Detailed Code Development Documentation

Title: Development of an Entity Resolution System to Identifya and group duplicate records in a Company Dataset
Author: Francesco Grasso
Date: March 14, 2025
Context: The dataset contains company records imported from multiple systems, leading to duplicate entries with slight variations.

Introduction

Entity resolution is the process of identifying and linking records that refer to the same entity (e.g., companies with slight variations in names or addresses). This project aims to:

- **Preprocess the data**: Normalize names, URLs, and phone numbers to standardize formats.
- **Optimize blocking**: Group similar records to reduce comparisons.
- **Train a model**: Use supervised machine learning to classify duplicates.
- **Deduplicate records**: Produce a clean dataset with unified entities.

I chose the **Dedupe** library for its flexibility in handling heterogeneous data, ability to learn from supervised training data, and interactive labeling interface. This project reflects an iterative approach, with progressive optimizations to balance performance and results.

Results

Total Records: 33365

Clusters: 7304

No duplicated: 12170

Results saved in 'entity_resolution_result.csv'

Results saved also in 'entity_resolution_result.parquet'

Analysis

count	7304.000000
average	2.901835
standard deviation	1.740159
minimum	1.000000
25th percentile	2.000000
median	3.000000
75th percentile	4.000000

```
maximum      31.000000
dtype: float64

Maximum cluster size: 31

Cluster 2314 - 31 records:
     company_name              website_url
431   Fresh Burger  https://freshburger.com.sa/
600   Fresh Burger  https://freshburger.com.sa/
3373  Fresh Burger  https://freshburger.com.sa/
4491  Fresh Burger  https://freshburger.com.sa/
5877  Fresh burger  https://freshburger.com.sa/

Cluster 5693 - 27 records:
              company_name            website_url
762   Recovera Využití zdrojů   http://www.recovera.cz/
1889  Recovera Využití zdrojů  https://www.recovera.cz/
2143  Recovera Využití zdrojů   http://www.recovera.cz/
4918  Recovera Využití zdrojů   http://www.recovera.cz/
5589  Recovera Využití zdrojů  https://www.recovera.cz/

Cluster 1077 - 26 records:
    company_name         website_url
1789   Inter Cars  http://intercars.cz/
3183   Inter Cars  http://intercars.cz/
5849   Inter Cars  http://intercars.cz/
7619   Inter Cars  http://intercars.cz/
8678   Inter Cars  http://intercars.cz/

Cluster 7287 - 23 records:
    company_name             website_url
753      Chatime    https://chatime.com.ph/
```

```
1372     Chatime  http://www.chatime.com.ph/
1624     Chatime    https://chatime.com.ph/
2345     Chatime  http://www.chatime.com.ph/
3387     Chatime    https://chatime.com.ph/

Cluster 6802 - 22 records:
   company_name            website_url
285     Tomaticos   http://www.tomaticos.com/
5321     Tomaticos   http://www.tomaticos.com/
8765     Tomaticos   http://www.tomaticos.com/
10103    Tomaticos  https://www.tomaticos.com/
10991    Tomaticos   http://www.tomaticos.com/

Clusters with consistent URLs: 68.24%

      id  company_name               website_url                     main_address_raw_text  primary_phone
431     432  Fresh Burger     https://freshburger.com.sa/  Aljawharah، طريق المطار _, Taif 26559, Saudi A...
+966920022097
600     602  Fresh Burger     https://freshburger.com.sa/                   Samtah 86735, Saudi Arabia
+966920022097
3373    3385  Fresh Burger     https://freshburger.com.sa/       King Faisal Rd, Muhayil 63311, Saudi
Arabia  +966920022097
4491    4505  Fresh Burger     https://freshburger.com.sa/  السلامه _، طريق الملك خالد، احد رفيده_, Abha 6...
+966920022097
5877    5891  Fresh burger     https://freshburger.com.sa/               10, Ad Darb 89741, Saudi Arabia
+966920022097
6012    6026  Fresh Burger     https://freshburger.com.sa/  5102 62454 طريق الملك فهد، القافلة، خميس مشيط ...
+966920022097
6641    6656  Fresh Burger     https://freshburger.com.sa/  Ali Bin Abi Talib Road, Al Suraif, Yanbu
46422...  +966920022097
7350    7366  Fresh burger  https://freshburger.com.sa/menu  امام اسواق القاضي, Ad Dammam, Damak، حي,
Khami...  +966920022097
7714    7730  Fresh Burger     https://freshburger.com.sa/  2406 طريق الملك سعود, Alathaybah Ash
Shamaliyy...  +966920022097
```

