Determining PMIDs for articles cited in patents

Bhaven N. Sampat

July 9, 2016

Contents

1 Background

This set of scripts generates a set of PMIDs, unique identifiers in PubMed, for references cited in the non-patent reference section of issued patents. The resulting data files can be used to link publications from PubMed to citing patents. They have been used in various analyses linking NIH grants to the publications they generate and to the patents that cite these publications [1, 2, 3, 4].

If you are interested only in documentation on the output files, rather than running the Perl script yourself on new input data (or modifying it), jump to the end of this document.

To run this script yourself on new data you will need to obtain two sets of data: Medline XML files [5] and USPTO data on patents and their non-patent references [6]. Both should be available freely. You specify the subset of patents you wish to search in a list of patent numbers you provide in a file called **patent_IDs.txt**.

The input and output files provided here are based on extracting and parsing references from the 1,310,826 "life science" patents issued between 1976 and 2012. The included file **patent_IDs.txt** lists these patents. "Life science" patents are those in patent classes associated with NBER categories 1 and 3 (`http://www.nber.org/patents/`) and/or those listed in the FDA Orange Book or IMS Patent Focus). If you wish to parse a different set, use those patents instead in **patent_IDs.txt**.

After you download and unzip the raw XML files, and provide a set of input patents, the parser does the following:

1. Creates a more compact version of MEDLINE files

2. Reads the input patents
3. Finds the non-patent reference for each input patent, and extracts information from them into fields
4. Matches elements of the MEDLINE citation versus the patent citation, recording number of elements matched
5. Chooses the best match based on how many elements match
6. Outputs a set of “strict matches” (at least 4 elements match) [this is NPLCITE_12_STRICT.TXT]
7. Outputs a set of “loose” matches (at least 3 elements match)

The details of the code, and usage, are provided below.

2 Details

2.1 Input files

The input files are described in the file **init_variables.pl**. File names and paths can be changed there as necessary. All the code files read this file at the beginning and use the variables therein.

2.2 Code files

- **medline_parser_citations.pl**
  - input: XML Medline files extracted to a directory called *medline-files/zip*. (This path can be changed in line 6)
  - output: citations_data.txt

Reads all the MEDLINE XML files and outputs a simplified version that is faster to read. This file only needs to be run once. (Or each time a new version of MEDLINE is loaded.)

- **filter_patent_ID_list.pl**
  - input: $patent_ID_file, $reference_file
  - output: $output_reference_file_filtered

Reads the patent references and outputs only those that are listed in $patent_ID_file.

- **parse_patent_references.pl**
  - input: $output_reference_file_filtered
  - output: patent_references_data.txt

Reads the patent references provided and identifies key elements such as authors, title, pages, journal and volume.

- **triple_match.pl**
  - input: patent_references_data.txt, journals_key_list.txt, citations_data.txt
  - output: triple_matches.txt

Tries to find all potential matches between the patent references provided by parse_patent_references.pl and Medline citations. A number of criteria are used to decide whether a citation is similar enough to a patent reference. More than one potential match may exist for each patent reference.

- **filter_matches.pl**
  - input: patent_references_data.txt, triple_matches.txt, $output_reference_file_filtered
  - output: triple_matches3.txt, triple_matches4.txt

Chooses which one of the potential matches found by triple_match.pl is the best for each patent reference. Triple_match.pl often finds several candidat matches for a given patent reference. The best match is chosen by how many elements are similar between the patent reference and the Medline citation. The output file triple_matches3.txt includes matches with at least 3 elements in common between the Medline citation and the patent reference. The file triple_matches4.txt includes matches with at least 4 elements.

- **output_loose_results.pl**
  - input: patent_references_data.txt, triple_matches3.txt, $output_reference_file_filtered
  - output: $final_output_file_loose

Produces the output file using the matches found in triple_matches3.txt. Loose results should have more false positive and true positives.

- **output_strict_results.pl**

input: patent_references_data.txt, triple_matches4.txt, $output_reference_file_filtered
output: $final_output_file_strict

Produces the output file using the matches found in triple_matches4.txt. Strict results should have more true negatives and false negatives.

2.3 Usage

You can run the shell script **run_patent_matching_ID_list.sh** which executes the following commands:

```
./filter_patent_class.pl
./parse_patent_references.pl
./triple_match.pl
./filter_matches.pl
./output_loose_results.pl
./output_strict_results.pl
```

3 Output files

If you are interested in only the output files (based on running the algorithm on all “life science” patents issued by the end of 2012 against the 2014 version of MEDLINE) they are in files **NPLCITE_12_STRICT.TXT** and **NPLCITE_12_LOOSE.TXT**, for the more and less conservative implementations of the algorithm respectively (more on the difference below). If you prefer a Stata file, **nplcite-pmid-2012.dta** includes all patents from the input set that match a PMID. The “strict” flag indicates whether the match is based on the conservative (strict) implementation (1=yes).

4 Acknowledgements

The original script was written by Raul Rodriguez-Esteban, who also helped with this documentation. It was further refined through funding from the NSF Science of Science Policy Program (Award SBE-0738142).

5 TODO

- Report the various benchmarking results

6 Notes

- You can now also download USPTO citations directly from Google hosted USPTO bulk patent data:

`https://bulkdata.uspto.gov/`

- We focused on life science patents for the outputfiles provided. You could instead include any other patents in patent_IDs, though the matching may be less accurate for non-life science patents (more false positives).
- While I haven't tried this, I imagine this is easily adaptable to other sources of publication data, e.g. *Web of Science*
- This script was created long ago, using fairly simple matching techniques. While it delivers reasonable results, there are now more sophisticated efforts to improve match rate, including for example: `https://www.ideals.illinois.edu/handle/2142/54885`.