THE UNIVERSITY OF MELBOURNE

FACULTY OF BUSINESS & ECONOMICS

Melbourne Institute Working Paper Series

Working Paper No. 15/14

Harmonising and Matching IPR Holders at IP Australia

T'Mir D. Julius and Gaétan de Rassenfosse

MELBOURNE INSTITUTE®
of Applied Economic and Social Research

Harmonising and Matching IPR Holders at IP Australia*

T'Mir D. Julius and Gaétan de Rassenfosse
Melbourne Institute of Applied Economic and Social Research, and Intellectual Property Research Institute of Australia, The University of Melbourne

Melbourne Institute Working Paper No. 15/14

ISSN 1328-4991 (Print)

ISSN 1447-5863 (Online)

ISBN 978-0-7340-4354-2

July 2014

***** This Working Paper explains the methodology for Release 1.0 of the Melbourne Institute IP Australia Dataset. Please cite this Working Paper when using the underlying data. The dataset accompanying this paper is presented as a good faith service to the scientific community and the public in general. The information herein provided should not be viewed as formally accurate scientific data. If any inaccuracy is observed, please inform the authors as soon as possible for verification and correction as necessary. Funding for this project comes from IP Australia. The authors are grateful to Benjamin H. Mitra-Kahn, sponsor of the project. Corresponding author: T'Mir Julius, Tel: +61 3 90358160; Email: tdjulius@unimelb.edu.au.

Melbourne Institute of Applied Economic and Social Research
The University of Melbourne
Victoria 3010 Australia
***Telephone* (03) 8344 2100**
***Fax* (03) 8344 2111**
***Email* melb-inst@unimelb.edu.au**
***WWW Address* http://www.melbourneinstitute.com**

Abstract

This document describes the methodology developed by the Melbourne Institute to:

(i) harmonise holders of intellectual property rights (IPRs) at IP Australia (applications for patent, designs, trademarks and plant breeder's rights);

(ii) match Australian IPRs holders to the Australian business register;

(iii) identify the ultimate owners within Australia; and

(iv) identify which holders are small and medium size enterprises.

JEL classification: O34

Keywords: Patent, trademark, design right, plant breeder's right, harmonizing, name cleaning, Patstat

1. Introduction

The code described in this document has been written for, tested on, and optimised for '*IPRIA data-Patents-all applicants.txt', 'IPRIA data-TM-all applicants.txt'*, and *'IPRIA data-Design.csv'* with emphasise placed on the patent data due to the availability of test datasets (in particular the OECD test dataset). The data in *'IPRIA data-PBR.csv'* were also included, though these data required somewhat more manual handling through the cleaning and entity identification stage. The raw data (csv and txt files) were provided directly to the Melbourne Institute by IP Australia (IPA). This dataset for this release has the following characteristics:

Table 1. Description of provided dataset

Type	Number
IPA Patent Applications	509,290
IPA TM Applications	1,032,560
IPA Design Applications	148,244
IPA Plant Breeder's Right Applications	7,651
IPA Applications (Total)	1,697,745
IPA Applicants (Total, unharmonised)	1,807,693
Entities (Total, unharmonised, cleanmysql.py)	1,465,014
Australian (Total, unharmonised, cleanaddresses.py)	888,189
Australian Entities (ABN/ACN Candidate)	605,951

The Melbourne Institute IP Australia Dataset (henceforth `ipa_applt`) contains the following information for all applicants. Note that all null entries have been converted to empty strings, and bits have been converted to their integer counterparts in the provided .csv's for ease of use:

- `ipa_applt_id` – The applicant identifier. An automatically incremented number corresponding to the order in which an applicant's information was inserted into the database. This number does not mean anything, but serves as a unique identifier for an applicant associated with an application;
- `ipa_appln_nr` – The application number (10 digit for patents, 9 digits for design and 5 or 6 digits for trademarks and 7 digits with a slash after the first 4 for PBR trademarks);
- `appln_type` – The application type. 'PAT ' for patents, 'TM' for trademarks, 'DES' for design and 'PBR' for PBR applications;
- name – The applicant name as given in the IP Australia files;
- `clean_name` – The name that has been cleaned as described in Section 2;
- `address` – The applicant's address;
- `country` – The ISO 3166-1 alpha-2 code for the country[1];
- `australian` – A bit that is set to 1 if the address has been identified as Australian. Null otherwise;

[1] http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

- `entity` – A bit that is set to 1 if the address has been identified as an entity. Null otherwise;
- `ipa_id` – An identifier of the group that an applicant belongs to. All applicants with the same identifier are believed to be the same applicant;
- `abn` – The applicant's Australian Business Number (ABN) if one was found;
- `acn` – The applicant's Australian Company Number (ACN) if one was found;
- `source` – The organisation from which the ABN or ACN was sourced;
- `big` – The small and medium enterprise (SME) status of an entity as described in the SME Status section;
- `ultimate` – The ACN of the ultimate owner;
- `lon` – The longitudinal result of the geocode results;
- `lat` – The latitudinal result of the geocode results;
- `qg` – An integer noting the resolution of the geocode results as described in Section 3; and
- `patstat_id` – The Patstat identifier for the application where one was found.