7796 7812 Fresh Burger https://freshburger.com.sa/menu Ankara, Al Samer، حي, Jeddah 23462, Saudi Arabia +966920022097

8571 8587 Fresh Burger https://freshburger.com.sa/ الاغر حي_عبداللة الملك طريق, Ranyah 29811, Sau... +966920022097

9075 9091 Fresh Burger https://freshburger.com.sa/ Qouraish, السداد_قريش، Taif 26514, Saudi Arabia +966920022097

9486 9502 Fresh Burger https://freshburger.com.sa/ 7627 بلجرشي ،الشطيبة حي ،عبدالعزيز الملك طريق ... +966920022097

12396 12428 Fresh Burger https://www.freshburger.com.sa/ حي_بدر طريق, Alkhaldiyah, Almajaridah 63971, S... +966920022097

13768 13804 Fresh Burger https://freshburger.com.sa/ 5576, Abha 62521, Saudi Arabia +966920022097

14350 14386 Fresh Burger https://freshburger.com.sa/ Al Abbas Ibn Ubadah, Al Hadiqah, Madinah 42383... +966920022097

15920 15956 Fresh Burger https://freshburger.com.sa/ البحرية الواجهة, Al Haridhah 89932, Saudi Arabia +966920022097

19814 19854 Fresh Burger https://freshburger.com.sa/ رغدان غابة ،2رقم بوابة _, Al Bahah 65541, Saud... +966920022097

20009 20052 Fresh Burger https://freshburger.com.sa/ Bisha 67612, Saudi Arabia +966920022097

20816 20860 Fresh Burger https://freshburger.com.sa/ مول جاردن ،, Al Muntazah Al Gharbi, Hail 55427... +966920022097

21581 21627 Fresh Burger https://freshburger.com.sa/ الخطاب بن عمر شارع ،حي, Alkhuzama, Bisha 67711... +966920022097

22066 22113 Fresh burger https://freshburger.com.sa/ 62583, Abha Saudi Arabia +966920022097

22412 22459 Fresh Burger https://freshburger.com.sa/ الحمداني فراس أبي, Al Hamadaniyyah, Jeddah 237... +966920022097

23980 24035 Fresh Burger https://freshburger.com.sa/ King Faisal Rd, Al Namas 67379, Saudi Arabia +966920022097

24266 24321 Fresh Burger https://www.freshburger.com.sa/ , Jeddah, Makkah Province, Saudi Arabia +966920022097

24673 24730 Fresh Burger https://freshburger.com.sa/ حي, King Abdulaziz Rd, Alulaya, Tabuk 47911, S... +966920022097

28691 28765 Fresh Burger https://freshburger.com.sa/ طريق الملك عبدالعزيز_الظفيره, Aldhafir, Al Bah... +966920022097

