THE GLOBAL STOCKTAKE
CLIMATE DATATHON

PROMPT OWNER

Data Driven Enviro-Lab (DDL)

PROMPT TOPIC

ClimActor and entity harmonization of climate actors

PROMPT BACKGROUND

Entity matching in climate policy

With an increasing number of cities, regions, and businesses engaging in global climate action, there has also been a corresponding proliferation of data platforms and data sources reporting data on these climate actors. This explosion of data, both in terms of number of sources and number of actors, has led to major challenges when working across data sources in ensuring that any information collected is attributed to the right actor. For example, the country Cote d'Ivoire may be recorded as "Ivory Coast", "Côte d'Ivoire", or "Republic of Côte d'Ivoire" in different databases. A lack of standardization of the actor's name prevents the aggregation of data from different sources.

While entity matching is a well-studied area[1], little has been done to apply entity matching techniques to subnational (cities and regions) actors, much less for applications in climate data and policy. Yet, entity matching is crucial in making datasets interoperable for conducting the Global Stocktake of our collective progress towards achieving global climate goals. The Data-Driven EnviroLab has developed an R package (ClimActor) to help address this gap in entity matching solutions for analyzing climate action, but much more is needed to be done to improve on current processes.

ClimActor

The ClimActor package[2] (more explanation here) is an open-source R package developed by the Data-Driven EnviroLab (DDL) to ease the data cleaning process for working with climate data across different sources. The fundamental tenets of the package involve the use of fuzzy matching approaches based on phonetic representations of actors' names. The fuzzy matching allows for attribution of actors from an incoming data source against a referential database containing a compiled list of actors based on years of research conducted by DDL

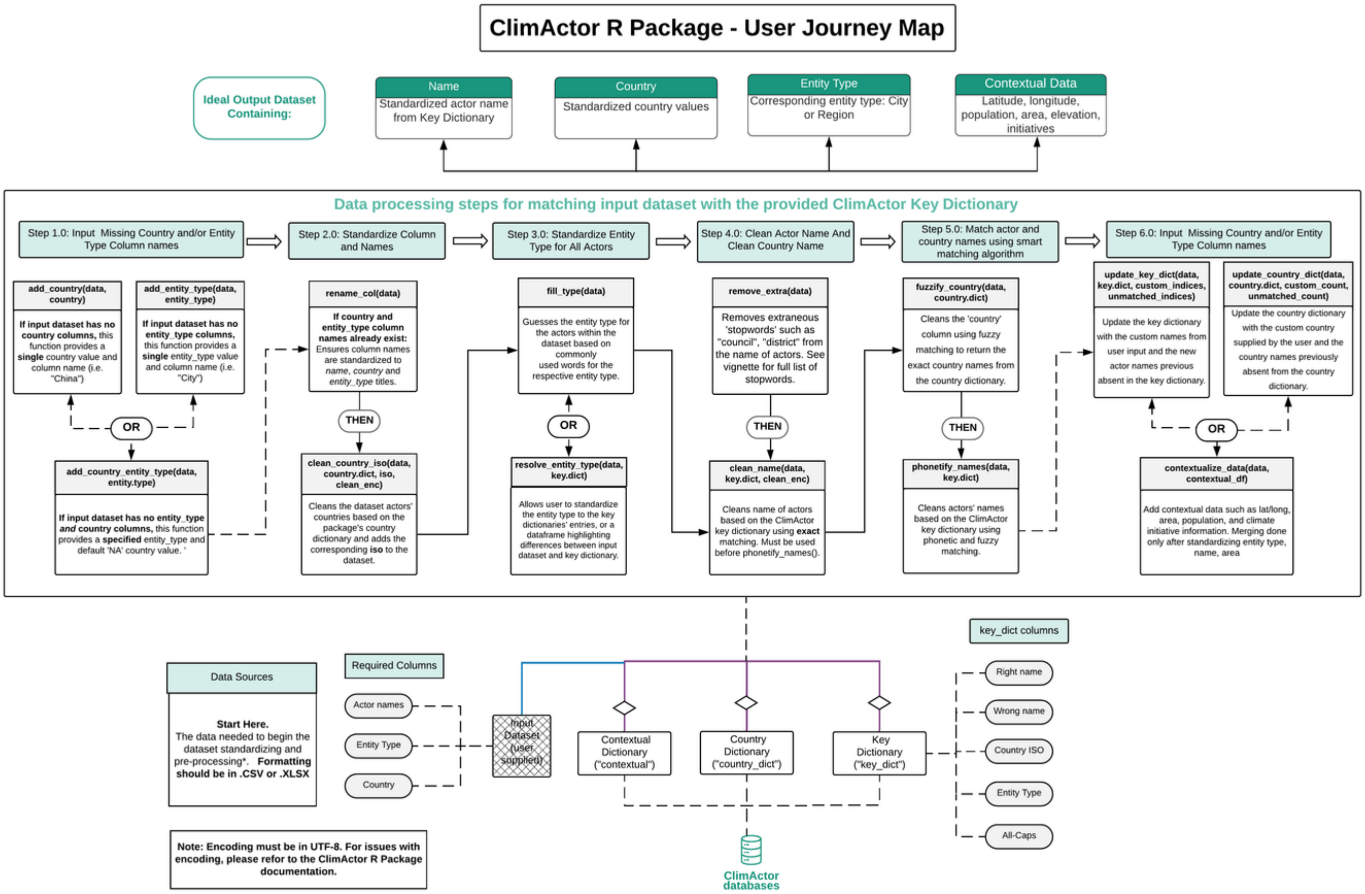
Currently, the algorithm generates the top 15 name matches within the referential database against an incoming actor's name and allows the user to select the correct name of the actor being referred to. However, this process still involves manual intervention from the user and limits the scalability of the amount of data that can be cleaned. A more automated and efficient algorithm is thus needed to enable timely and consistent updates of climate data for analysis.