2. Cleaning and Harmonising

This section provides an overview of the python scripts (.py) used to clean and harmonise applicant names. The scripts were inspired by the methodology proposed by the OECD to build the harmonised applicant names (HAN) data table for Patstat.[2] The dictionary of legal company forms and their abbreviations was then extended using a list of international entity types online.[3,4] The dictionary was converted in to a series of regexes replacements (`makecleanmysql.py`). The names and addresses were cleaned of any non-English and non-ascii characters (by converting ö to oe, í to i, etc.) as well as any unmatched inverted commas (`nonascii_char_mysql.py`).

a. `cleanmysql.py` :

   i. Pulls `ipa_applt.ipa_applt_id` and `ipa_applt.name` from the database, applies regex to correct legal labels to a single, harmonised form, and often removes them entirely.

   ii. Splits names that contain terms such as '*AS RERPESENTED BY*' and keeps only the part preceding this, and terms such as '*ALSO TRADING AS*' and keeps only the following part. See Appendices 1 and 2 for the full dictionary of string delimiters.

   iii. Finds names that contain 'ABN' followed by 11 digits, or either 'ARBN' or 'ACN' followed by 9 digits (as well as some combination of the above) and removes these, but only if a sufficient portion of the name remains after.

   iv. Identifies entities by checking for strings that were:
      1. Affected by harmonisation;
      2. 1-2 words ending in a comma (3 words if one of them is a nobiliary particle such as 'VAN' or 'DI') followed by 1–3 more words with no commas, and none of the words are contained in a dictionary of common business words that are not affected by the harmonisation.

   v. Removes non alphanumeric characters.

   vi. Puts the cleaned name into the `ipa_applt_clean.clean_name` field, and sets the `entity` bit to a 1 or 0

[2] Magerman et. al., (2006)
[3] The dictionary has 1,928 entries, and is available upon request.
[4] http://en.wikipedia.org/wiki/Types_of_business_entity

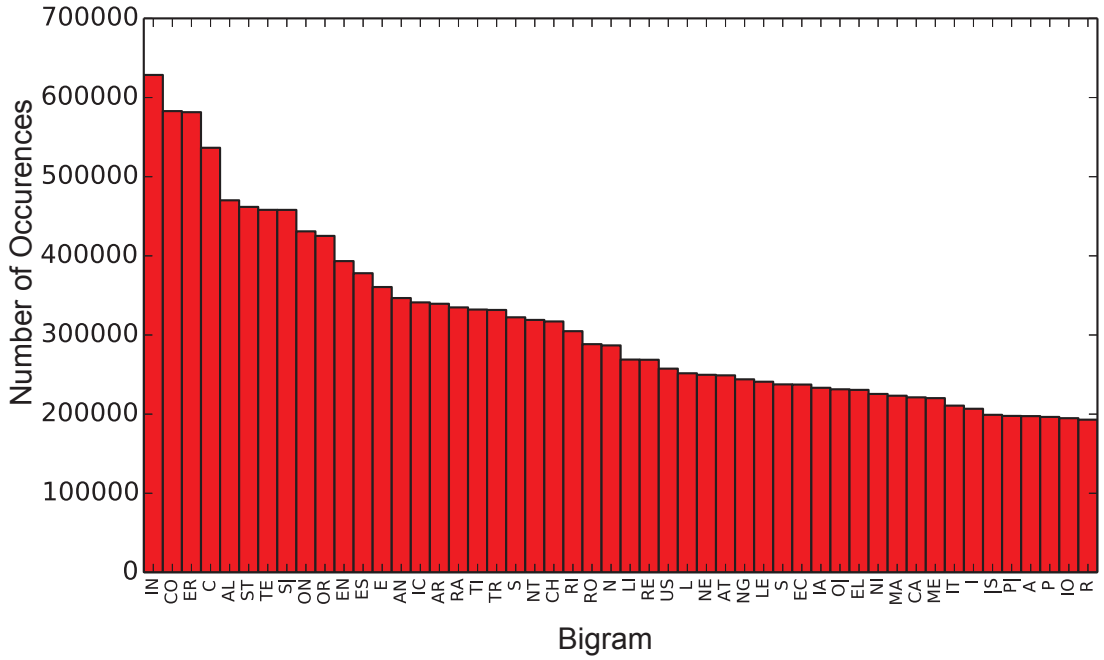
b. `makedictmysql.py` – Creates a dictionary based on bigrams and frequencies. A histogram showing the 50 most common bigrams for the entity applicants is shown in Figure 1.

c. `vectorisemysql.py` – Creates vectors and writes these in a list of tuples to the database.

d. `cosine_distance_fast_multi-mysql.py` – Calculates the weighted cosine distance of given words and pairs the ones remaining within a given tolerance (1.0 approximates a simple string match).[5,6,7]

e. `regroup_mysql.py` – Examines the applicants grouped together in the last step, and splits them up based on the extracted country information. This step was originally slated to involve matching the IPC subclasses of patent applications as well, but this has been sidelined for reasons discussed in Section 4.

Figure 1. Frequency of the most common 50 bigrams for the application data



The entries in the PBR file were provided with applicants in a very different form from that of the other applications. Due to the small amount and unique nature of the applications, it was deemed faster and more efficient to assign entity statuses manually. As a result, the entries from the PBR database needed to be updated with the manually assigned flags between steps a. and b. (`update_pbr_mysql.py`).