28714 28788 Fresh Burger https://freshburger.com.sa/ طريق الملك خالد_دوار آل مسعد, Sabt Al Alayah 6... +966920022097

```
31226  31302  Fresh Burger     https://freshburger.com.sa/   Abu Huraira Rd, المنسك، Abha 62527, Saudi Arabia  +966920022097

32097  32173  Fresh Burger     https://freshburger.com.sa/  King Fahd Rd, حي العزيزيه، Al Makhwah 65614, S...  +966920022097

32369  32447  Fresh Burger     https://freshburger.com.sa/  طريق الملك سعود_النقره_العثيم، مول, Hail 55431...  None


Number of unique entities: 12170


Top 10 clusters with at least 10 records:


**Cluster 2314 (31 records):**

 Most common names:

  - Fresh Burger (28)

  - Fresh burger (3)

 Most common URLs:

  - freshburger.com.sa (27)

  - freshburger.com.sa/menu (2)

  - www.freshburger.com.sa (2)


**Cluster 5693 (27 records):**

 Most common names:

  - Recovera Využití zdrojů (24)

  - Recovera Využití zdrojů logistické centrum (1)

  - Recovera Využití zdrojů a.s. (1)

 Most common URLs:

  - www.recovera.cz (27)


**Cluster 1077 (26 records):**

 Most common names:

  - Inter Cars (26)

 Most common URLs:

  - intercars.cz (20)
```

```
   - www.intercars.cz (6)

**Cluster 7287 (23 records):**
  Most common names:
   - Chatime (21)
   - ChaTime (2)
  Most common URLs:
   - chatime.com.ph (17)
   - www.chatime.com.ph (6)

**Cluster 6802 (22 records):**
  Most common names:
   - Tomaticos (21)
   - TOMATICOS (1)
  Most common URLs:
   - www.tomaticos.com (21)
   - tomaticos.com (1)

**Cluster 6497 (20 records):**
  Most common names:
   - Söderberg & Partners (20)
  Most common URLs:
   - www.soderbergpartners.se/om-oss/kontor/sverige/kristianstad/kristianstad-vastra-storgatan-29/%3futm_medium%3dgooglemybusiness%26utm_source%3dgoogle%26utm_campaign%3dsoderberg-partners_brand_google-my-business&opi=79508299&sa=u&ved=0ahukewj-nylhhmwhaxxof1kfhdrwb_4q61gieigp&usg=aovvaw1zyoak-ubwshduy_le8t4t (1)
   - www.soderbergpartners.se/om-oss/kontor/sverige/goteborg/goteborg-kungstorget-5/%3futm_medium%3dgooglemybusiness%26utm_source%3dgoogle%26utm_campaign%3dsoderberg-partners_brand_google-my-business&opi=79508299&sa=u&ved=0ahukewioq7jyspoiaxxxsvedhvjen8qq61gieigp&usg=aovvaw2h9lgwg6q8a0qpmzod_9b- (1)
   - www.soderbergpartners.se/om-oss/kontor/sverige/gavle/gavle-norra-skeppargatan-5/%3futm_medium%3dgooglemybusiness%26utm_source%3dgoogle%26utm_campaign%3dsoderberg-partners_brand_google-my-business&opi=79508299&sa=u&ved=0ahukewi1jl2v5cwgaxwhi0qihdnsmk0q61gieigp&usg=aovvaw0lba6yqago3y1tfnbmqxkd (1)
```

Cluster 310 (18 records):

Most common names:

- Avangard (18)

Most common URLs:

- www.avangard.ru (18)

Cluster 7124 (17 records):

Most common names:

- Toys R Us (16)
- Toys“R”Us (1)

Most common URLs:

- www.toysrusmena.com/en-ae (11)
- www.toysrusmena.com (6)

Cluster 5771 (16 records):

Most common names:

- Ramsay Sante (16)

Most common URLs:

- www.ramsaysante.fr (16)

Cluster 1234 (16 records):

Most common names:

- Plastic Surgery Assoc. (6)
- PLASTIC SURGERY ASSOCIATES OF SOUTH DAKOTA (4)
- Plastic Surgery Associates of South Dakota Ltd. (1)

Most common URLs:

- www.plasticsurgeryassociatesofsd.com (16)

Device resolution explains with simple words

Let's sit together and interact on what we have discovered with this analysis, in a way that is easy to understand, even if you are not a computer expert. The goal is to make it clear what we did, what we got, and why all this means something.

We took a database packed with company information - 33 365 items, to be accurate - and asked ourselves: How many times the same company appears here with slightly different names or details? To answer this, we used an "entity resolution" algorithm, originally a digital detective: it seeks a clue to find out which items talk about the same company, whether they are written in a slightly different way or come from different sources. This work is important because counting the same company repeatedly causes errors, while merging duplicated data gives us a complete, more reliable image for any future analysis.

What do we have?

So what did we get? Well, there was a lot of repetition! Of the 33 365 original items:

There were 21 195 duplicates, which means they had already referred to companies in the database.

