Soft Foreign Keys: Linking Drug Review Data with Standardized Drug Tables

Abhay Shashidhara
University of Minnesota, USA
shash024@umn.edu

Bhuvana Gopalam
University of Minnesota, USA
gopal188@umn.edu

Mohith Siva Sai Bayana
University of Minnesota, USA
bayan009@umn.edu

Sahitya Avadhanam
University of Minnesota, USA
avadh015@umn.edu

Abstract

Although extensive real-world input can be found in online drug evaluations, medicine names are rarely typed exactly as they appear in clinical databases. This makes it challenging to use conventional foreign keys to link reviews to standardized drug tables. To address this challenge, we create a "soft foreign key" pipeline: an LLM predicts the general name for a noisy free-text input, embeddings and a vector database recover close matches, and a last LLM step selects the best one with a confidence score. In order to join reviews to a generic-drug table and make them accessible through a straightforward chatbot for drug and review queries, we store these links in PostgreSQL. This demonstrates how LLMs in conjunction with vector search may transform noisy input dataset into a useful format for relational databases.

1. Introduction

This work is motivated by the challenge of integrating noisy, real-world drug review data with standardized pharmaceutical databases. We focus on building a “soft foreign key” between noisy review data and a standardized generic-drug table using a mix of SQL and modern LLM’s.

1.1 Background

People talk a lot about medicines online, sharing what a drug felt like, whether it worked, and which side-effects they saw. But the names they use informal or abbreviated references like “BP med”, and these typos rarely match the clean generic names stored in databases. In our project this challenge is evident through two public datasets. The generic_drugs table comes from the Mendeley dataset, which lists standardized generic and brand names along with dosage strengths and manufacturers [1]. The drug_reviews table comes from Kaggle[2]. Both tables describe the same kinds of medicines, but because urlDrugName does not reliably match any primary key in the generic table we cannot use a traditional foreign key to join them.

1.2 Goals

The goal of this project is to build a usable bridge between the noisy review data from the Kaggle Patient Ratings: Identifying Best Drugs dataset and the standardized catalog in the Mendeley Medicinal Products in Bangladesh dataset. At a technical level, we aim to infer, for each free-text urlDrugName in the review table, the most likely generic drug in the reference table and to record this relationship inside PostgreSQL as a soft foreign key with an associated confidence score. This linked representation should behave like a conventional foreign key for downstream analysis, allowing queries that aggregate ratings, side effects, and comments at the generic-drug level, while still acknowledging uncertainty in the mapping. A secondary goal is to make these links directly useful to end users by exposing the integrated data through an LLM-based interface .

2. Soft Foreign Key

In a normal relational schema, a foreign key is an exact match: every value in the child table must equal a primary key value in the parent table. With our data this is not realistic, because the urlDrugName field in the Kaggle reviews is noisy and often does not match any generic name in the Mendeley catalog exactly. We still want a join between reviews and generic drugs, but we have to accept that it can only be approximate.We therefore treat the link as a soft foreign key. For each review, our pipeline infers a most likely generic drug and writes this into a standard_name column in the unified review table, together with a confidence score. When the confidence is above a threshold, standard_name behaves like a foreign key into the generic_drugs table, and analysts can join on it as if it were exact. When the confidence is low, the field is left empty and the review is treated as unmapped. In this way we keep the relational structure of the schema while acknowledging that referential integrity is probabilistic rather than guaranteed.Taken together, this soft foreign key design preserves the usual advantages of SQL analysis, such as joins, grouping and indexing, while still working with the messy drug names that appear in patient reviews.

3. Data and Schema

Our system uses two main sources of information: a structured catalog of medicinal products and a collection of patient reviews. This section describes what each dataset looks like, how we cleaned and trimmed them, and how they were mapped into a relational schema in PostgreSQL that the soft foreign key can connect to.

3.1 Dataset Overview

We leverage the Kaggle Patient Ratings: Identifying Best Drugs! dataset for user-generated ratings, and the Mendeley Medicinal Products in Bangladesh dataset as a reference table for standardized pharmacological information. When combined, they provide two contrasting perspectives on the same domain: a curated, pharmacy-facing table and a table that depicts patients' regular conversations about medications. Once the soft foreign key is available, these two views can be connected thanks to the schema's architecture.