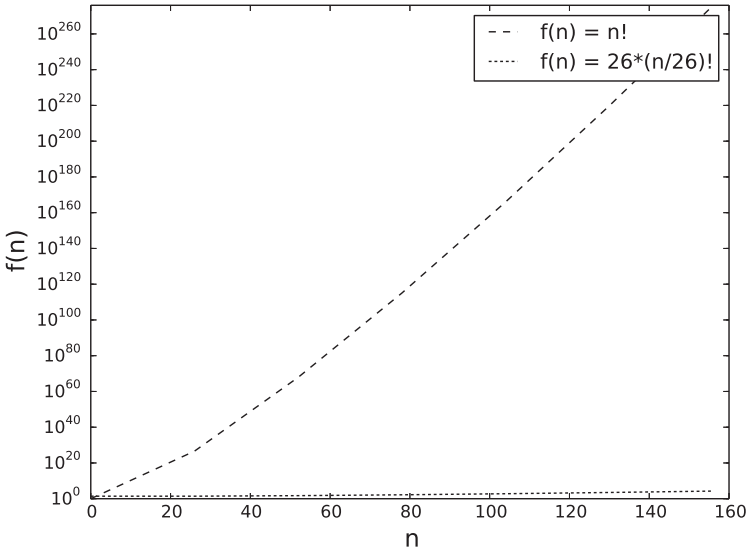
[5] See Raffo et. al. (2009)

[6] Names were broken up into a set of 2 character bigrams, with each weight calculated as $\frac{1}{\log n_{occurences}+1}$ where $n_{occurences}$ is the number of times a given bigram appears within the dataset.

[7] The cosine distance of 2 vectors, v_1 and v_2 is $\frac{v_1 \cdot v_2}{|v_1||v_2|}$

Each entry in the applicant list needs to be compared with every other entry of the applicant list, leading to a processing time that scaled like *n!* with the number of list entries (*n*). Originally *n* was the number of entity applicants (47,695 for the benchmark set and 1,465,014 for the final data). *n* has been reduced to be the number of unique cleaned names belonging to entities (18,457 for the benchmark dataset, and 330,203 for the final data). In order to further reduce processing time, the algorithm requires matching applicants to begin with the same letter. This reduced the processing to something more like *O(26*(n/26)!)*, which is much faster (around 48 hours for the full dataset on a system with the specifications described in Appendix 3). A plot of *O(n!)* and *O(26*(n/26)!)* can be seen in Figure 2. This shortcut is equivalent to increasing the weighting of the first bigram, which is justifiable as errors are more likely to be noticed and therefore corrected in the first letter of a name.[8] Additionally, this has allowed parallelisation of the project, bringing the processing of the entire list to be around 12 hours.

Figure 2. Graphical illustration of O(n!) and (26*(n/26)!)



[8] See Veronis (1988).

3. Geocoding and Address Processing

In order to identify which patents are owned by applicants that are operating out of Australia, it is required to identify which applicants are in fact Australian through the examination of their address. Additionally, geocoded addresses will be very useful in improving the accuracy of harmonisation as well as in ABN matching in future releases developed in conjunction with access to the Australian Business Registry.

Address cleaning and geocoding take place in the following way:

a. `formatcountrycodes.py` – A dictionary is created using 2 character country codes and their matching country names. This dictionary is extended to include variations on the name (e.g., United States of America, The United States) common typos of the name (e.g., Untied States of America, United Stats of America) and abbreviations (e.g., USA, U.S.A) as well as common places in that country (Washington DC, California) in the case that the applicant failed in included the country name.

b. `cleanadressesmysql.py` – Using the dictionary created in the previous step, addresses are examined word-by-word starting from the end and working backwards to see whether they include any country specific dictionary words. When a country is identified, the `ipa_applt_clean.country` field is set to the two character country code. In the case of no country being found, this code is set to *'ZZ'*. If the country is Australia, the `ipa_applt_clean.australian` flag is set to 1. In a final effort, any addresses without an identified country in which the last word is a 4 digit number are marked as Australian under the assumption that they end in a 4 digit postcode. Using this method over 99% of applicants have had countries assigned to them. Of the 11,254 addresses that did not have countries recovered, 10,658 were blank.

c. `auaddressparsermysql.py` – Cleans Australian addresses and puts them in to a geocodable format. Addresses marked as Australian in the previous step are extracted from the data.

d. `getgeocodelistmysql.py` – Addresses are separated in to useful addresses worth geocoding (full street addresses) and non-useful addresses (addresses that are only given to city or state level). A simple string match is applied so that work is not duplicated.

e. Remaining addresses are geocoded using:
   http://www.aus-emaps.com/bulk_geocoder.php
   Example input:
   251 OXFORD ST, BONDI JUNCTION, NSW
   Example output:
   1,"251 Oxford Street, Bondi Junction NSW 2022, Australia",8,-33.891642,151.252472
   (number in list, street address, city, state post code, country, accuracy of match, lon, lat)

The accuracy of match is a measure of how close the address was matched: 9: Premise or building; 8: Address; 7: Intersection; and 6: Street. Less than a 6 is useless (5 is the postcode) and even a 6 is a bit dubious. Useful addresses and their quality score are added to the database.

The PBR database needed to be treated differently as all applicants were included on one line with a single address. This address was always an Australian address, but there was a separate country code column that was often from overseas. The country code as shown in the .csv provided has been used rather than found using the script above. However, the Australian addresses have still had geocoding attempted and some of the entities have had ABN and ACNs attached, despite having a non-Australian country. The few with country codes that were *'TBA'* in the PBR data have been set to *'ZZ'*.

Results:

A total of 779,688 Australian addresses (288,830 unique) were geocoded successfully.

