2425-2A-DS 54 submission

Edition: 2024 2A
Project: ALBUM
Primary Topic: DINT
Secondary Topic:
Members: Miglena Pavlova, Gabriela Todorova
Last update: 30/04/2025, 02:04:18

Motivation

In large music databases, the same album may appear multiple times under different names, spellings, or formats. This problem, known as semantic duplication, significantly affects user experience, catalog management, and data analytics. For example, an album might be listed as "Taylor Swift - Red" and again as "Taylr Swift - RED (Deluxe)", leading to fragmented data and misleading insights. Deduplication improves the clarity, reliability, and usability of music metadata. This project is important for both industry and research. In business, clean metadata enhances search results, recommendations, and content aggregation. In research, it contributes to the field of data integration and entity resolution, which are critical for combining heterogeneous datasets. By leveraging string similarity techniques and rule-based logic, this project offers a structured, repeatable method for resolving duplicate records in music databases.

(Business/Research) questions

1. How can we automatically detect and flag duplicate music album entries with inconsistent metadata?
2. How well does our model perform compared to the provided ground truth duplicates? These questions follow the SMART criteria by being:

- Specific: Focused on deduplication of music albums.
- Measurable: Evaluated using precision, recall, and F1 score.
- Achievable: Implemented with string similarity libraries and thresholding logic.
- Relevant: Addresses practical metadata issues in music databases.
- Time-bound: Conducted and evaluated within the project timeline.

Source data

The dataset used was cddb_discs.xml, a large XML-formatted music album collection. It presented various data quality challenges that required careful handling. Many records contained typographical errors, such as misspellings in artist and album names. Inconsistent casing, punctuation, and spacing further complicated the matching process. Additionally, a number of entries had incomplete metadata, with missing IDs, artist fields, or tracklists. Another issue was the variability in track listings—some albums had reordered or partially listed tracks, which reduced the reliability of simple comparisons. Furthermore, inconsistencies in the ID fields emerged, with some discs using a cid field instead of the standard id.
To address these problems, we implemented fallback parsing logic to handle alternative ID fields and standardized the data using lowercase transformation and punctuation normalization. Entries missing key fields were excluded from scoring to maintain the accuracy of the similarity model. Track list inconsistencies were managed using Jaccard similarity, which compares sets rather than sequence, allowing it to tolerate reordering and partial overlaps. Evaluation was supported by a separate file, cddb_9763_dups.xml, which provided ground truth duplicate pairs.

Method

The methodology followed a structured pipeline beginning with XML data parsing to extract each disc's metadata including artist, title, and tracks. Next, a preprocessing phase standardized the data by

converting text to lowercase and removing punctuation and excess spacing. Once the data was normalized, similarity scores were computed between every possible pair of albums using attribute-specific methods. For artist and title fields, Jaro-Winkler similarity was applied due to its effectiveness in handling minor spelling errors and character transpositions. For track lists, Jaccard similarity was used to assess the overlap of track titles, treating them as sets to account for order and minor inconsistencies.

A composite similarity score was then calculated for each pair using a weighted average of the individual attribute scores. The weights were assigned as follows: 40% to artist similarity, 40% to album title similarity, and 20% to track list similarity. These weights were selected based on the perceived reliability and discriminative power of each attribute in uniquely identifying an album. To classify pairs, a dual-threshold system was used: album pairs with a similarity score above 0.65 were labeled as matches.

The validation of this method involved comparing predicted matches to the ground truth dataset using standard evaluation metrics. Precision, recall, and F1-score were calculated using binary-averaging from Scikit-learn's metrics module. This allowed for an objective assessment of the model's accuracy and robustness.

Results

Metric	Value
Precision	0.99
Recall	0.93
F1 Score	0.96
Duplicates Predicted	280
Ground Truth Pairs	298

The model achieved a precision of 0.99, meaning that nearly all duplicates identified were correct. This high precision is significant as it minimizes false positives, ensuring that matches recommended for merging are indeed valid. The recall was 0.93, indicating that approximately 93% of all true duplicates in the ground truth dataset were successfully identified. The resulting F1 score of 0.96 reflects a balanced and effective performance in reconciling the trade-off between precision and recall. These results validate the effectiveness of the weighted scoring strategy. While the model slightly favors precision over recall, this trade-off was intentional to prioritize confidence in match decisions. Additional improvements could be achieved by refining preprocessing steps and considering more advanced similarity measures or machine learning approaches. The visual representations used, such as the pipeline flowchart and attribute weighting bar chart, further helped to communicate the logic and results of our approach.

