The Keymaker, Opening the Door to Energy Data Possibilities

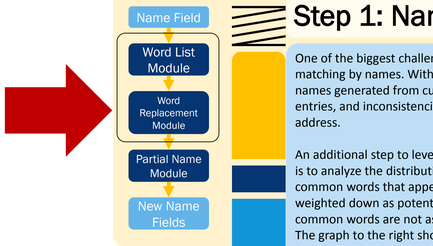
Kai Zhou, Managing Consultant, Data Science, Opinion Dynamics

Introduction

The ability to flexibly create, combine, or subset any dataset relies on a unique identifier; otherwise known as a unique key or primary key. In many Energy Efficiency Evaluations, evaluators work with utility customer and program tracking data. The utilities or program implementer typically set up these datasets such that they can be linked using account numbers. However, when evaluators report results or draw samples, they often do so in terms of sites, households or business facilities. This results in a disconnect in the unit of analysis because account level information does not perfectly capture households or facilities. This is especially true in commercial or multifamily facilities that often contain more than one utility account.

Methodology Overview

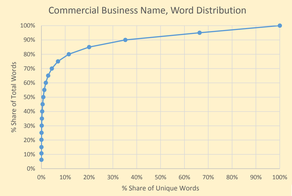
This poster lays out a framework for creating unique keys at the site level from utility customer data, leveraging tools such as regular expressions, address unit parsing, geocoding, word weighting, logical criteria grouping and sorting. This method is very fast because it minimizes comparisons between records to only those that are likely to match. The end product is a key mapping that shows how every original record in the starting data corresponds to the generated unique key.

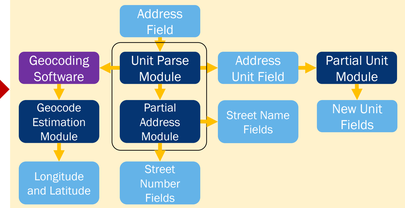


Step 1: Name De-aggregation

One of the biggest challenge for generating a key involves matching by names. With many tracking systems that contains names generated from customer inputs, misspellings, shorthand entries, and inconsistencies are common issues we have to address.

An additional step to leveraging the use of partial text data fields is to analyze the distribution of most common words. The common words that appear in the data field would have to be weighted down as potential matches such that a match between common words are not as likely to produce wrongful matches. The graph to the right shows the % share of total words to the % share of unique words. This indicates that we can down-weight almost 80% of the text by just tweaking 10% of the total unique words in a name field (Pareto Distribution).





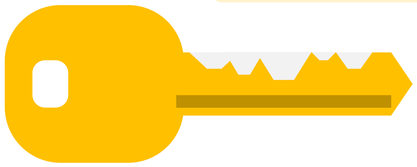
Step 2: Address De-aggregation

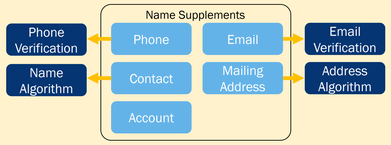
Cleaning the address fields in a way that produce the following types of output:

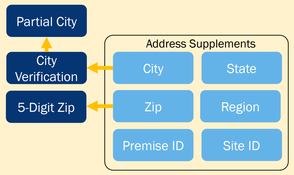
- Standard formatted addresses that can be used to obtain fast geocode matches
- Standard and partial addresses that can be used for matching purposes
- Potential unit fields that are used for identifying unique, or the same spaces in a building and also for identifying non-survey-able or retrofit-table sites.

Standardizing addresses requires a list of standard street suffixes, a list of unit number prefixes, and additional text matching rules that looks for the most ideal unit parsing location. In addition, it will also address abbreviation along with the non-abbreviated versions of the same text.

The geocoder can solve the issue of the same building with different doors on different street and having different addresses. This affects larger buildings where we want to determine, for example, usage for the whole building and not for a meter that accounts for half of it.







Step 3: Leverage Additional Data

Leveraging additional data can help in loosening some of the partial matching rules for name and address. For example, using phone numbers and contact names we can loosen our partial match criteria to use fewer letters. Because of definitional reasons, name supplements cannot be a substitute for name because we want to be able to differentiate between a homeowner or business owner vs a facility or property manager.

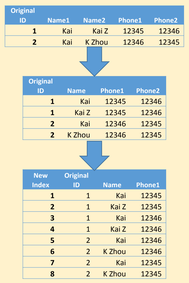
Because address names are reused across cities, it is essential to supplement addresses with city or zip when working with very large datasets. In addition, premise IDs and site IDs that are used in program tracking can often be used as substitutes for address since those variables are generally well defined within the same tracking database.

