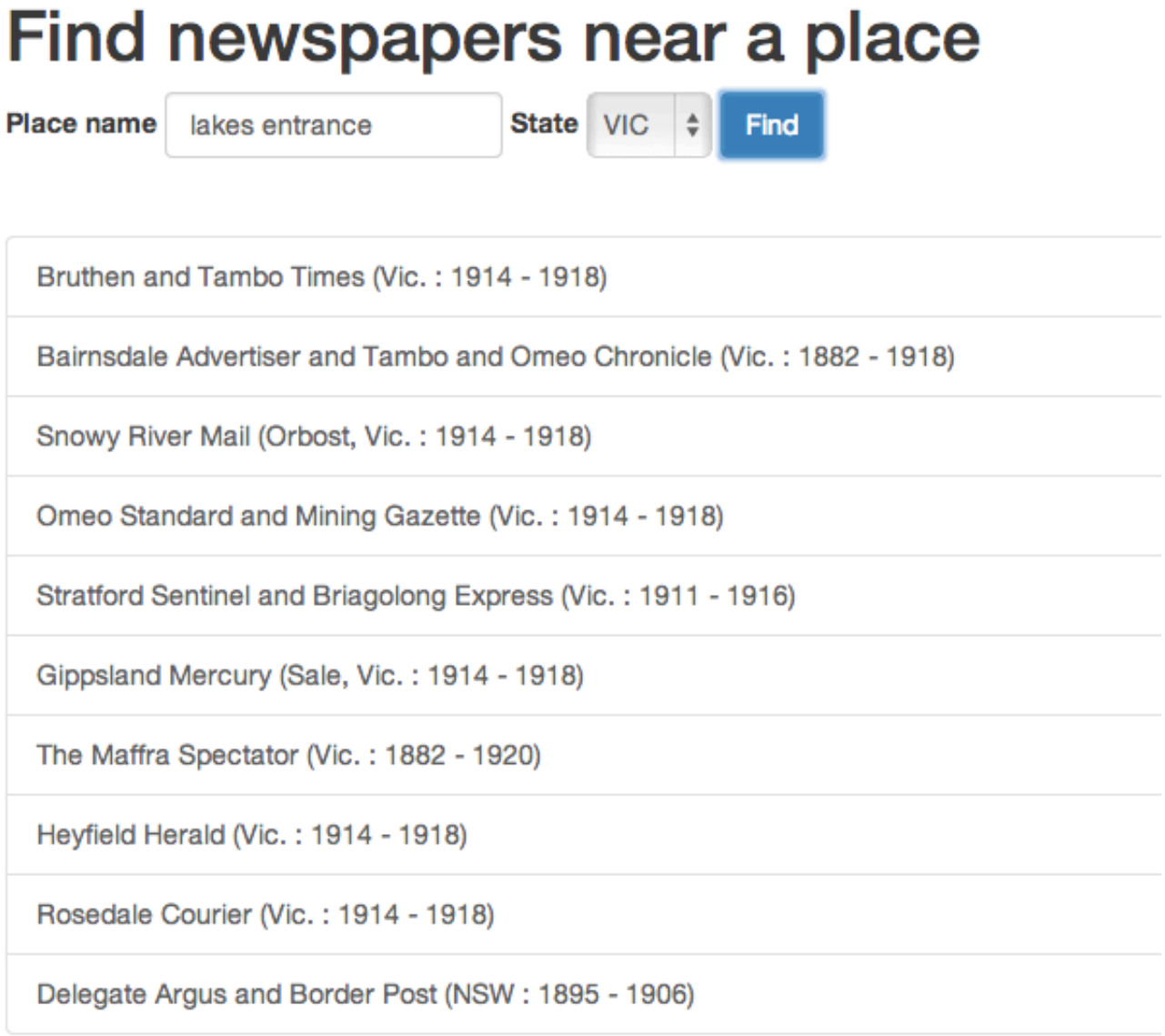
Next I went to work on the newspaper titles. Extracting the places from the summary information was easy because they followed a regular pattern, but finding them in the body of the title was trickier. First I had to exclude those words that were obviously not place names. Aside from the usual stopwords ('and' and 'the'), there are many words that commonly occur in newspaper titles — 'Herald', 'Star', 'Chronicle' etc. To find these words I pulled apart all the titles and calculated the frequency of every word. You can explore the raw results — 'Advertiser' (116) wins by a large margin, with 'Times' (67) in second place. From these results I could create a list of words that I knew were not places and could safely be ignored.