Reliability of results

The results presented by the deduplication model are highly reliable, particularly due to its near-perfect precision. This makes it well-suited for applications where incorrect merges must be avoided. However, the recall rate indicates that there is still room for improvement in capturing all valid duplicates, especially those that differ significantly in metadata formatting or language. The thresholds used for classification were conservatively selected to favor high precision. The choice of Jaro-Winkler and Jaccard as similarity functions adds to the model's robustness, as these techniques are known for handling minor text inconsistencies and unordered lists effectively.

Technical depth

This project integrated several technical methods beyond basic entity matching. We utilized the jellyfish library for string similarity calculations, particularly Jaro-Winkler for its ability to handle common textual errors in names. For evaluating the track lists, Jaccard similarity was implemented to assess the overlap between unordered sets of strings. XML parsing and data structuring were carried out using Python's xml.dom.minidom, while performance validation was conducted using scikit-learn's evaluation metrics. In addition to implementing existing techniques, the project also introduced a composite scoring

mechanism that balanced multiple attribute similarities through weighted averages. And lastly, we adjusted the threshold to get the optimal results.

Conclusions & recommendations

The deduplication system developed in this project effectively identifies duplicate music album records within a complex and noisy dataset. With a precision of 99% and a balanced F1 score, the method offers a practical solution for metadata cleaning in music databases. Stakeholders such as music platform engineers, catalog managers, and data curators can benefit from the increased accuracy and consistency in metadata that this approach enables. It is recommended to implement an automated merging system for records classified as high-confidence duplicates. Preprocessing can also be enhanced with more advanced techniques, such as regularizing abbreviations, removing special characters, and applying language normalization. Future work may involve applying a probabilistic approach, where instead of strict yes-or-no decisions, a confidence score is assigned to each potential match. This will help identify duplicates even when the match is less obvious, improving coverage and adaptability.

Reflection

The project presented several real-world data challenges, especially in dealing with inconsistent and incomplete metadata. These challenges were addressed through iterative development, experimentation with multiple similarity functions, and continuous evaluation against ground truth data. The hands-on application of concepts from data integration and entity resolution proved crucial in successfully navigating these difficulties. Skills learned in the course, such as similarity metrics and schema matching, directly supported the implementation. However, additional knowledge in natural language processing, particularly in named entity recognition or string embeddings, could have improved performance further. ChatGPT was used to clarify technical concepts.

Handling of Schema Differences, Data Inconsistencies, and Redundancies

To deal with schema differences in the CDDB dataset, we implemented a logic in our XML parsing function. This logic first checks for the standard 'id' field and, if it's missing, looks for the alternative 'cid' field instead. We used the same approach when parsing the ground truth file, ensuring consistency in how we identified and compared albums across both datasets. Beyond structure, the data itself often came with inconsistencies artist names and album titles varied in capitalization, punctuation, and spelling. To minimize the impact of these differences, we normalized all strings to lowercase and stripped punctuation before performing comparisons. For the actual similarity calculations, we used Jaro-Winkler similarity for artist and title fields, which is particularly effective at recognizing minor typos or transpositions. For track lists, we used Jaccard similarity, which compares sets rather than sequences, making it naturally tolerant to changes in order or incomplete lists. We also treated track lists as sets to eliminate duplicate tracks within a single album, reducing noise during similarity computation. Finally, to avoid recording the same match twice, we stored all matched disc pairs as sorted tuples in a set, ensuring (A, B) and (B, A) were treated as one. Together, these steps allowed us to manage schema variations, clean up inconsistent metadata, and prevent

Challenges and Lessons Learned

The code explicitly demonstrated the challenge of establishing an optimal similarity threshold, which we initially considered at 0.8 but ultimately chosen as 0.65. Deciding on this threshold required iterative adjustments to find the best trade-off between precision and recall. Additionally, the lesson about balancing evaluation metrics became evident through the implementation of the precision, recall, F1-score, and accuracy calculations, which were critical in objectively evaluating our deduplication results. These metrics informed us about the model’s strengths and areas needing improvement, highlighting the necessity of thorough performance validation.

References

Christen, P. (2012). Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.
Naumann, F. (2002). Data Fusion and Data Quality.
CDDB Dataset: http://hpi.de/naumann/projects/repeatability/datasets/cd-datasets.html
Scikit-learn: https://scikit-learn.org
Jellyfish Library: https://github.com/jamesturk/jellyfish