Name Cleaning

Michael Cahana, Yixin Sun

January 2019

Contents

1 Introduction

We use ”firm” to refer to either the operator or lessee name we are performing the cleaning on. We use ”DI alias” to refer to the cleaned up version of the firm name that is in the relevant DI dataset. We use ”alias” to refer to the slightly cleaned up version of DI alias that results from Viraj’s regex work.

2 Data

The following datasets are relevant to these cleaning procedures:

- DI Landtrac Leases: 1:1 mapping of lessee names to lessee addresses
- DI Flatfiles
  - NPH_OPER_ADDR: maps operator ids to operator addresses. Note that an operator usually has more than one unique operator address.
  - pden_desc: maps operator ids to operator names
- ”company list.xlsx”: excel sheet with a column for regular expression patterns and a column for the standardized name. This excel sheet was created by the intern Viraj Kumar.

3 Procedure Outline

1. Use DI’s existing standardized firm names
2. Normalize punctuation and spacing, and apply prior name mappings in a regex call
3. Take list of unique firm names and compares every element in that list to every other element in that list using Cahana’s string comparison methods. This outputs a list of pairwise matches.
4. We clean a list of unique address names, (getting rid of commas, periods, P.O. etc). We then compare every element in the list of addresses to every other element. Here we only keep perfect matches.

5. Concatenate the two matched datasets. This outputs a dataset that has name, matches, method used for matching (name vs address).
6. For the remaining matches that have a decent chance of being correct (address does not match and fuzzy string matching inconclusive), check over these by hand

3.1 Pre-existing DI Cleaning

DI has already done some of the cleaning for lessee and operator names.

- landtrac leases: "alias grantee" is the standardized version of "grantee".
- pden_desc: "common oper name" is the standardized version of "reported oper name".

Use DI alias as the firm name for the subsequent cleaning steps.

3.2 Initial regex

Before cleaning names using a regex call, clean names for punctuation and spacing. Ensure that all characters, if apart, are only one space apart, remove parentheses and dots, make all names uppercase, replace ampersands with "AND", etc.

Once this basic normalization is applied to DI aliases, clean names further using "company list.xlsx", an excel sheet with 2 columns: the first column is a regular expression pattern, and the second column is the standardized name. An example of its data is shown below:

regex name	alias
FOUR C	FOUR C OIL AND GAS
CROSS TIMBERS	XTO
OXY	OCCIDENTAL
BURLINGTON	CONOCO PHILLIPS
ANADARCO	ANADARKO

For each firm in our dataset (be it the lessee dataset or the flatfile dataset), if the firm contains a "regex name", map the firm to the "alias" associated with that "regex name". That is, if we observe a firm in our DI Landtrac Leases dataset called "OXY OIL & GAS", regex will detect that "OXY" is contained in the firm name, and consequently map "OXY OIL & GAS" to the alias "OCCIDENTAL". The table below demonstrates how this regex join would work for the mock data described above:

firm	regex name	alias
FOUR C INC	FOUR C	FOUR C OIL AND GAS
CROSS TIMBERS PRODUCTION	CROSS TIMBERS	XTO
OXY OIL AND GAS	OXY	OCCIDENTAL
BURLINGTON O&G	BURLINGTON	CONOCO PHILLIPS
ANADARCO	ANADARCO	ANADARKO

The new alias column is a first pass at cleaning, and will be used to cut down on the number of potential matches we have to consider in subsequent steps. When DI alises are not matched via regex, we replace alias with the DI alias. Once we have a new set of firm names, we clean them once more by dropping out common words (such as LLC, OPERATING, COMPANY, etc.)

3.3 String Matching

Given a cleaned set of firm names, we apply three separate string matching algorithms to determine matches between names. All three algorithms loop through each firm name, and compare it to all other firm names, declaring two names to be a match if they satisfy a certain criterion. The three algorithms are as follows:

3.3.1 Shared Word

Transform firm names into bags of words, and declare one name to match another if their bags share at least one word. For example, "JAMES L MARSHALL" would be matched to "MARSHALL RICHARD R" due to the shared "MARSHALL". This algorithm is our most conservative, and will likely catch a lot of false matches. But it is also likely to catch most true matches.

3.3.2 Jaro Distance

Calculate the Jaro distance from every name to every other name using the stringdist package. Declare a match pair to be two names with a distance less than or equal to a pre-determined threshold (0.15 threshold suggested). This algorithm is useful for detecting typos. For example, it would determine that "SANDDRIDGE ENERGY INC." and "SANDRIDGE EXPLORATION AND PRODUCTION, LLC" are a match.

