Task 1 - Text Normalization

Methodology

The unique difficulties presented by composer name normalization led to the adoption of a hybrid strategy that combines rule-based normalization with a deep learning model. Rule-based approaches struggle with invisible variations but provide excellent precision in clearly defined circumstances. Although they are excellent at generalization, deep learning models—especially transformer-based models like T5—can produce inconsistencies. Thus, T5 was chosen due to its efficacy in text-to-text conversions, which makes it especially appropriate for jobs involving sequence-based normalization. A hybrid approach can lead to well-defined instances being handled deterministically while permitting flexibility for intricate or invisible modifications, given the structured yet variable nature of composer names.

Data Preprocessing & Normalization

The raw dataset consists of composer names, often containing irregular formatting, missing values, and special characters. **Encoding correction** is performed using the “ftfy” module in Python, to ensure that all characters are correctly represented, mitigating issues arising from different encoding standards. **Noise removal** is then applied to eliminate extraneous symbols and artifacts such as publisher information or copyright attributions (e.g., "Copyright Control"). This step ensures that the model focuses exclusively on the name components. Then, **normalization rules** were applied to standardize name formats, including converting "Last Name, First Name" structures into a uniform "First Name Last Name" format. Lastly, a **data augmentation** method was utilized to introduce controlled variability through synonym replacement and noise injection, enhancing the model’s robustness against minor perturbations in input data.

Moreover, in this step, regular expression rules were utilized to strip metadata (e.g., ASCAP info) and standardize delimiters (slashes/commas) into a uniform format. This reduces input ambiguity and ensures consistent, clean data for the transformer model. In addition, a fine-tuned T5-small model was used to normalize the composer names, training it on paired raw-to-standardized name data. The T5Predictor, combined with T5Tokenizer, encodes and decodes sequences.

Evaluation Metrics

To evaluate the model's performance four complementary metrics were used:

- **BLEU Score** measures how well the model preserves word sequences compared to ground truth names.
- **Character-Level Accuracy** provides a detailed assessment by calculating the percentage of correctly predicted characters.
- **Word-Level Accuracy** evaluates the model's performance at the word level, measuring the percentage of correctly predicted words.
- **Normalized Edit Distance** calculates the minimum number of operations (insertions, deletions, or substitutions) needed to transform the predicted output into the reference name.

Metric	Score
BLEU Score	0.2090
Character-Level Accuracy	48.57%
Word-Level Accuracy	45.00%
Normalized Edit Distance	0.3460

Based on the given table, the observations show that the model performs well on simple name pairs (e.g., "Ricky Wilde/Kim Wilde") but struggles with complex entries, special characters, and multi-name formats, leading to inconsistent outputs. Repetitions like "Kor Kor" indicate overfitting to specific patterns, while special character preservation remains unreliable.

Improvement Recommendations and Future Directions

1. Data Quality Enhancement
The dataset will be expanded to include a broader range of special characters, diverse name formats, and varied delimiters to improve generalization capabilities. Synthetic data augmentation techniques will be employed to enhance robustness against noisy inputs [[19]]. Additionally, rigorous curation of training pairs will ensure high-quality ground truth labels, reducing noise and ambiguity during model training.

2. Model Architecture Refinements
Larger transformer-based models, such as T5-base or T5-large, will be fine-tuned to capture more complex dependencies and improve overall performance. Character-level embeddings will be integrated to better handle special characters and diacritics. Furthermore, the sequence-to-sequence loss function will be refined to mitigate repetitive outputs and improve generation quality.

3. Evaluation Framework Enhancements
Domain-specific metrics will be introduced to complement traditional evaluation measures like BLEU and accuracy, enabling a more nuanced assessment of normalization performance. Detailed error analysis will be conducted, particularly focusing on edge cases such as abbreviation mismatches and multi-name entries, to inform targeted improvements. Additionally, the model's generalization capability will be rigorously tested on out-of-distribution samples to ensure robustness across diverse input types].

4. Methodological Trade-offs
A hybrid approach combining rule-based methods and deep learning will be further optimized to balance deterministic precision with flexible generalization. While rule-based systems provide high accuracy for well-defined patterns, neural models excel at handling variability but risk-generating hallucinated text. Larger models, such as T5-large, offer improved accuracy but incur higher computational costs. Beam search decoding will be fine-tuned to balance output quality and inference latency.

5. Future Work
Future efforts will prioritize expanding the dataset to include rare name structures and non-Latin scripts, enabling broader applicability. Lightweight architectures will be explored to optimize computational efficiency without sacrificing performance. Post-processing pipelines will be strengthened to address residual formatting inconsistencies, ensuring reliable outputs in real-world scenarios.

References

[1] Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. Journal of big Data, 8(1), 101.

[2] Li, B., Hou, Y., & Che, W. (2022). Data augmentation approaches in natural language processing: A survey. Ai Open, 3, 71-90.

[3] Satapathy, R., Li, Y., Cavallari, S., & Cambria, E. (2019, July). Seq2seq deep learning models for microtext normalization. In 2019 international joint conference on neural networks (IJCNN) (pp. 1-8). IEEE.

[4] Rohatgi, S., & Zare, M. (2017). DeepNorm-A Deep learning approach to Text Normalization. ACM ISBN, 123-4567.

[5] Zhang, Y., Bartley, T. M., Graterol-Fuenmayor, M., Lavrukhin, V., Bakhturina, E., & Ginsburg, B. (2024, April). A chat about boring problems: Studying gpt-based text normalization. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 10921-10925). IEEE.

[6] Tiwari, A. S., & Naskar, S. K. (2017, December). Normalization of social media text using deep neural networks. In Proceedings of the 14th international conference on natural language processing (ICON-2017) (pp. 312-321).

[7] Aliero, A. A., Bashir, S. A., Aliyu, H. O., Tafida, A. G., Bashar, U. K., & Nasiru, M. D. (2023). Systematic review on text normalization techniques and its approach to non-standard words.