Suggestions for Improving or Scaling the Solution

Objective

While the current solution provides accurate normalization with a combination of rule-based preprocessing, spaCy NER, and heuristic approaches, there are opportunities for improvement and scaling. These suggestions aim to enhance the pipeline's accuracy, efficiency, and robustness for larger datasets and real-world use cases.

1. Improving Accuracy

a. Fine-Tune spaCy NER Models

- **Why**: The pre-trained multilingual model (xx_ent_wiki_sm) is general-purpose and not optimized for the music publishing domain.
- **How**:
    - Training or fine-tuning a custom spaCy model on a domain-specific dataset with labeled entities (e.g., PERSON, ORG, GPE).
    - Including examples of ambiguous names, publishers, and multilingual names in the training data.
    - Validating the model's performance on edge cases such as single-word names and non-Latin characters.
- **Potential benefit**: Improved entity recognition accuracy tailored to the specific requirements of the dataset.

b. Supervised Learning for Normalization

- **Why**: Rule-based systems can struggle with edge cases and ambiguous data.
- **How**:
    - Creating a labeled dataset of raw inputs and normalized outputs.
    - Training a supervised model (e.g., BERT) for sequence-to-sequence tasks or classification to predict normalized structures.
    - Using the labeled dataset to identify patterns and handle cases where spaCy struggles, such as "GC/KIZO" or "UNKNOWN WRITER."
- **Benefit**: A supervised approach would generalize better to unseen data and adapt to complex patterns.

c. Expanded Exclusion List

- **Why**: Some irrelevant terms (e.g., "COPYRIGHT CONTROL") may still slip through the current exclusion dictionary.

- **How**:
    - Iteratively expanding the exclusion dictionary by analyzing low Jaccard similarity rows and identifying recurring patterns of noise.
    - Using regex patterns for more flexible matching of ambiguous keywords.
- **Benefit**: Reduces false positives and improves overall accuracy.

d. Improve Handling of Multilingual Data

- **Why**: Names in non-Latin scripts or with diacritical marks may not be handled consistently.
- **How**:
    - Incorporating transliteration libraries (e.g., unidecode) to normalize names across different scripts.
    - Using pre-trained multilingual embeddings like mBERT to enhance the understanding of non-Latin names.
- **Benefit**: Ensures consistent handling of names in different languages and scripts.

2. Scaling the Solution

a. Cloud Deployment

- **Why**: For real-time or large-scale use cases, deploying the solution in the cloud ensures scalability and accessibility.
- **How**:
    - Containerizing the pipeline using Docker.
    - Deploying the solution on cloud platforms (e.g., AWS, GCP, Azure) with serverless architectures like AWS Lambda for on-demand processing.
    - Using APIs to enable easy integration with other applications.
- **Benefit**: Enables real-time normalization and seamless integration into music publishing workflows.

b. Automated Feedback Loop

- **Why**: Real-world data may contain new patterns or edge cases not covered in the training or exclusion rules.
- **How**:
    - Logging rows with low Jaccard similarity or high uncertainty scores for manual review.
    - Periodically retraining models or refine exclusion rules based on logged cases.

- **Benefit**: Continuously improves the pipeline's performance and adaptability to new data.

3. Enhancing Robustness

a. Fallback Mechanisms

- **Why**: Some names may not be recognized by spaCy or the heuristics, leading to empty normalized outputs.
- **How**:
    - Adding a fallback mechanism to retain the raw input if no valid segments are identified.
    - Using language-specific heuristics for ambiguous cases (e.g., single-word names in non-Latin scripts).
- **Benefit**: Reduces the number of empty outputs and ensures no data is entirely discarded.

b. Confidence Scoring

- **Why**: Not all normalized outputs are equally reliable.
- **How**:
    - Assigning confidence scores to each normalized result based on spaCy's NER probabilities and heuristic rules.
    - Highlighting low-confidence rows for manual review or secondary processing.
- **Benefit**: Improves transparency and allows stakeholders to focus on critical cases.

4. Advanced Techniques

a. Integration with Large Language Models (LLMs)

- **Why**: Pre-trained LLMs like OpenAI's GPT-4 or HuggingFace transformers can better understand contextual nuances in text.
- **How**:
    - Using a fine-tuned LLM for contextual understanding and normalization of raw text.
    - Training the model on domain-specific data for improved performance.
- **Benefit**: Superior handling of ambiguous cases, multilingual data, and complex patterns.

Summary of Recommendations

Immediate Enhancements

- Expand exclusion rules for improved filtering.
- Integrate fallback mechanisms to reduce empty outputs.
- Use confidence scoring to prioritize manual review of low-confidence results.

Medium-Term Improvements

- Fine-tune spaCy models on domain-specific data.
- Introduce supervised learning for normalization tasks.

Long-Term Scaling

- Deploy the solution on cloud platforms for scalability.
- Incorporate LLMs for advanced contextual understanding.
- Develop a feedback loop for continuous improvement.