3.2 Generic Drug Catalog (Mendeley)

Each entry in the Mendeley dataset, which lists pharmaceuticals sold in Bangladesh, describes a specific formulation. We view it as a list of generic medications with additional features for our purposes. Columns including the generic name, one or more brand names, dose strength, dosage form, and manufacturer are retained from the original file. These fields are sufficient to provide a drug's level of description required for mapping. This becomes the generic_drugs table in PostgreSQL after obvious duplicates and incomplete data are removed. Each row is given a synthetic identification, and the generic name column serves as the logical reference point for the soft foreign key.

3.3 Patient Review Dataset (Kaggle)

Patient reviews for a variety of medications can be found in the Kaggle dataset. The urlDrugName string from the website, a numerical rating, a condition label, and many free text boxes detailing advantages, side effects, and general remarks are all included in each row, which represents a single review. Since these are the components that are important for mapping and the chatbot, we retain the review identifier, urlDrugName, the primary rating column, and the consolidated review text. Since they cannot be connected to any generic item, records with missing or empty urlDrugName are filtered out. The resulting table retains the original noisy names precisely as they appear in the dataset and is loaded into PostgreSQL as drug_reviews_raw.

3.4 Exploratory Data Analysis (EDA)

Before defining the final schema, we performed a brief exploratory analysis of both datasets to understand uniqueness and missingness patterns. For the Mendeley catalog, we counted distinct values per column after dropping the brand field. Figure 1 shows that the packageMark column has far more distinct values than genericName, strength, manufacturer, and dosageType, which confirms that package level strings are too granular to act as keys, while the normalized generic name is a better anchor for mapping. We also inspected missing values in the same table. Strength is informative but cannot be regarded as a necessary property for every row since, as Figure 2 illustrates, nearly all nulls are concentrated in the strength column while the other attributes are practically complete. Figure 3 shows the number of missing values for each column in the Kaggle review dataset. While urlDrugName and rating are essentially complete, fields like benefitsReview and sideEffectsReview show obvious omissions. These missing entries are dispersed throughout the dataset, not limited to a small group of reviews, as the null map in Figure 4 illustrates. Based on these observations, we treat urlDrugName and genericName as the primary text signals for mapping and keep reviews even when some of the detailed text fields are missing, since they still provide useful ratings and partial feedback.

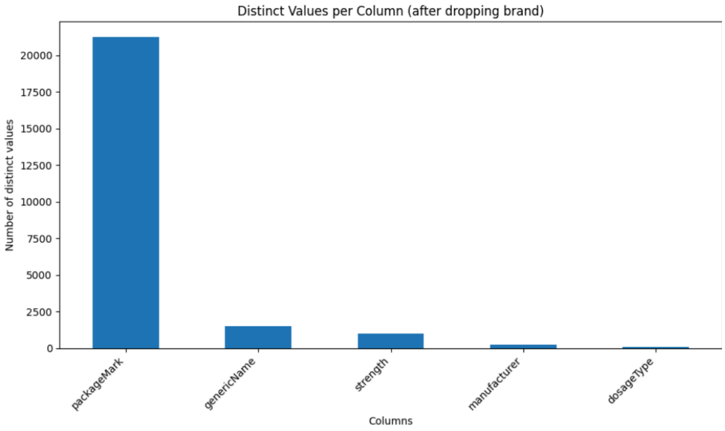


Figure 1:Distinct values per column in the Mendeley catalog after dropping the brand field

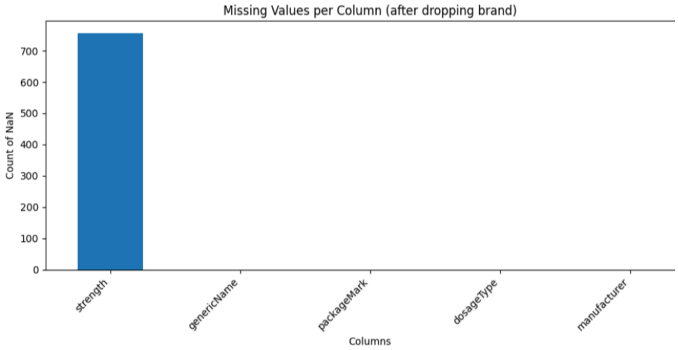


Figure 2:Count of missing values per column in the Mendeley catalog after dropping the brand field

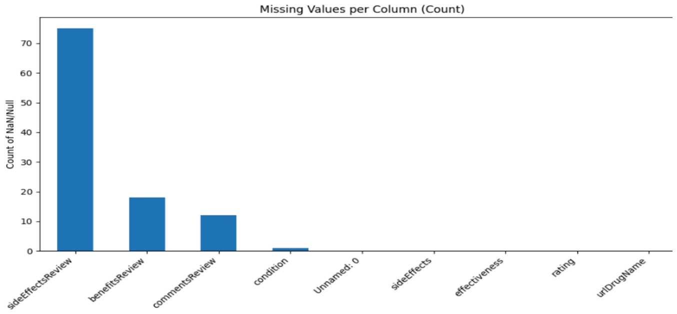


Figure 3: Count of missing values per column in the Kaggle patient review dataset.

