Read Me - HCAD Enrichment

HCAD Real Property Enrichment Script (Script 4)

Project Goal
This script (Script 4) is designed to enrich probate leads with current property ownership information from the Harris County Appraisal District (HCAD) websi te. It takes property information linked to a decedent (from a previous proces s, Script 3), searches for that property on HCAD, and extracts current owner d etails and property values.

Features
- Takes a CSV input containing property details and probate lead informatio n.
- Uses a tiered search strategy on the HCAD Advanced Real Property Search to find matching properties.
- Handles different HCAD search outcomes: unique match, multiple matches, no matches.
- For multiple HCAD matches, attempts to disambiguate using a scoring logic based on legal description components.
- Fetches detailed information from the HCAD property detail page for match ed properties.
- Compares the current HCAD owner with the original decedent (grantor) an d grantees from the input deed information using fuzzy string matching.
- Outputs an enriched CSV with HCAD data and new match status flags.
- Includes configurable parameters for file paths and search behavior.

Prerequisites
1. **Python 3.9+**
2. **Pip** (Python package installer)
3. **Playwright Browsers:** The script uses Playwright for browser automatio n. You'll need to install the browser binaries:

```bash
pip install playwright
playwright install
# or playwright install chromium
```
4. **Required Python Packages:** Install them using pip:
```bash
pip install pandas rapidfuzz
```

Setup

1. **Clone/Download the Script:**
Place `script4_hcad_enrichment.py` (and any helper files if it's broken into modules) into your project directory.

2. **Input CSV File:**
* This script expects an input CSV file generated by "Script 3".
* **Crucial Columns Expected in Input CSV** (ensure your Script 3 output provides these with these *exact* names, or update the script's column name references):
    * `rp_file_number`: Unique identifier for the property transaction/deed.
    * `rp_party_type`: Type of party in the row (e.g., "Grantor", "Grantee").
    * `is_potential_decedent_match`: Boolean (True/False) indicating if the party in this row is the probate decedent.
    * `probate_lead_case_number`: Unique identifier for the probate case.
    * `match_confidence_level`: Confidence of the link from Script 3 (e.g., "High", "Medium", "Low").
    * `probate_lead_decedent_first`, `probate_lead_decedent_last`: First and last name of the decedent from the probate lead.
    * `rp_party_first_name`, `rp_party_last_name`: First and last name of the party on the RP deed row (used for creating full names).
    * `rp_legal_description_text`: Subdivision name or main legal text.
    * `rp_legal_lot`: Lot number.
    * `rp_legal_block`: Block number.
    * `rp_legal_tract`: Tract number.

    *   `rp_legal_sec`: Section number.
    *   *(And any other columns from Script 3 you wish to carry through to the final output).*
*   **Delimiter:** The script currently expects the input CSV to be **semicolon-delimited (`;`)**. If your CSV uses commas or another delimiter, you must update the `pd.read_csv()` line in the `if __name__ == '__main__':` block.

3. **Configure Input File Path:**
*   Open `script4_hcad_enrichment.py`.
*   Locate the `if __name__ == '__main__':` block at the end of the script.
*   Modify these lines to point to your input data:
    ```python
    INPUT_DATA_FOLDER = "/path/to/your/script3/output_folder"
    INPUT_FILENAME = "your_script3_output_file.csv"
    TARGET_INPUT_CSV_PATH = os.path.join(INPUT_DATA_FOLDER, INPUT_FILENAME)
    ```
    Replace with your actual folder and file name. The script can also be configured to use a `find_latest_csv_in_folder` function if you prefer.

Running the Script

1. Navigate to the script's directory in your terminal.
2. Run the script using Python:
```bash
python script4_hcad_enrichment.py
```

Current Test Configuration (Important):
*   By default, the script is configured in the `if __name__ == '__main__':` block to load your specified QA CSV sample.
*   It is also currently set to filter and process **only records where `match_confidence_level` is 'High'**.
*   To process all records or different confidence levels, you will need to modify the filtering logic within the `if __name__ == '__main__':` block.