3.3.3 Cosine Similarity

Transform firm names into bag of word vectors using the text2vec package, and use the tf-idf method to weight words by relevance, down-weighting common words. Then compute the cosine similarity between every pair of vectors. Declare a match pair to be two vectors with a cosine similarity greater than or equal to a pre-determined threshold (0.7 threshold suggested). This algorithm captures the same types of matches as the shared word algorithm, except in a more elegant fashion.

With three sets of potential matches (one set for each method) in hand, the sets are combined together and duplicate matches are removed such that a single list of potential matches is outputted.

3.4 Address Matching

In parallel to the string matching, we also match firms through matching addresses. The processes differ whether we are matching operators versus lessees.

- Operators: DI has created an m:m mapping of operator ids to operator addresses, housed in NPH_OPER_ADDR. We first join this mapping onto pden_desc, mapping operator names to operator addresses
- Lessees: DI's landtrac lease dataset has a 1:1 mapping of lessee to lessee addresses

3.4.1 Manual Cleanup

In both cases, we first clean the list of the unique address names for punctuation and spacing. Ensure that all characters are only one space apart, remove commas and periods, make all names uppercase, normalize all variations of PO BOX, keep only the first five letters of a zip code, etc.

3.4.2 Google Address Geocoding

We further standardize the addresses using Google's geocoding service, which cleans up address names and provides coordinates for the address. We will need to clean around 250,000 addresses for the time being, requiring about $1000 worth of geocoding credits. The current pricing scheme is as follows:

- It costs $4 to geocode 1,000 addresses, but the first $200 worth are free (50,000 addresses) each month
- The 12 month free trial of Google Cloud Computing gives us an additional $300 worth of credits (75,000 addresses)
- If we successfully apply for an educational grant, we could receive $5,000 in credit.

We can access the Google maps API using the R package googleway, which will return a standardized version of the firm addresses.

We will be skipping PO boxes in the Google cleaning method for several reasons.

- PO box addresses are pretty easy to standardize
- PO box searches often fail in Google and return just the zipcode of the original address
- Successful PO box searches on Google maps usually returns the building the mailboxes are in, such as the mail center it is housed in. It is feasible that two PO boxes belong to two different companies, housed in the same building. We therefore want to avoid incorrectly matching these companies.

This will greatly cut down the number of addresses we have to geocode, so it is possible that we will stay within the amount allotted from the free trial.

Once the addresses are fully cleaned, we loop over each address in this cleaned list and compare it to all other addresses, keeping perfect address matches. Lessee names are declared a match if their cleaned addresses match. For operators, if any of their operator addresses are a match, we declare the operators names to be a match.

We save an address standardization dataset of original address names to cleaned address names to be used in subsequent cleanings.

3.5 Inconclusive Matches

Join the two matched datasets, producing a dataset with alias1, alias2, and match type (string match vs addresses). Check over the remaining inconclusive matches by hand (address does not match and string match just below the threshold). This includes manually looking for spelling errors, simplifications, and searching for subsidiary/shell company information.

This hand checked dataset is joined with the previous two datasets to produce a final list of matches between aliases.

3.6 Graph Theory Magic

We want to group matches that share an alias. For example, if Chesapeake is matched to Chesapeake Marcellus, and Chesapeake Marcellus is matched to Chesapeake Utica, we want to map all three to Chesapeake. If we construct a graph with the nodes Chesapeake Marcellus, Chesapeake Utica, and Chesapeake, and edges in between Chesapeake/Chesapeake Marcellus and Chesapeake Marcellus/Chesapeake Utica, all three names are said to form a connected component. R's igraph package finds all connected components, and our code then assigns every operator within a component the name of the operator that is first in alphabetical order.

The final ”firm standardization” dataset will have a column for firm, a column for the DI alias, and a column for group name that can be merged back to the lease/production dataset.

4 Subsequent Iterations

If/when DI data is updated, or if we want to clean other DI datasets, we can re-run the procedure outlined above. First, join the dataset to the firm standardization dataset by the DI alias. We now use the standardized group name to cut down the number of unique firm names to be cleaned. Run this list of unique standardized names through the procedure listed above (regex, string matching, address matching, hand cleaning, and graph theory grouping).

Note that in subsequent address matching routines, we can use the existing address standardization dataset to cut down on the number of unique addresses to clean.

This new firm standardization map should be concatenated with the old firm standardization map to create a new master dataset.