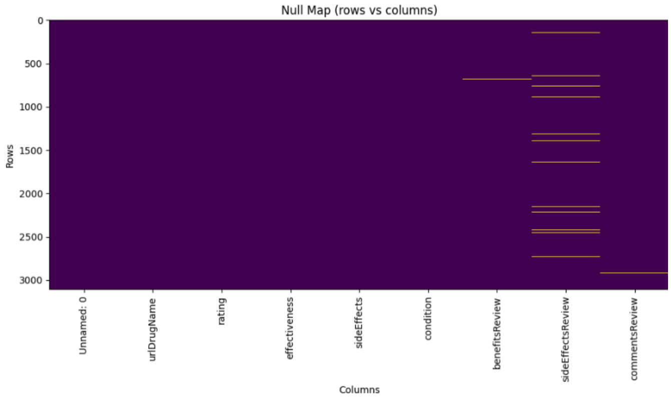


Figure 4: Null map for the Kaggle patient review dataset

3.5 ER Design and Relational Schema

The entity-relationship diagram in Figure 5 is reflected in the final database schema. The main entity is StandardDrug, which is a representation of the generic drug catalog that was obtained from the Mendeley dataset. It contains a foreign key to the Manufacturer entity along with attributes including generic name, packageMark, dose type, and strength. A single manufacturer may be linked to numerous StandardDrug records, and each manufacturer has a name and manufacturer identification. With characteristics like urlDrugName, rating, and efficacy, the BrandedDrug entity captures how medications appear in patient-facing data and reflects the Kaggle review side. BrandedDrug is connected to StandardDrug through the is brand for relationship, which models the soft foreign key: a branded name in the reviews is mapped to the most likely StandardDrug entry inferred by our pipeline, and this mapping is later stored in the unified review table as the standard_name column and its confidence score. The SideEffects entity stores side-effect information extracted from the review text, with attributes such as side-effect name and side-effect review, and is linked to BrandedDrug through the has relationship so that a single branded drug may have multiple associated side-effect records.