4. Benchmarking and Tolerance Decisions

The test dataset used is a selection of patents from Patstat which have had the OECD harmonisation applied and have the application authority as Australia.[9] The Australian applications were chosen as they were expected to be more likely to exhibit the properties of the data to be fitted. The set contains 42,560 patent applications with a total of 49,260 applicants. A total of 47,695 of these were identified to be entities. These data were output in to a database mirroring the actual database.

Code Fragment 1. MySQL Select statement used to extract the benchmark dataset from Patstat with the HAN database

```
SELECT c.person_name, d.han_id FROM tls201_appln a
        JOIN
  tls207_pers_appln b ON b.appln_id = a.appln_id
               AND a.appln_auth = 'AU'
        JOIN
  tls206_person c ON c.person_id = b.person_id
        JOIN
 ext_oecd_han_person d ON c.person_id = d.Oct12_Person_id
```

As the harmonisation of this analysis had the luxury of being focussed on applicants predominantly based in Australia and other English speaking regions, the cleaning code was expanded to take in to account phrases such as *'as represented by'* and *'also trading as'* as well as *'c/-'* and *'care of'*. By taking only the part of the name that occurs before '*as represented by*' and similar such phrases, government entries such as *"THE COMMONWEALTH OF AUSTRALIA AS REPRESENTED BY THE DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION OF THE DEPARTMENT OF DEFENC*E", *"THE COMMONWEALTH OF AUSTRALIA"* and *"THE COMMONWEALTH OF AUSTRALIA CARE OF THE SECRETAR"* and many other variations that were assigned to 18 separate han_ids to be harmonised in to a single ipa_id. As a result of these extensions, the benchmark dataset was found to be no longer adequate to suit the purposes of this study without some manual adjustments. The dataset was corrected to consider the han_ids consisting of variations of commonly occurring variations as matches. A table of the manually corrected han_ids can be seen in Appendix 4.

To keep the results in line with the literature, the ***precision*** *(n(true pos)/((n true pos + n false pos))* and the ***recall*** *(n true pos/(n true pos + n false neg))* have been calculated, in which *n true pos* is the number of matches that were identified in this study and also identified in the HAN data, *n false pos* is the matches that were identified here that but were not identified in the data, and *n false neg* is the number of matches that were not identified by this code, but were identified in the HAN data.

Table 2 describes the results of the benchmark dataset, and Figure 3 shows the recall vs. precision for various tolerances of the cosine distances. Ideally one wants to find a point that maximises the recall without sacrificing too much precision.

[9] The European Patent Office (EPO) Worldwide Patent Statistical Database (Patstat) offers bibliographic patent data for more than 100 patent offices. For more details, see de Rassenfosse et al. (2014)

Figure 3. Precision v. Recall for cosine distance tolerances between 0.85 and 1.0

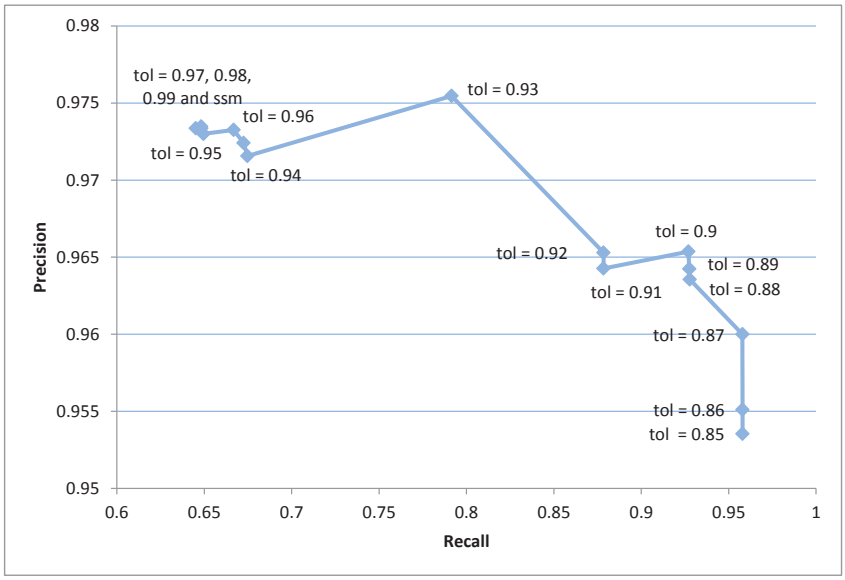


Table 2. Results of fast cosine distance on the benchmark dataset

Tolerance	Precision	Recall	n false neg	n false pos	n true pos
1.0 (SSM)	0.973360	0.645162	1,226,876	61,053	2,230,686
0.99	0.973478	0.648209	1,216,340	61,061	2,241,222
0.98	0.973363	0.648402	1,215,671	61,352	2,241,891
0.97	0.973004	0.649593	1,211,555	62,316	2,246,007
0.96	0.973257	0.666783	1,152,119	63,349	2,305,443
0.95	0.972406	0.672510	1,132,316	65,983	2,325,246
0.94	0.971562	0.674710	1,124,712	68,284	2,332,850
0.93	0.975464	0.791455	721,058	68,832	2,736,504
0.92	0.965293	0.878391	420,472	109,197	3,037,090
0.91	0.964264	0.878414	420,392	112,560	3,037,170
0.90	0.965364	0.926918	252,684	114,987	3,204,878
0.89	0.964235	0.927446	250,860	118,941	3,206,702
0.88	0.963559	0.927793	249,659	121,321	3,207,903
0.87	0.959998	0.957864	145,688	138,001	3,311,874
0.86	0.955104	0.957928	145,466	155,689	3,312,096
0.85	0.953534	0.957945	145,407	161,401	3,312,155

Based on these results, an initial tolerance of 0.90 was chosen. This is expected to give reasonable precision and recall. The use of the secondary information of the country in which an applicant was located was used to separate groups on geographical bounds.