After they were met, we were left with 12 170 unique institutions - that is, separate companies.

The algorithm created 7,304 clusters, each cluster represents the same company with all its duplicated records.

On average, each cluster has about 3 items, which means a company usually appears three times in the original database, perhaps with a slight change in the data.

A real example: fresh burgers

To clarify this, let's look at the cluster "Fresh Burger", which is the largest group of 31 records. All of them are called "Fresh Burgers" (with a slight difference in capitalization), and they share:

The same site: freshburger.com.sa

Same phone number: +966920022097

However, the addresses are different because each mail represents another location for this fast food chain in Saudi Arabia. Thanks to the algorithm, instead of considering these 31 places as separate companies, we now know that they are part of the same fresh burger. It's like feeling that many contacts on your phone - such as "John Home" and "John Office" - really the same person!

Did the algorithm did well?

How do we know that our digital detective has worked well? A clue is URL stability: In 68.24% of the groups, all items contain the same webdom, which is a strong indication that the group makes sense. Seeing the largest groups also helps:

Fresh Burger: 31 items, the same URL.

Recora vyujití zdrojů: 27 items, the same URL.

Different cars: 26 items, same domain.

These examples show the algorithm to properly recognized companies with many places or spread data, logical Lund.

What can I do about it?

Now that we have these results, there are some practical uses here:

Clean the database: Keep just one post per company - perhaps the most complete - to dig the duplicate.

Create a rich profile: For a complete image of each company, mix information from each cluster (phone number, address, site).

Map site: Fresh burger -like plot chain to see their spread.

Horoscope marketing: Target the original company instead of each place, save time and effort.

In short, our algorithm naked it: About 63% of the original items were duplicate, and it arranged them in intelligent groups. Now we have a cleaner of the database, Clear View, where companies are correctly grouped. This is a great victory that opens up many opportunities - everyone explained, I hope in a simple and friendly way!

Code Evolution

The development occurred in several stages, each addressing specific technical challenges:

1. Initial Version

- **Objective**: Create a basic working system.
- **Implementation**:
  - Loaded the dataset (e.g., CSV with columns like name, URL, address, phone, country).
  - Preprocessing: Normalized data (e.g., removed legal suffixes from names).
  - Blocking: Created keys based on name, URL, address, and country.
  - Interactive training: Used console_label to manually label record pairs as duplicates or distinct, saving results in training.json.

    - Deduplication: Clustered similar records and saved the output.
- **Result**: A functional prototype but inefficient for large datasets.

2. Blocking Optimization

- **Issue**: Analysis showed too many small blocks (average of 1.93 records per block), with many containing only one record, increasing unnecessary comparisons.
- **Solution**:
    - Modified the create_blocking_key function to use only name and country, reducing the number of keys and increasing average block size.
- **Rationale**: Reduce computational load while maintaining meaningful groupings.

3. Memory Management

- **Issue**: Processing large datasets caused memory issues.
- **Solution**:
    - Introduced process_block_batch: Processed blocks in batches rather than all at once.
    - Used gc.collect() to free memory between batches.
- **Rationale**: Ensure scalability on real-world datasets, avoiding hardware crashes.

4. Separation of Interactive Labeling

- **Objective**: Improve modularity and reuse of training data.
- **Implementation**:
    - Created a separate script (interactive_labeling.py) for interactive labeling, saving results in training_data.json.
    - Modified the main code to load the pre-existing training file and proceed with deduplication.
- **Rationale**: Separate training and deduplication phases for flexibility and easier debugging.

5. Debugging and Robustness

- **Objective**: Resolve errors and improve stability.
- **Implementation**:
    - Added checks to ensure required fields were present in records.
    - Introduced logging to track execution and identify issues.
- **Rationale**: Make the system robust and capable of handling incomplete or malformed data.

Errors Encountered and Solutions

During development, I faced several technical challenges, resolving them with targeted solutions:

1. Memory Error

- **Issue**: MemoryError when processing large blocks.
- **Solution**: Implemented process_block_batch to work on small groups of records and freed memory with gc.collect().
- **Rationale**: Prevent system crashes on large datasets, maintaining scalability.