MAIN PROMPT QUESTION/CHALLENGE

How can a more automated and efficient entity matching algorithm be implemented for entity matching of climate data?

[1] Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Peter Christen

[2] ClimActor, harmonized transnational data on climate network participation by city and regional governments, Angel Hsu et al.


ClimActor R Package - User Journey Map
Ideal Output Dataset Containing:
Name
Standardized actor name from Key Dictionary
Country
Standardized country values
Entity Type
Corresponding entity type: City or Region
Contextual Data
Latitude, longitude, population, area, elevation, initiatives
Data processing steps for matching input dataset with the provided ClimActor Key Dictionary
Step 1.0: Input Missing Country and/or Entity Type Column names
Step 2.0: Standardize Column and Names
Step 3.0: Standardize Entity Type for All Actors
Step 4.0: Clean Actor Name And Clean Country Name
Step 5.0: Match actor and country names using smart matching algorithm
Step 6.0: Input Missing Country and/or Entity Type Column names
add_country(data, country)
If input dataset has no country columns, this function provides a single country value and column name (i.e. "China")
add_entity_type(data, entity_type)
If input dataset has no entity_type columns, this function provides a single entity_type value and column name (i.e. "City")
OR
add_country_entity_type(data, entity.type)
If input dataset has no entity_type and country columns, this function provides a specified entity_type and default 'NA' country value. '
rename_col(data)
If country and entity_type column names already exist: Ensures column names are standardized to name, country and entity_type titles.
THEN
clean_country_iso(data, country.dict, iso, clean_enc)
Cleans the dataset actors' countries based on the package's country dictionary and adds the corresponding iso to the dataset.
fill_type(data)
Guesses the entity type for the actors within the dataset based on commonly used words for the respective entity type.
OR
resolve_entity_type(data, key.dict)
Allows user to standardize the entity type to the key dictionaries' entries, or a dataframe highlighting differences between input dataset and key dictionary.
remove_extra(data)
Removes extraneous 'stopwords' such as "council", "district" from the name of actors. See vignette for full list of stopwords.
THEN
clean_name(data, key.dict, clean_enc)
Cleans name of actors based on the ClimActor key dictionary using exact matching. Must be used before phonetify_names().
fuzzify_country(data, country.dict)
Cleans the 'country' column using fuzzy matching to return the exact country names from the country dictionary.
THEN
phonetify_names(data, key.dict)
Cleans actors' names based on the ClimActor key dictionary using phonetic and fuzzy matching.
update_key_dict(data, key.dict, custom_indices, unmatched_indices)
Update the key dictionary with the custom names from user input and the new actor names previous absent in the key dictionary.
update_country_dict(data, country.dict, custom_count, unmatched_count)
Update the country dictionary with the custom country supplied by the user and the country names previously absent from the country dictionary.
OR
contextualize_data(data, contextual_df)
Add contextual data such as lat/long, area, population, and climate initiative information. Merging done only after standardizing entity type, name, area
Data Sources
Start Here.
The data needed to begin the dataset standardizing and pre-processing*. Formatting should be in .CSV or .XLSX
Required Columns
Actor names
Entity Type
Country
Input Dataset (user supplied)
Contextual Dictionary ("contextual")
Country Dictionary ("country_dict")
Key Dictionary ("key_dict")
key_dict columns
Right name
Wrong name
Country ISO
Entity Type
All-Caps
ClimActor databases
Note: Encoding must be in UTF-8. For issues with encoding, please refor to the ClimActor R Package documentation.

FURTHER DESCRIPTION AND SUPPLEMENTARY QUESTIONS

As mentioned above, the current entity matching algorithm relies on fuzzy matching based on phonetic representations of an actor's name and also requires manual intervention by the user to complete the matching algorithm. However, with advances in machine learning and artificial intelligence, techniques such as neural networks and deep learning have increasingly been applied to solve entity matching challenges and increase scalability and efficiency of entity matching algorithms. This prompt seeks an implementation of a machine-learning driven entity matching algorithm for use in entity matching of climate actors. Additionally, the ability to draw from and combine publicly available data across multiple sources will also be judged favorably.

Some potential supplementary questions to consider when approaching the prompt:

- What is the data validation process going to be for validation and verification of edge cases?
- How do we ensure that incoming data are not duplicated and have no existing records in the referential database?
- Can we expand the entity matching to include non-English names (e.g., Japanese, Spanish) of actors?
- What data features (e.g., population, revenue, etc.) should be used as part of the entity matching process?
- Can the algorithm be used across different entity types (e.g, cities, regions, companies)?

The above dimensions are by no means exhaustive and are meant to serve only as guiding questions or potential areas of investigation. Participants are also encouraged to explore different questions using a combination of the ClimActor data as well as supplementary datasets from external sources.

Strong submissions should also build on the existing exploration and research as well as provide clear and striking visuals as much as possible that illustrate the data from datasets used/overall process.