The use of IPC codes was also considered for patent data, but it was found that this method primarily separated research institutions with many discrete departments, such as universities with clearly defined faculties. Address information has not been examined for effectiveness due to a lack of a suitable benchmark dataset. The ABN/ACN of an applicant was also considered, however it is

thought that this method will be more appropriate in future releases when the ABN information is improved through access to the ABR.

Results:

Of the 1,465,014 applicants identified as entities 372,192 had unique names before cleaning, 330,203 after cleaning, 321,790 after matching, and 330,950 were considered unique after separating based on country of origin.

5. Matching ABN's and ACN's

The ABN and ACN information was collected from two sources. The ASIC Australian Corporate Update DVD's each contain a Microsoft Access Database (.mdb) with a small package to assist users in utilising the data on a company-by-company basis.[10] The past 12 years' worth of .mdb's were converted into .csv's and loaded in to the `acn_abn_name` table. These DVD's contained the ACN, name, current name if the name has changed, registration status, registration date or deregistration date (whichever was most recent) and ABN if one has been registered with ASIC. The Department of Industry hosts ABN Lookup, which provides the ABN, ACN, name, past name(s), trading name(s), registration date and company type among other data for companies registered with the Australian Business Registry.[11] The table was designed to hold the following information collected from either source:

- `id_abn_acn_name` – An auto-incremented primary key;
- `name` – The name of the company (stored in upper case);
- `clean_name` – The name of the company with some cleaning to improve matching (removal of legal form, the word 'The', punctuation);
- `name_start` – Start date for that name;
- `name_end` – The date that name stopped being used;
- `acn` – Associated ACN if there is one, null otherwise;
- `abn` – Associated ABN if there is one, null otherwise (ABN and ACN should never both be null);
- `source` – The source the information was collected from (ASIC for ASIC DVD's or ABNL for ABN Lookup);
- `type` – If collected from ABN Lookup, the 3 character entity type. ASIC companies will either be a PRV (private company) or PUB (public company);
- `nametype` – Trading, or Main; and
- `big` – An integer describing the SME status as guessed, and the source on which that guess is based.

A unique key was imposed on the combination of `abn`, `name`, `name_start` and `nametype`. To build the database, the following steps were taken:

a. Initially ASIC ACN DVD's were converted from an .mdb to a .csv using MS Access.

b. .csv's were loaded into the ipa_2014 database – `loadacnmysql.py`

c. Applicant names and addresses were selected where `entity` = 1 and `australian` = 1.

   i. The name and address were examined to check for the phrases *'ACN'* and/or *'ARBN'* followed by 9 digits, or *'ABN'* followed by either 9 or 11 digits (there seems to be some confusion amongst applicants as to what constitutes and ABN or ACN). If a

[10] http://www.asic.gov.au/asic/asic.nsf/byheadline/Australian+Corporate+Update
(The first link, not the *'Business Names'*)
[11] http://www.abn.business.gov.au/

valid ABN or ACN (one that is present in the `abn_acn_name` table) is found, this is taken to be the ABN/ACN.

ii. If not, names were queried using `MySQLdb` library in Python.

iii. If no match was found, the `clean_name` was queried. The `clean_name` has the legal form and punctuation removed.

iv. If this also had no match, the name was checked for phrases such as *'REPRESENTED BY '*, *'ALSO TRADING AS'*, *'TRUSTEE FOR'* and the like (see Appendices 1 and 2).

v. Steps iii. and iv. were combined such that the result of step iv. was cleaned as a final match attempt.

d. In the case of multiple matches, the ACN/ABN selected was the first one found that had a beginning date before and an end date after the filing date – `acnmysql.py`. (This matching algorithm will be greatly improved when the ABR database with addresses and geocoding information is made available.)

e. In the case of no match, the name was queried using ABN lookup – `abnlookup.py`

f. All matches with ABN match score of 100 were queried, and the historical information was collected – `gethistory.py`

g. Each collected ABN was added to the `abn_acn_name` table – `loadhistorymysql.py`

Results:

Matches have been found for 85% of unique Australian entity names, and 90% of entities. There are 19,173,508 entries in the `abn_acn_name` table covering 3,523,654 unique ABN's and ACN's.

6. Shareholder Information and Ultimate Owner Extraction

The aim of this part of the project was to identify the top level owner with an ACN for each applicant, as well as a very simplistic percentage owned.

The steps involved in were:

- Creating a table containing all share holdings in effect on a given date;
- For each share holding entry creating an owned-owner pair of ACN's; and
- Finding any ACN that appears in both the 'owned' and 'owner' column and any entry with that ACN as owner with its parent companies as owners.

The shareholder information has been loaded in to the `share_history` table ipa_2014 database. A unique key is required due to the fact that many overlapping datasets were used to generate the final share history. The unique key has been created based on:

- the owner;
- the ownee;
- number of shares; and
- start date.

A graphic worked example of the algorithm used to simplify the shareholder information is shown in Figure 4. In the figure, the following steps are taking place:

- In the first column, company A and company Y appear in both the owner and ownee column. The information for company Y is simplified first such that any company Y owns is transferred to its owner (in this case, C).

