# Enriching WWI data with the Trove API

**Author :** Tim Sherratt

**Tagged as :** API, newspapers, Trove, World War I

**Date :** March 9, 2014

I can't resist a challenge, particularly when it involves lots of new historical data and an excuse to muck around with the Trove API. So when Katie Hannan from the State Library of South Australia asked me about putting the API to work to enrich one of their World War I datasets, I had to dive in and have a play.

The dataset consists of references to South Australian WWI service personnel published in the Adelaide *Chronicle* between 1914 and 1919. In a massive effort starting back in 2000, SLSA staff manually extracted more than 13,000 references and grouped them under 9709 headings, mostly names. You can explore the data in the SLSA catalogue as part of the Heroes of the Great War collection.

It's great data, but it would be even better if there was a direct link to each article in Trove — hence Katie's interest in the possibilities of the API!

HEROES OF THE GREAT WAR: THEY GAVE THEIR LIVES FOR KING AND COUNTRY.

Chronicle (Adelaide, SA : 1895 – 1954) 14 Sep 1918, p. 24, http://nla.gov.au/nla.news-page8611372

Katie sent me a spreadsheet containing the data. Each row corresponds to an individual entry and includes an identifier, a name, a year, and a list of references separated by semicolons. My plan was simple, for each row I'd construct a search based on the name, then loop through the search results to try find an article that matched the date and page number of each reference. This might seem a bit cumbersome, but currently there's no way of searching Trove for newspaper articles published on a particular day.

You'll find all of the code on GitHub. I've tried to include plenty of comments to make it easy to follow along.

Let's look at the entry for **Lieutenant Frank Rosevear**. It includes the following references: **'Chronicle, 7 September 1918, p. 27, col. c;Chronicle, 14 September 1918, p. 24, col. a p.38, col. d'**. If you look closely, you'll see that there's two page numbers for 14 September 1918, so there's actually three references included in this string. The first thing I had to do was to pull out all the references and format them in a standard way.

Assuming that the last name was the surname, I then constructed a query that searched for an exact match of the surname together with at least one of the other names. In Lieutenant Rosevear's case the query would've been 'fulltext:"Rosevear" AND ("Lieutenant" OR "Frank")'. Note the use of the 'fulltext' modifier to indicate an *exact* match. To this query I added a date filter to limit the search to the specified year and an 'l-title' value to search only the Adelaide *Chronicle*.

You can see the results for this query in my Trove API console. Try modifying the query string to see what difference it makes.

Once the results came back from the API I compared them to the references, looking for matches on both the date and page number. You might notice that the second result from the API query, dated 7 September 1918, is a match for one of our references. Yay! This gets saved to a list of strong matches. But what about the other references?

Just in case there's been a transcription error, or the page numbering differed across editions, I relax the rules a bit in a second pass and accept matches on the date, but *not* the page. These are saved to a list of close matches.

This second pass doesn't help much with Lieutenant Rosevear's missing references, so we have to broaden our search query a bit. This time we search on the surname only. Bingo! The first result is a match and points us to one of the 'Heroes of the Great War' series.

Lieut. F. ROSEVEAR.

‘HEROES OF THE GREAT WAR: THEY GAVE THEIR LIVES FOR KING AND COUNTRY.’, Chronicle (Adelaide, SA : 1895 – 1954) 14 Sep 1918, p. 24, http://nla.gov.au/nla.news-article87553854

It took a while, but my script eventually worked it’s way through all 9709 entries like this, writing the results out to csv files containing the strong and close matches. It also created a summary for each entry, listing the original number of references alongside the number of strong and close matches.

Ever since I read Trevor Munoz’s post on using Pandas with data from the NYPL’s *What’s on the Menu?* project, I’ve wanted to have a play with it. So I decided to use Pandas to produce some quick stats from my results file.

```
>>> import pandas as pd  >>> df = pd.read_csv('data/slsa_results.csv')  # How many en
tries?  >>> len(df)  9709  # How many references?  >>> df['references'].sum()  13504
 # How many entries had strong matches?  >>>len(df[df.strong > 0])  6440  # As a perc
entage thank you...  >>> 100 * len(df[df.strong > 0]) / len(df)  66  # In how many en
tries did the number of refs = the number of strong matches  >>> len(df[df.references
 == df.strong])  4399  # As a percentage thank you...  >>> 100 * len(df[df.references
 == df.strong]) / len(df)  45  # How many entries had at least one strong or close ma
tch?  >>> len(df[df.total > 0])  8989  # As a percentage thank you...  >>> 100 * len(
df[df.strong > 0]) / len(df)  92
```

Not bad. The number of strong matches equalled the number of references in 45% of cases, and overall 66% of entries had a least one strong match. I might be able to get those numbers up by tweaking the search query a bit, but of course the main limiting factor is the quality of the OCR. If the article text isn't good enough we're never going to find the names we're after.

Katie tells me that the State Library intends to point volunteer text correctors towards identified articles. As the correctors start to clean things up, we should be able to find more matches simply by re-running this script at regular intervals.

But what articles should they point the volunteers to? Many of them included the title 'Heroes of the Great War', so they're easy to find, but there were others as well. By analysing the matches we've *already* found we can pull out the most frequent titles and build a list of likely candidates. Something like this:

```
title_phrases = [  'heroes of the great war they gave their lives for king and countr
y',  'australian soldiers died for their country',  'casualty list south australia ki
lled in action',  'on active service',  'honoring soldiers',  'military honors austra
lians honored',  'casualty lists south australian losses list killed in action',  'au
stralian soldiers died for his country',  'died for their country',  'australian sold
iers died for the country',  'australian soldiers died for their country photographs
of soldiers',  'quality lists south australian losses list killed in action',  'list
killed in action',  'answered the call enlistments',  'gallant south australians how
they won their honors',  'casualty list south australia died of wounds',  ]
```

Now we can feed these phrases into a series of API queries and automatically generate a list of articles that are likely to contain details of WWI service people. This list should provide a useful starting point for keen text correctors.

I might not have completely solved Katie's problem, but I think I've shown that the Trove API can be usefully called into action for these sorts of projects. Taking this approach should certainly save a lot of manual searching, clicking, cutting and pasting. And while I've focused on the South Australian data, there's no reason why similar approaches couldn't be applied to other WWI projects.

## Share this:

- Click to email this to a friend (Opens in new window)
- Click to print (Opens in new window)
- Click to share on Twitter (Opens in new window)
- Share on Facebook (Opens in new window)
- Click to share on Google+ (Opens in new window)
-