Then it was just a matter of tokenising the titles (breaking them up into individual words), removing all the stopwords (the standard list and my special list), and then looking up the words in my place name database. I did this in two passes, first as bigrams (pairs of words), and then as single words — this allowed me to find compound place names like 'North Melbourne'. The Trove API gives you s 'state' value for each title, so I could use this in the query to increase accuracy.

If I found a place name, I added the place details, including the latitude and longitude, to the title record from the API and included it in my own newspaper title database.

So I ended up to two databases — one with geolocated places, and another with geolocated newspapers. That meant I could build a simple interface to find newspaper titles by place. It's

nothing fancy — just another proof of concept — but it works pretty well. Just type in a place name and select a state and a query is run against the place name database. If the place is found then the latitude and longitude is fed to the titles database to find the closest newspapers. After removing some duplicates, the 10 nearest newspapers are displayed.



Find Trove newspapers by place

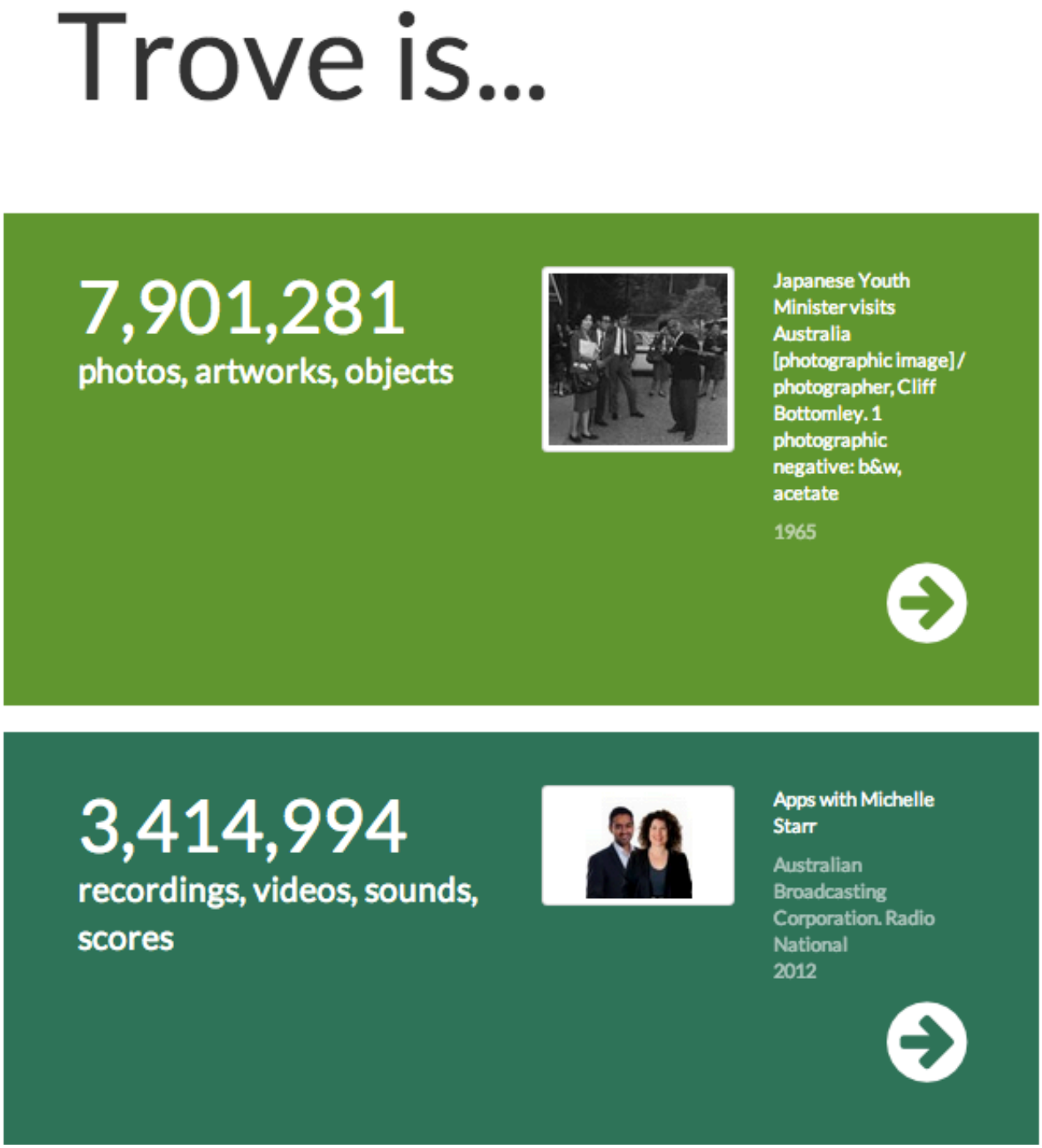
Building some sort of map interface on top of this is pretty trivial. What's more important is to do some analysis of my place matching to see what I might have missed. But so far so good!