- In the second column, company A is owned by 2 companies, C and B. For the company A owns (X) a new entry is created for each of its owner (C and B). However, since there is already an entry for X is owned by B, these two entries are merged.

Figure 4. Visual representation of the process used to simplify the shareholder information to a usable form

X owned by A X owned by B Y owned by C A owned by Y A owned by B	X owned by A X owned by B Y owned by C A owned by C A owned by B	X owned by C X owned by B Y owned by C A owned by C A owned by B

One concern brought at the beginning of this exercise was the presence of a circular ownership structure, in which two entities had shares in each other; however this scenario did not come up with this dataset. Due to resource constraints, any entries in which the shares held were less than 1 percent of the total shares for that entity were dropped.

This information was created by loading only entries that were in effect on 1/1/2006 in to the `share_history_2006` table and then running the algorithm over these shares. The total number of shares that are available from a company was calculated as the sum of the shares in effect on 1/1/2006. This date was chosen as the most recent share data available was for 2006.

Due to a reduction in the number of IT staff at ASIC, the data team has not been able to provide us with an update of ownership data. However, once those data are made available, it will be straightforward to use them using the tools developed.

The owner with the largest percentage of shares in a company was considered to be the ultimate owner of an ACN.

7. SME Status

There are two primary sources for company size information implemented in the test dataset: the Bureau van Dijk (BvD) and the Australian Business Database. The small and medium enterprises (SME) codes given indicate the information in the following way:

BvD:

- The subsidiary-parent list was acquired from Wharton/BvD;
- ABN's and ACN's were found for Australian companies in the list where possible using same methods as applicant ABN matching;
- The `big` variable was set to 1 for these 10,032 unique ABN's and ACN's, equating to 49,377 different ABN/ACN/name sets; and
- `big` was set to 11 for the 2,717 unique daughters of these companies (16,358 ABN/ACN/name sets), identified using the 1/1/2006 share data where `big` was null.

Australian Business Database:[12]

- Companies that had identified themselves as having > 200 employees were pulled from the list;
- ABN's and ACN's were found where possible;
- The `big` variable was set to 2 for the 470 unique ABN's and ACN's (2,200 ABN/ACN/name sets) where big was not already 1 or 11; and
- `big` was set to 22 for the daughter identified using the 1/1/2006 share data where `big` was not already 1, 11 or 2.

Manual Checks:
It may be worthwhile to manually check for entities with a large number of applications with size information missing, and setting:

- 4 for big; and
- 0 for small.

At this point there are about 100 Australian entities that have been matched to ABN's with more than 100 applications, but without an SME status. There are over 200 ABN's with more than 80 applications. At a rate of 5 minutes per company it would take around 8 hours to find size information on 100 companies.

Results:

Using the two datasets mentioned, 112,668 Australian applicant entities (6,075 unique ACNs) were identified as being large.

[12] Details of ADB can be found online at www.screechmedia.com

8. Extracting Patstat Application IDs

The Patstat identifiers available in the final dataset were found in the following way:

- To begin, the Patstat application number was assumed to map directly to the IP Australian Application number. The Patstat database was queried for the IPA application number with the application authority as 'AU'.

- If this found no hits, the alternative, short form Patstat ID, was tried. In this case the Patstat application number is expected to be of the form xxxxxyy corresponding to an IPA number of yyyy0xxxxx where yy is the 2 digit year. However, this date relates to the filing date in Patstat, which can occasionally differ from the filing date provided in the data from IP Australia at the commencement of this project. As a result, if no hit was found for the second attempt, a third and fourth attempt would occur by varying the year by +/- 1.

- In the case of multiple hits, the Patstat ID used was the one that corresponded to the application with a kind of *'A'*. In the case that no application had that kind value, the first hit was taken.

- Once the Patstat ID's had been collected and stored in the `ipa_patstat_2012` table, a second table, `ipa_patstat_ipc` was filled by extracting the IPC codes for a given Patstat ID. The IPC codes for each IPA application were collected in to `ipa_patent_supp` from the Auspat DVD data, and applications that had at least one matching IPC code were considered to match. This served as a check that correct the Patstat ID was paired with the Auspat number, particularly in instances where the year was varied by +/- 1. Approximately 97% of matched Patstat identifiers were found to be correct using this method