2. Data Validation Error

- **Issue**: ValueError: Records do not line up with data model when a field (e.g., clean_company_name) was missing in a record.
- **Solution**: Added checks to set missing fields to None and ensure all records conformed to the model.
- **Rationale**: Make the code resilient to imperfect data, typical in real-world datasets.

3. Inefficient Blocks

- **Issue**: Too many single-record blocks slowed the process without improving results.
- **Solution**: Simplified blocking keys (only name and country), creating larger, more useful blocks.
- **Rationale**: Optimize performance by reducing the total number of comparisons.

4. Invalid Training Data

- **Issue**: Errors loading training_data.json due to malformed records.
- **Solution**: Rigorous cleaning of training data, checking for fields and validity.
- **Rationale**: Ensure the model learned from consistent data, improving accuracy.

Architectural and Technological Choices

Dedupe:

After evaluating several options, the main libraries were assessed, Dedupe, Recordlinkage, Name Matching and others mentioned in other online resources, such as Splink and Pyjedai. Below is a detailed evaluation:

- Dedupe: This library uses machine learning to do and device resolution on structured data. It is designed to handle large datasets and is especially useful for removing duplicate in the same dataset as needed. It provides a python interface that can work with panda data frame, making it compatible with the user's wood. This requires interactive training, where the user manually notes a post pair such as duplicate or non-duplicates, which improves accuracy. Examples include lack of business names and addresses, with support from the company's name and addresses as fields.

- Recordlinkage: This library provides equipment for determinable and potential record coupling including didup. It provides features to block the records to find matches, secure and compare records. However, compared to Dedupe, this requires more manual configuration and seems less focused on machine learning, which can make less suitable for complex datasets without significant intervention.

- Name matching: Developed by Dutch Central Bank, this library is specifically to match commercial names between two data sets, with the options for handling the legal suffix and distance measurements (eg bags, typos, typos, refined_soundax). However, it is more oriented towards matching between different data sets, not internal dedication, it makes it less ideal for the user's function. This can be useful for advance, before using Dedupe before standardizing the company's name.

- Other options: libraries such as Splink (scalable with SQL or kick) and pyjedai (advanced grouping algorithms) were assessed, but more complex configurations (eg SQL/Spark Backend) are required, which may not be necessary for a standalone Pythan project. Given the lack of details about the dataset, Dedup seems more accessible and straightforward.

Details of Dedupe

Dedupe requires interactive training, where the user must notice some pairs as a duplicate or non-duplicate, which may be timing, but improves the results. There is also a library called "Nam-Milan" for business names, but it is more favorable to match between two data sets, not internal cuts, to make Dedup more appropriate.

Rating of dataset and preaching

The dataset, with three three, may be of all sizes, but Dedupe is designed to handle large versions of data. However, to improve the results, it is recommended for preprise data, standardization of company names (eg removal of punctuation, converting to lowercase letters, handling legal suffix). Although Dedupe has some underlying preparatory features, extra cleaning may be required, especially for addresses or other complex areas.

2. Data Preprocessing

- **Company Names**: clean_company_name function to remove legal suffixes (e.g., "LLC", "Inc") and irrelevant characters.
    - **Rationale**: Standardize names for better comparison.
- **URLs**: Normalization with normalize_url to standardize format (e.g., removing "www").
    - **Rationale**: Reduce irrelevant variations.
- **Phones**: Conversion to E164 format using the phonenumbers library.
    - **Rationale**: Ensure consistency in international numbers.

3. Blocking Strategy

- **Approach**: Used doublemetaphone for names (phonetic) and parts of the URL as keys.
- **Optimization**: Reduced attributes to create larger blocks.
- **Rationale**: Balance efficiency (fewer blocks) and accuracy (meaningful groupings).

4. Memory Management

- **Technique**: Batch processing with process_block_batch and gc.collect().
- **Rationale**: Adapt the system to varying dataset sizes, avoiding overloads.

5. Code Modularity

- **Choice**: Separated interactive labeling into a dedicated script.
- **Rationale**: Facilitate reuse of training data and simplify debugging, making the code more maintainable.