Trove is...

Trove is more than newspapers. This is a message the Trove team tries to emphasise at every opportunity. The digitised newspapers are an incredible resource of course, but there's so much other interesting stuff to explore.

To try and give a quick and easy introduction to this richness, I created a simple dashboard-type

view of Trove, imaginatively titled [Trove is…]



What is Trove?

Trove is… gives a basic status report on each of the 1o Trove zones, with statistics updated daily (except for the archived websites as there's no API access at the moment). The BIG NUMBERS are counter-balanced by a single randomly-selected example from each zone. It's a summary, an overview, a portal and a snapshot. Reload the page and the zones will be reordered and the examples will change.

It's pretty simple, but I think it works quite well, and thanks to Twitter Bootstrap it looks really nice on my phone! But while the idea was simple, the implementation was pretty tricky — particularly the balance between randomness and performance. If all the examples were truly random, drawn from the complete holdings to Trove on every page reload, you'd spend a lot of time watching spinning arrows waiting for content to appear. I tried a number of different approaches and finally settled on a system where random selections of 100 resources per zone are made every hour by background processes and cached. When you load the page, this cache is queried and an item selected. So if you keep hitting reload you'll probably notice that some examples reappear. It's random, but at any moment the pool of possibilities is quite limited. Come back later in the day and everything will be different.

Anyway, if anyone asks you what Trove is, you now know where to point them…

Who listens to the radio?

After a lot of hard work, the Trove team was excited to announce recently that more than 200,000 records from 54 ABC Radio National programs were available through Trove.

To make it a bit easier to explore this wonderful new content, I created a simple search interface. All it really does is help you build a query using the RN program titles, and then sends the query off to Trove. Not fancy, but useful (my family motto).

Of course, I couldn't leave my Twitter bot family out of the action. @TroveBot has been Radio National enabled. Just tweet the tag #abcrn at him to receive a randomly-selected Radio National story. To search for something amidst the RN records, just tweet a keyword or two and add the #abcrn tag to limit the results. Consult the TroveBot manual for complete operating instructions.

In a word…

But the Radio National content is not just findable through the Trove web interface — all that lovely data is freely accessible through the Trove API. That includes just about every segment of every edition of the ABC's flagship current affairs programs, AM, PM, and The World Today from 1999 onwards. What sort of questions could you ask of this data?

I'll be writing something soon on the Trove blog about accessing these riches, but I couldn't resist having a play. So I harvested all the RN data via the API and built a new thing…

What's in a word?

It's called In a word: Currents in Australian affairs, 2003–2013, and for once it's quite well documented, so I won't go into details here. I'll just say that it's one of my favourite creations, and I hope you find it interesting.

Addendum (21 April) — The Tung Wah Newspaper Index

See, I told you I forget things…

I recently finished resurrecting the Tung Wah Newspaper Index. Kate has described the original project on her blog, and there's a fair bit of contextual information on the site, so I won't go into details here. Suffice it to say it's an important resource for Chinese Australian history that had succumbed to technological decay.

The original FileMaker database has been MySqld, Solrised, and Bootstrapped to get it all working nicely. I also took the opportunity to introduce a bit of LOD love, with plenty of machine-readable data built-in.

The whole site follows an interface as API type pattern. So if you want a resource as JSON-LD, you just change the file extension to .json. To help you out, there are links at the bottom of each page to the various serialisations, and of course you can also use content negotiation to get what you're after. There's some examples of all this in the GitHub repository, as well as a CSV dump of the whole database.