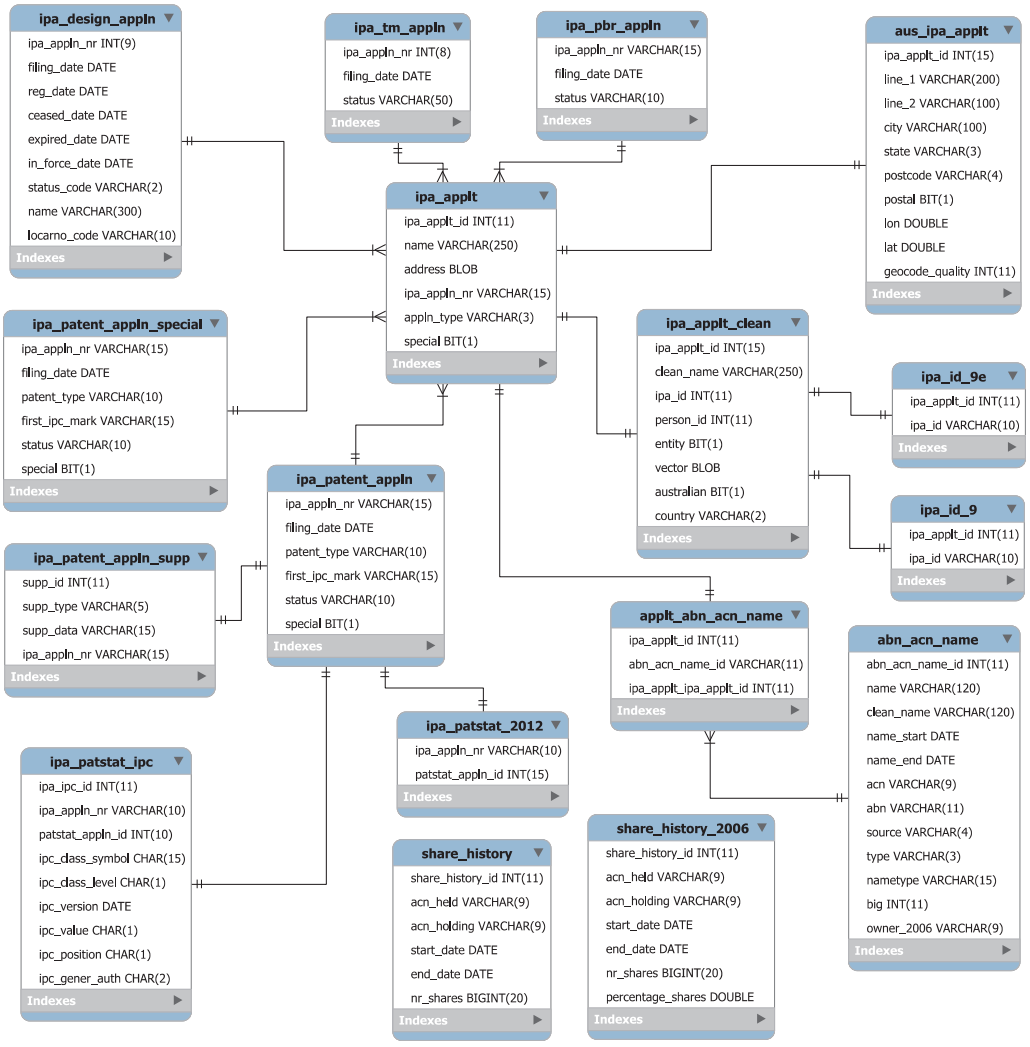
Results:

Patstat ID's were found for 438,120 IPA patent applications. However, it has since been found that the IPA Patent application number can sometimes be the same as the WIPO publication number of an application. This number is available on the Auspat DVD, and will be extracted for future releases of the dataset.

9. Database Design

The database for the storage and access of the data has been inspired by the design of the European Patent Office's Patstat database. It is hoped that this will enhance usability and decrease the learning curve required of users, as some will have already been exposed to the EPO database.

Figure 5. An enhanced Entity-Relationship diagram of the database designed for this project



The four types of applications have been separated in to separate tables, as each has different properties that would be difficult to store in a uniform way. Additionally, the `ipa_patent_special` table was created to store additional patents that were not included in the original data received from IP Australia, but were requested as a part of additional datasets. Applicants are assigned one of the types of *'PAT', 'TM', 'DES'* and *'PBR'* application types (`appln_type`) as described earlier, as well as the type *'SP'* which denotes applicants belonging to members of the `ipa_patent_special` table.

The `ipa_patent_appln` table has a one-to-one or zero relationship with the `ipa_patstat_2012` table. In here the `ipa_appln_nr` is paired with a `patstat_id`. It is envisioned that this will eventually be a one-to-one relationship.

The ipa_pastat_ipc and ipa_patent_appln_supp tables both have many-to-one relationships with the ipa_patent_appln table. These tables hold IPC codes from Patstat and from Auspat respectively for applications. These tables were used to perform quality checks on the matched patst_id's. It is planned that the ipa_patent_appln_supp table can be expanded to hold different kinds of supplementary information for patent applications, such as prior art information and linked provisional application numbers.

The ipa_applicant table has a one-to-one relationship with the ipa_applicant_clean table. This is to create a clear separation between the raw data and any processing of the applicant information that takes place. Applicants have had an auto-incremented ipa_applt_id generated for ease of use. The applicant table has a one-to-one or zero relationship with the aus_ipa_applt table (which holds the cleaned addresses for Australian identified applicants) and the applt_abn_acn_name table. This table connects the applicants identified as Australian entities that have been linked to ABNs and ACNs with the entry that they were matched to in the abn_acn_name table (see section 5 by creating a row that links the abn_acn_name_id and the ipa_applt_id.

The ipa_applt_clean table has a one-to-one or zero relationship with the ipa_id_9 and ipa_id_9e tables. These tables hold the result of the two harmonisation processes, and have entries that assign each ipa_applt_id to a harmonisation group. ipa_id_9 holds the results of the initial harmonisation, in which entities were matched on the basis of their name alone. The 9 in the name denotes the tolerance of the harmonisation. ipa_id_9e holds the results of the extra harmonisation, in which harmonisation groups from ipa_id_9 were separated based on the address country.

Finally, the share_history and share_history_2006 tables all store data relating to ownership structures. share_history stores the raw data from ASIC as of 1/1/2006. share_history_2006 stores the share history after it has been simplified to reflect ultimate share owners, as well as some additional shareholder - subsidiary information derived from Bureau van Dijk.

