INFLUENCERS BASED ON NFS DATASET USING LARGE LANGUAGE MODELS

Date: April 30, 2025

1. INTRODUCTION

The identification of influential researchers within specific academic domains is crucial for understanding research trends, fostering collaboration, and informing funding strategies. This report details a computational methodology developed to identify and rank influential researchers based on quantitative metrics derived from publicly available research award data. The approach combines systematic data processing, the calculation of specific influence-related metrics, and the utilization of a Large Language Model (LLM) for multi-criteria ranking. This methodology was applied to identify potential influencers within the domain of Computer Science.

2. METHODOLOGY

The process involved several distinct stages, from data acquisition to verification, ensuring a robust and transparent workflow.

2.1. Data Source and Acquisition

The primary data source consisted of structured JSON files, each representing a research award. These files contained detailed information, including, but not limited to:

- Award identifiers, titles, abstracts, and types.
- Funding organization details.
- Performing institution information.

- Program Element and Program Reference codes indicating research areas.
- Investigator details, including Principal Investigator (PI) ID, full name, role (e.g., Principal Investigator, Co-Principal Investigator), department, and award start date.

Data was systematically read from a directory structure containing these JSON files. Error handling was implemented to manage potential file reading or JSON parsing issues.

2.2. Data Preprocessing and Structuring

Raw data extracted from the JSON files was processed and structured using the Pandas library:

1. Record Creation: Each investigator associated with an award generated a distinct record, linking PI details with the corresponding award information.
2. DataFrame Construction: These records were aggregated into a central Pandas DataFrame.
3. Data Cleaning:
    - Records lacking a crucial pi_id were removed.
    - The dataset was filtered to retain only records where the investigator's role was explicitly 'Principal Investigator' or 'Co-Principal Investigator', focusing the analysis on individuals in key leadership/participation roles.
    - Missing values (NaN) in critical text fields used for analysis were imputed with empty strings.

2.3. Feature Engineering and Influence Metric Definition

To quantitatively assess influence, three key metrics were defined and calculated for each investigator:

1. Project Volume: Defined as the total number of unique, distinct awards on which an individual served as either PI or Co-PI. This metric reflects the breadth of an investigator's funded research activity. Calculated by counting unique award_title entries associated with a given pi_id after role filtering.
2. Network Size (Collaborators): Defined as the total number of unique *other* PIs or Co-PIs with whom an individual collaborated across all their projects. This metric quantifies the size and reach of an investigator's collaborative network. Calculated by:
    - Identifying all unique projects associated with the target PI.
    - Finding all PIs/Co-PIs listed on those same projects.
    - Creating a unique set of these collaborators' full names.
    - Excluding the target PI's own name from this set.
    - Counting the number of names remaining in the set.

3. Field Diversity: Defined as the number of unique research fields associated with an investigator's awards. This metric indicates the breadth of an investigator's expertise or cross-disciplinary engagement. Calculated by:
    - Collecting all non-null program_element and program_reference codes associated with the PI's awards.
    - Creating a unique set of these codes.
    - Counting the number of elements in the set.

Additional features like leadership (binary indicator for PI role) and experience_years (calculated from award start dates) were also engineered but not directly used in the final LLM ranking prompt for this specific analysis.

2.4. Candidate Selection

For this specific analysis focused on Computer Science:

1. The primary DataFrame (df) was filtered to identify all records where the department field contained "Computer Science" (case-insensitive partial match).
2. Unique pi_ids from this subset were extracted.
3. To narrow the field for detailed LLM analysis, these PIs were ranked based on their total award_count.
4. The top 20 PIs according to this award count ranking were selected as the candidate pool for influencer analysis.

(Note: The framework also supports topic-based candidate selection using text embeddings and cosine similarity).

2.5. LLM-Based Ranking (Google Gemini)

A Large Language Model (Google Gemini, specifically **gemini-2.0-flash-thinking-exp-01-21**) was employed to rank the selected candidates based on the defined influence metrics:

1. Input Formatting: A structured text summary was generated, listing each candidate PI along with their calculated Project Volume, Network Size (Collaborators), and Field Diversity metrics.
2. Prompt Engineering: A detailed prompt was constructed, providing the LLM with context, data, and explicit instructions to rank candidates based *solely on the provided metrics* and to justify each rank by citing those metrics.
3. API Interaction: The prompt was sent to the Gemini API, and the textual response containing the ranking and justifications was retrieved.

2.6. Verification Protocol

To ensure the integrity of the data presented to the LLM:

1. A specific PI from the LLM's ranking was selected.
2. A dedicated function (verify_pi_metrics) recalculated the three key influence metrics directly from the source DataFrame (df) for this specific PI, using the same logic as defined in Section 2.3.
3. The results of this independent calculation were compared against the metrics reported for that PI in the LLM's output justification.

3. RESULTS

The application of the methodology to the Computer Science domain yielded the following key results:

- **Candidate Pool:** 2,938 unique PIs were initially identified within the "Computer Science" department filter. The top 20 PIs based on award count were selected for LLM analysis.
- **LLM Ranking:** The Gemini model successfully processed the input and generated a ranked list of the 20 candidates. The model provided justifications for each rank, referencing the provided metrics. Prasad Calyam (ID: 269779708) was ranked #1.
- **Metric Verification:** The verification protocol was executed for Prasad Calyam.
    - The independent verification calculated: Project Volume=14, Network Size=25, Field Diversity=23.
    - These verified metrics precisely matched the metrics cited by the LLM in its justification for ranking Prasad Calyam #1 (Projects: 14, Collaborators: 25, Fields: 23).

4. DISCUSSION

This methodology provides a data-driven, quantitative approach to identifying potentially influential researchers. By defining influence based on measurable metrics related to activity volume, network connectivity, and disciplinary breadth, it offers a systematic alternative or complement to traditional assessments.

The use of an LLM allows for sophisticated ranking based on multiple criteria, interpreting the provided metrics according to the specified definition of influence. The verification step is critical for ensuring that the LLM's ranking is based on accurate input data.

Potential limitations include data quality dependencies, the specific definition of influence metrics used, potential candidate selection bias, and the precision of department matching.

5. CONCLUSION

The described methodology successfully identified and ranked potential research influencers within the Computer Science domain using a combination of data processing, quantitative metric calculation, and LLM-based analysis. The verification process confirmed the accuracy of the metrics used for the top-ranked individual in the presented run. This approach demonstrates a viable and transparent technique for leveraging research award data to gain insights into research leadership and connectivity within academic fields.

6. APPENDIX: TECHNICAL DETAILS

- **Core Libraries:** Python 3.x, Pandas, NumPy, Sentence Transformers (all-MiniLM-L6-v2), Scikit-learn, Google Generative AI SDK (google-generativeai).
- **LLM:** Google Gemini (gemini-1.5-flash-latest).
- **Data Format:** JSON.