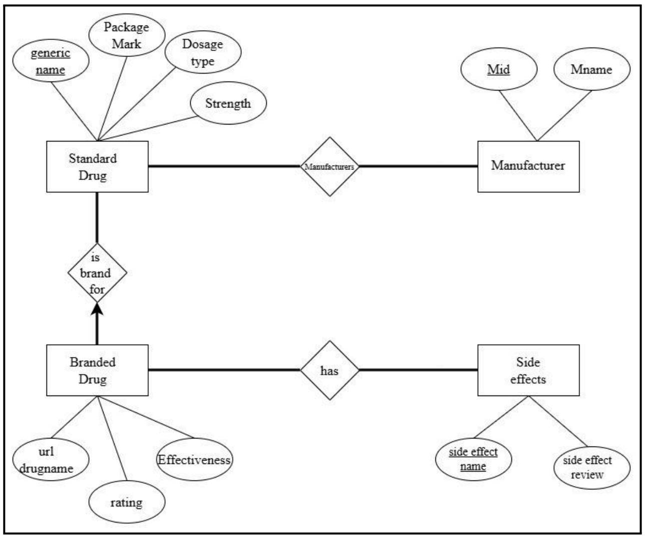


Figure 5: Entity–relationship diagram

4. Methodology

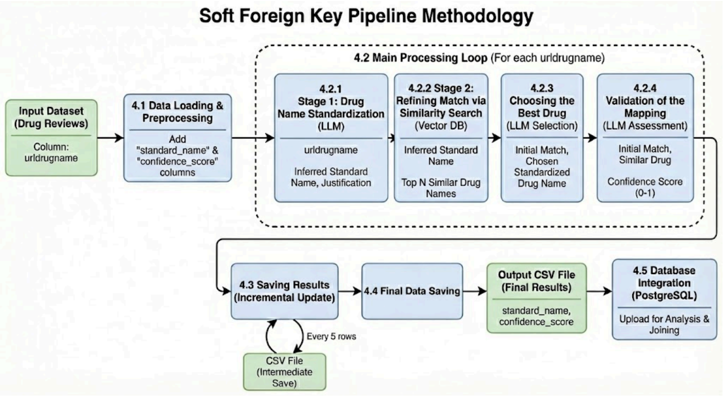

Soft Foreign Key Pipeline Methodology
4.2 Main Processing Loop (For each urldrugname)
Input Dataset (Drug Reviews)
Column: urldrugname
4.1 Data Loading & Preprocessing
Add "standard_name" & "confidence_score" columns
4.2.1 Stage 1: Drug Name Standardization (LLM)
urldrugname
Inferred Standard Name, Justification
4.2.2 Stage 2: Refining Match via Similarity Search (Vector DB)
Inferred Standard Name
Top N Similar Drug Names
4.2.3 Choosing the Best Drug (LLM Selection)
Initial Match, Chosen Standardized Drug Name
4.2.4 Validation of the Mapping (LLM Assessment)
Initial Match, Similar Drug
Confidence Score (0-1)
4.3 Saving Results (Incremental Update)
Every 5 rows
CSV File (Intermediate Save)
4.4 Final Data Saving
Output CSV File (Final Results)
standard_name, confidence_score
4.5 Database Integration (PostgreSQL)
Upload for Analysis & Joining

Figure 6: Pipeline of the proposed approach

The objective of this project is to build a "soft foreign key" pipeline that maps noisy drug names from patient reviews to standardized generic drug names using advanced language models (LLMs) and similarity-based retrieval techniques. The methodology involves four key steps: retrieving common drug names from the review dataset, validating them using a standardized drug catalog, refining the mapping through similarity searches, and storing the results along with a confidence score. Below is a detailed breakdown of the methodology. Figure 6 shows the pipeline of the proposed approach.

4.1 Data Loading and Preprocessing

The first step involves loading the drug review dataset, which contains a column urldrugname representing the common or brand drug names in patient reviews. The goal is to infer the standardized generic name for each drug name. We first ensure that the necessary columns (standard_name and confidence_score) are present in the dataset. These columns will store the standardized drug names and their respective confidence scores.

4.2 Main Processing Loop

For each urldrugname in the dataset, the goal is to map the common drug name to a standardized generic drug. The process is carried out through a series of steps, including querying an LLM and using vector-based similarity matching.

4.2.1 Stage 1: Drug Name Standardization

The get_standard_drug_name() function is used to send the common drug name to a pre-trained LLM. The LLM generates two outputs:

1. Standardized Drug Name: The generic name of the drug as inferred by the LLM.
2. Justification: The rationale behind why the LLM considers the inferred drug name correct.

The LLM prompt used for this task is designed to provide a standardized drug name in lowercase, along with an explanation of why that name was chosen.

The output is split into the standardized drug name and justification. If the output format is incorrect, a fallback is triggered.

4.2.2 Stage 2: Refining the Match via Similarity Search

Once a preliminary standardized drug name is identified, we further refine the match using vector-based similarity. The get_top_similar_words() function retrieves the top N most similar words from a pre-trained embedding model based on their semantic similarity to the initially identified drug name.

This step is critical for improving the accuracy of the mapping by leveraging the drug database's vectors and finding the closest matches.

4.3 Choosing the Best Drug from Similar Matches

After retrieving a list of similar drug names, the choose_best_drug() function selects the most appropriate drug name from the list based on the initial match. The model is prompted to evaluate the drug names based on their accuracy and provide a justification for the choice.

The output of this function is the chosen drug name along with the justification for why it was selected. If no suitable drug name is identified, the function returns None.

4.4 Validation of the Mapping

The next step involves validating the chosen standardized drug name using the validate_drug_mapping() function. This function sends a prompt to the LLM to assess how likely it is that the chosen drug name corresponds correctly to the common drug name. The model provides a confidence score ranging from 0 (completely incorrect) to 1 (completely correct).

The confidence score is extracted from the model's response, and it determines whether the drug name is valid enough to be recorded.

4.5 Saving Results

Once the standardized drug name and its associated confidence score have been determined, they are stored in the DataFrame. Every 5 rows, the updated DataFrame is written to a CSV file to ensure that progress is saved incrementally, and no data is lost.