10. Further Work

The database can be further improved in the future. Besides correcting errors that could be spotted once the database will be used, possible extensions include:

- Improving individual and entity separation before cleaning to avoid harmonisation of words appearing in names.
- Creating targeted legal forms to be harmonised based on applicant country.
- Extending the various dictionaries (e.g., trading as, care of, common business words).
- Improving ABN matching through the use of the ABR address and geocoding information.
- Creating a benchmark dataset manually in order to more accurately estimate the precision and recall of harmonisation, as well as the accuracy of ABN/ACN matching.
- Incorporating the ABN and geocoding information into harmonisation.
- Presenting ultimate owners as of the most recent share data.
- Expanding the database to include information on innovation patents and provisional patent applications as well as secondary patent information such as WIPO and PCT numbers.
- Extracting trademark information directly from the TM database to improve truncated data.

11. References

de Rassenfosse, Gaetan, Helene Dernis, and Geert Boedt. “An introduction to the Patstat database with example queries”, *Australian Economic Review*, (2014): forthcoming.

Magerman, Tom, Bart Van Looy, and Xiaoyan Song. “Data production methods for harmonized patent statistics: Patentee name harmonization.” DTEW-MSI_0605 (2006): 1-88.

Raffo, Julio, and Stéphane Lhuillery. “How to play the ‘Names Game’: Patent retrieval comparing different heuristics.” *Research Policy* 38.10 (2009): 1617-1627.

Veronis, Jean. “Computerized correction of phonographic errors.” *Computers and the Humanities* 22.1 (1988): 43-56.

12. Appendices

Appendix 1 String delimiters that signify the end of the text of interest

'AS REPRESENTED BY ', 'REPRESENTED BY ', 'CARE OF ', 'C-/ ', 'C/- ', 'C/O ', 'C- ', 'C/ '

Appendix 2 String delimiters that signify the beginning of the text of interest

'AS TRUSTEES FOR ', 'AS TRUSTEES OF ', 'AS THE TRUSTEES FOR ', 'AS THE TRUSTEES OF ', 'ALSO TRADING AS ', 'TRADING AS ', 'T/A ', 'AS TRUSTEE FOR ', 'AS TRUSTEE OF ', 'AS THE TRUSTEE FOR ', 'AS THE TRUSTEE OF ', 'AS A TRUSTEE FOR ', 'AS A TRUSTEE OF ', 'ATF '

Appendix 3 Specifications of the desktop used during this study

Table 3. Desktop specifications

CPU	Intel Xeon E5620 2.40GHz
Memory	8GB
Storage	1 TB Hitachi HDS72010
Operating System	Windows 7 Enterprise SP 1
SQL Server	MySQL Server 5.6.15
Python	Python 2.7.6 MySQLdb 1.2.4b4 numpy 1.8.0 matplotlib 1.3.1

Appendix 4 Corrected HAN ID's

Table 4. HAN ID's that were combined in to one ID for the purposes of this study

UNITED STATES OF AMERICA		UNITED STATES GOVERNMENT	GOVERNMENT OF UNITED STATES		COMMONWEALTH SCIENTIFIC INDUSTRIAL RESEARCH ORGANISATION	COMMONWEALTH OF AUSTRALIA
2406351	3529940	3140428	3524795	3524810	491680	7995
3006842	3529953	3002891	3002891	3524815	7997	491560
3006843	3529956	3215026	3006731	3524859	7998	491588
3016344	3529957	3529209	3006732	3524867	7999	491592
3027949	3548998	3529220	3027949	3524871	311460	491609
3140428	3549014	3548901	3214810	3524876	490995	491610
3214953	3549020	3548905	3214848	3524882	491630	2395561
3524680	3549021	3548907	3214849	3524884	491638	2395570
3529154	3549039	3548917	3214851	3524896	491643	2395572
3529274	3549045	3548929	3214852	3524898	491657	2395573
3529275	3549050	3548931	3214858	3524908	491694	2395574
3529322	3549051	3548955	3214872	3524909	491696	2395576
3529327	3549072		3214873	3524910	491758	2395577
3529328	3549081		3214890	3524911	491760	2395583
3529356	3549082		3214891	3524927	491768	2395585
3529359	3549083		3214896	3524932	491779	2395609
3529382	3549084		3214899	3524935	491803	
3529399	3549089		3214909	3524964	491806	
3529400	3549097		3214912	3524975		
3529402	3549099		3214913	3524976		
3529446	3549103		3214916	3524982		
3529447	3549104		3214923	3524984		
3529474	3549116		3214924	3524999		
3529554	3549122		3214933	3525001		
3529555	3549127		3524657	3525009		
3529556	3549155		3524661	3525016		
3529580	3549156		3524677	3525020		
3529583	3549157		3524681	3525024		
3529595	3549164		3524690	3525028		
3529596	3549170		3524693	3525029		
3529622	3549171		3524694	3525033		
3529636	3549172		3524725	3525036		
3529658	3549179		3524729	3525043		
3529677	3549180		3524732	3525047		
3529697	3549187		3524734	3525097		
3529704	3549188		3524735	3525098		
3529706	3549189		3524740	3525100		
3529732	3549192		3524741	3525101		
3529881	3549195		3524749	3525109		
3529890	3549196		3524758	3525128		
3529896	3549203		3524760	3525130		
3529913	3549204		3524761	3525131		
3529914	3549207		3524763			
3529915	3549213		3524766			
3529938			3524786			
3529939			3524809			