Output
The script will generate a new CSV file named something like `script4_hcad_enriched_QA_output_HIGH_ONLY.csv` (the name changes based on input and filters). This file will contain:
* All columns from your input CSV.
* **New HCAD-derived columns:**
  * `hcad_detail_url_visited`: Link to the HCAD property page.
  * `hcad_account`: HCAD account number.
  * `hcad_owner_full_name`: Current owner(s) from HCAD.
  * `hcad_mailing_address`: Owner's mailing address.
  * `hcad_legal_desc_detail`: HCAD's legal description.
  * `hcad_site_address`: Property's physical address.
  * `hcad_pct_ownership`: (Currently not scraped, likely None).
  * `hcad_market_value_detail`, `hcad_appraised_value_detail`: Property values.
  * `hcad_land_area_sf`, `hcad_total_living_area_sf`: Property size info.
  * `parsing_error`: Any error message if detail page parsing failed.
* **New Script 4 Processing & Match Columns:**
  * `rp_grantee_full_names_list`: A list of grantee names from the original deed (consolidated).
  * `hcad_search_status`: Overall outcome of the HCAD search for this property (e.g., `SUCCESS`, `NO_HITS`, `MULTIPLE_HITS_NO_WINNER_TIER_X`, `PAGINATION_TOO_LARGE`, `COMMON_SURNAME_TOO_BROAD`, `DETAIL_PARSE_ERROR`, `SKIPPED_INSUFFICIENT_DATA`).
  * `hcad_final_tier_hit`: Which search tier (T1-T4, Fallback) successfully found the match.
  * `hcad_owner_matches_rp_grantor_score`: Fuzzy match score (0-100) between HCAD owner and original grantor (decedent).
  * `is_owner_grantor`: `1` if HCAD owner matches grantor, `0` otherwise.
  * `hcad_owner_matches_rp_grantee_score`: Highest fuzzy match score (0-100) between HCAD owner and any of the original grantees.
  * `is_owner_grantee`: `1` if HCAD owner matches any original grantee, `0` otherwise.
  * `hcad_owner_match_type`: A summary status (e.g., `MATCHES_GRANTOR`, `MATCHES_GRANTEE`, `NEW_THIRD_PARTY`, `UNKNOWN_NO_GRANTEES_ON_DEED`).

```
    *   `needs_review_flag`: `1` if the script suggests manual review for this reco
rd, `0` otherwise.
    *   `review_reason`: A brief explanation if `needs_review_flag` is `1`.

## Understanding the Search Tiers (Simplified)
The script tries to find properties on HCAD using a series of search methods:
1.  **Tier 1 (Exact Legal):** Uses the most complete legal description (Tract, Bl
ock, Subdivision, Section).
2.  **Tier 2 (Drop Section):** Same as Tier 1 but without the Section number.
3.  **Tier 3 (Owner + Key Legal):** Searches by Owner Last Name + Subdivisi
on (plus Block/Tract if available). Tries with Grantee names first, then Decede
nt name. Skips if owner name is too common and legal info is too broad.
4.  **Tier 4 (Subdivision + Block):** Searches by Subdivision + Block. If multip
le results, compares full legal descriptions.
5.  **Fallback (Owner + Basic Legal):** Decedent Last Name + Subdivision (pl
us Block OR Tract if available).

## Key Files Generated During Run
*   **Output CSV:** e.g., `script4_hcad_enriched_QA_output_HIGH_ONLY.csv`
*   **Screenshots (`.png`):** Taken automatically if errors occur (e.g., `form_el
ements_not_visible_...png`, `search_exception_...png`). These help diagnose is
sues.
*   **HTML Dumps (`.html`):** Full HTML of a page is saved if certain errors o
ccur (e.g., `iframe_content_NO_SPECIFIC_INDICATOR_...html`). Useful for seei
ng what Playwright saw.

## Troubleshooting Common Issues
*   **`ERROR: Could not read CSV file ...`:** Ensure `TARGET_INPUT_CSV_PAT
H` in the script correctly points to your input file and that the file uses semicol
ons (`;`) as delimiters.
*   **`ERROR: Column '...' not found in the input DataFrame.`:** Verify that your
input CSV contains all the "Crucial Columns Expected" (listed above) with the
exact names the script uses.
*   **Script processes 0 records after "Processing X 'High' confidence record
s":** This usually means your input file, after preprocessing and filtering for
'High' confidence, resulted in an empty list of records to process. Check your
```

input data and the `match_confidence_level` values.
*   **Many `NO_HITS` or `MULTIPLE_HITS_NO_WINNER`:** This might be expected for some data. If the rate is too high, the scoring logic in `choose_best_from_multiple` or the search query construction in `construct_search_query` may need tuning.
*   **`DETAIL_PARSE_ERROR` or missing HCAD fields:** The XPaths in `parse_hcad_detail_page` might need adjustment if HCAD's detail page structure changes or varies.

Future Enhancements / TODOs
- Refine scoring logic in `choose_best_from_multiple`.
- Implement more sophisticated handling for 'Medium' confidence leads (e.g., lighter disambiguation).
- Consider limited pagination for broad search tiers if necessary.

How to Use This README:

1. Save it as `README.md` in the same directory as your `script4_hcad_enrichment.py`.
2. **Fill in the Blanks/Placeholders:**
    - Update the "Crucial Columns Expected" list if your column names from Script 3 are different.
    - Modify the `INPUT_DATA_FOLDER` and `INPUT_FILENAME` examples in the "Configure Input File Path" section to reflect your typical setup.
3. **Review and Adapt:** Read through it and adjust any explanations to better match the exact final state of your script and the understanding of your VA team. For example, you might add more detail on how VAs should interpret specific `review_reason` codes.

This README should give your VAs (and your future self) a good starting point for understanding and using the script!