4.6 Final Data Saving

Once all rows have been processed, the final results are written to the output CSV file, which contains the standardized drug names and their confidence scores.

4.7 Database Integration

The results stored in the output CSV can be uploaded to a PostgreSQL database, where the standardized drug names (soft foreign keys) can be used for further analysis and joined with the original drug reviews to perform aggregated queries, such as side effect analysis and patient feedback summaries.

5. Results and Analysis

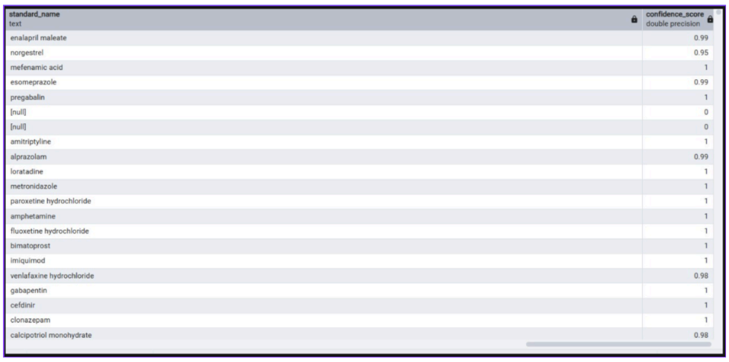
5.1 Soft Foreign Key Mapping Results

The proposed three-stage foreign key pipeline was applied to the Kaggle Patient Ratings dataset to infer standardized generic drugs names for noisy ,user-provided identifiers. The resulting unified table contains 3,107 review records and 10 attributes, including the inferred standard_names and an associated confidence_score. The standard_name column functions as a probabilistic foreign key that enables join between the patient review data and the standardized generic drug catalog.

A substantial proportion of review entries were successfully mapped to a corresponding generic drug with confidence scores exceeding the predefined acceptance threshold of 0.5. For these entries, the inferred standard_name can be reliably joined with the reference table, allowing aggregation and analysis at the generic-drug level. Review records with confidence scores below the threshold were intentionally left unmapped to prevent erroneous joins and to preserve uncertainty in the data integration process.

These results demonstrate that the proposed approach can approximate referential integrity in scenarios where exact string matching fails due to inconsistent or informal naming conventions.

5.2 Confidence Score Distribution and Mapping Quality



standard_name text	confidence_score double precision
enalapril maleate	0.99
norgestrel	0.95
mefenamic acid	1
esomeprazole	0.99
pregabalin	1
[null]	0
[null]	0
amitriptyline	1
alprazolam	0.99
loratadine	1
metronidazole	1
paroxetine hydrochloride	1
amphetamine	1
fluoxetine hydrochloride	1
bimatoprost	1
imiquimod	1
venlafaxine hydrochloride	0.98
gabapentin	1
cefdinir	1
clonazepam	1
calcipotriol monohydrate	0.98

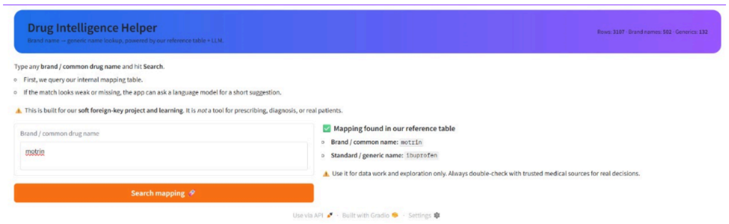
Figure 7: Excerpt from the unified SQL table showing standardized generic names and associated confidence scores.

The distribution of confidence scores indicates that most accepted mappings fall within a high-confidence range,with many values between 0.95 and 1.00. These cases typically correspond to well-known brand-to-generic relationships, where semantic similarity and pharmacological knowledge align strongly. Examples include mappings such as Xanax to alprazolam and Neurontin to gabapentin. Figure 7 is an excerpt from the unified SQL table showing standardized generic names and associated confidence scores.

Mappings with intermediate confidence scores (approximately 0.50-0.70) were primarily associated with ambiguous drug names, overlapping therapeutic classes, or incomplete reference entries. The inclusion of a final LLM-based validation step proved critical in filtering these borderline cases, reducing the likelihood of incorrect foreign key assignments.