Note on Geocoding

Geocoding software such as ArcGIS or GeoLytics typically do not clean fields and only output partially geocoded results. To obtain estimation for geocodes for all of records, I employ the following:

- Determine street name by stripping out street numbers and street suffixes.
- Estimate geocode centroids using street name with a combination of city and/or zip code.
- Estimate city centroids, estimate zip centroids.
- Apply estimates to data without geocodes.

Having geocode estimates for matching can also resolve some more complex issues with address, city and zip misspellings and typos that 3rd party geocode software generally can fix.



Note on Multiple Versions of the Same Variable

One version of an imperfect data issue is when we get multiple versions of the same type variable, such as multiple versions of address or phone numbers. While it is feasible to run the algorithm separately for them and code in cross comparison between one and the other, it is much easier to create copies of the observations such that all variables of the same type are in 1 single data field. An example of this is presented in the figure to the left. As such, the original ID would be treated like a perfect match variable that will be able to connect together all of the duplication we introduced as part of this reshape.

Note on Missing Values within Variables

A separate version of an imperfect data issue is dealing with missing data and partially missing data. As a general rule, a missing and a non-missing comparison should always result in a non-match, except in the special case of address unit numbers. In cases where there are many supplemental variables and a lot of missing values in the primary name or address data field, a combination of the supplemental variables can be used depending on the situation.

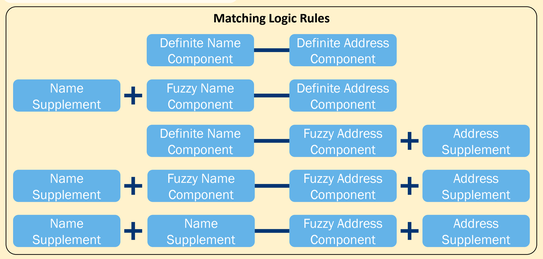
Step 4: The Big Loop

The main purpose of the big loop is to use a set of logical rules to make matches. It must address several things:

- Be fast and efficient.
- Avoid unnecessary comparisons.
- Be universally applicable.
- Export consistent results

As a general rule for the first two bullets for programming purposes, if one can't see potential matches by scrolling though a sorted data, it's very unlikely to exist. Therefore, the comparisons are limited to only records that are sorted right next to each other. For a match that is found, we change the unique id to the lowest unique id within the match group. Therefore, the resulting IDs are always going to converge to the lower end.

Applying Rules



Mapping File

At the end of the loop, the algorithm produces a mapping file that essentially links every record in the starting file to another ID that would be the same for each records that it thinks belong together.

Because the end result is a master mapping file, the initial strategy for implementing the algorithm is to create a complete list of all the names and address that we want to analyze. Such as customers from various different programs, list from various different utility companies, and list from internet databases such as info group and D&B.

The mapping file will then be applied individually to each of the datasets.

Applications

Applications for this technique go beyond rolling up utility account numbers to site level, it can also be applied to a wide range of tasks that used to be difficult or time consuming. For example:

- Finding the total kWh usage of a commercial facility with multiple accounts in order to obtain more accurate Energy Usage Intensity.
- Hedging against sampling bias by eliminating over-reliance on any one variable such as phone numbers to determine a sample frame.
- Matching data from third party data sources that might not have any common identifiers in order to obtain demographic or firmographic data for modeling in potential studies or market characterizations.
- Matching accounts from various different electric and gas utilities to generate household level population data from which to draw sample from or to analyze populations at a statewide level.
- Checking for data errors in existing customer and participant lists and help identify duplicative records, fix data entry errors, and fill data gaps with related records.

Checking Results

Obviously, the result of creating one unique key is prone to make some errors while fixing others. There are two types errors that may occur.

- Type 1 error: not matching when it should have
- Type 2 error: matching when it should not have

The best way to identify these things is to 1) look at unique copies of various old identifier within the newly generated key, and 2) looking at unique copies of the key within other variables.

In the end, the programmer just have to tweak rules and thresholds to find the right amount of error that one is willing to accept for the data.

Conclusion

With this reliable approach to creating unique identifiers, we open the door to new analyses and improved data connections while at the same time providing quality checks of the data. The most elegant part is that this method can be consolidated into a single customizable program or script and applied to millions of records. We have used this approach for over a dozen clients across the U.S. and Canada.