5.3 System Validation via LLM Interface

To demonstrate end-to-end system behavior, the unified dataset was exposed through an LLM-powered interface.The system first queries the internal soft foreign key table and returns the standardized generic name when a high-confidence mapping exists. For low-confidence or missing mappings, the interface transparently falls back to general pharmacological knowledge and explicitly indicates the source of the response.



Drug Intelligence Helper

Brand name → generic name lookup, powered by our reference table + LLM.

Rows: 3107 · Brand names: 502 · Generics: 132

Type any **brand / common drug name** and hit **Search**.

- First, we query our internal mapping table.
- If the match looks weak or missing, the app can ask a language model for a short suggestion.

⚠ This is built for our **soft foreign-key project and learning**. It is *not* a tool for prescribing, diagnosis, or real patients.

Brand / common drug name: motrin

✅ **Mapping found in our reference table**

- **Brand / common name:** `motrin`
- **Standard / generic name:** `ibuprofen`

⚠ Use it for data work and exploration only. Always double-check with trusted medical sources for real decisions.

Figure 8: LLM-powered chatbot interface demonstrating soft foreign key lookup and explanation.

Figure 8 shows the user interface. This layered design emphasizes transparency and reinforces the role of soft foreign keys as probabilistic links intended for exploratory analysis rather than authoritative clinical use.

5.4 Limitations and Error Analysis

Several limitations were identified during evaluation. Some brand names present in patient reviews were absent from the reference catalog, resulting in unavoidable unmapped records.

In addition, drugs with similar names or multiple formulations occasionally produced ambiguous similarity scores, requiring conservative rejection to avoid incorrect links.

The current implementation focuses primarily on name normalization and does not explicitly account for dosage strength or formulation,which may be clinically relevant. Incorporating these attributes remains an important direction for future work.

5.5 Implications for Database Systems

The results highlight a common data integration challenge in which strict foreign key constraints are impractical due to data quality issues.The proposed soft foreign key approach demonstrates how probabilistic mappings,supported by LLMs and vector similarity search, can extend relational schemas without abandoning core database principles.

Overall, this work illustrates how modern AI techniques can complement relational databases by enabling approximate referential integrity while maintaining transparency, queryability and analytical rigor

6. Conclusion

From this project we were able to successfully integrate real-world drug review data with structured generic drug information and draw meaningful insights about how AI driven soft matching can support healthcare data systems. We observed that even though review data is noisy, inconsistent, and entirely user generated, modern embedding methods and large language models still allow us to build reliable connections between unstructured text and standardized drug records. It is also interesting to note that despite the complexity of pharmaceutical terminology and the wide variation in brand and generic names, the system consistently identified accurate generic mappings with strong similarity scores and confident LLM validation.

- Soft matching using embeddings turned out to be a very effective substitute when traditional foreign keys are missing.
- The unified dataset made it easy to explore conditions, ratings, and patterns in user experiences.
- LLM based confidence scoring added an extra check that helped us avoid unclear or incorrect mappings.
- Once the data was loaded into the DBMS, the entire ETL pipeline became straightforward to automate and extend for future updates.

7. Future work

Several improvements could make this system more robust and practical. A natural extension is to include dosage information and support queries involving multiple drugs, which would allow the system to check for possible interactions rather than treating each drug in isolation. And adding a simple feedback loop where users can mark incorrect mappings would also help the model refine its decisions over time.

Using a larger and more diverse review dataset would likely improve both the embedding quality and the overall confidence of the mappings. Integrating the LLM earlier in the pipeline, rather than at the end, could lead to a more consistent mapping strategy. Better logging of mapping failures would also make the system easier to monitor and improve.

Finally, reorganizing the database for scalability and moving all credentials into secure environment variables would make the pipeline more suitable for deployment and long-term use.

References

[1] Md Mahmudur Rahman, Md M KHAN. (2024). Medicinal Products in Bangladesh. Mendeley https://data.mendeley.com/datasets/zhtvkny53n/1

[2]https://www.kaggle.com/datasets/rabieelkharoua/patient-ratings-identifying-best-drugs

[3] Dobbins NJ. Generalizable and scalable multistage biomedical concept normalization leveraging large language models. Res Synth Methods. 2025 Mar 12;16(3):479–90. doi: 10.1017/rsm.2025.9. PMCID: PMC12527512.

[4] Ornstein JT. Probabilistic Record Linkage Using Pretrained Text Embeddings. Political Analysis. Published online 2025:1-12. doi:10.1017/pan.2025.10016.