ICMLM
A Language Model For Indian Classical Music

Swagnik Roychoudhury
New York University, College of Arts and Sciences
Courant Institute of Mathematical Sciences, Center for Data Science

NYU COLLEGE OF ARTS & SCIENCE
ENSURE RESEARCH GROUP
Energy-Aware, Secure and Reliable Computing

INTRODUCTION

- The advancement of artificial intelligence in music composition has seen remarkable developments, notably with initiatives like Google's MusicLM[1]
- However, these models fall short when applied to the unique structure of Indian Classical Music (ICM)
- Music consists of three parts: Rhythm, Melody, and Harmony[2]. A distinctive characteristic of ICM is its emphasis on rhythm over melody (Fig. 1). In ICM, rhythm dictates everything from instrumental patterns to dance movements and the conveyed emotions. As a result, ICM pieces maintain their integrity in the absence of melody
- Existing music generation models like MusicLM focus on on high level melody generation, and so struggle to capture the rhythmic intricacies of ICM.
- Additionally, due to ICM's oral nature, existing formal music notation systems, such as the Bhatkande[3] system, are not sufficiently defined for sequential computation
- Hence, we propose ICMLM, a group of models that generate rhythmic compositions at a syllable/word level. Furthermore, we have developed our own orthography and built a computer-compatible dataset with over 100 compositions

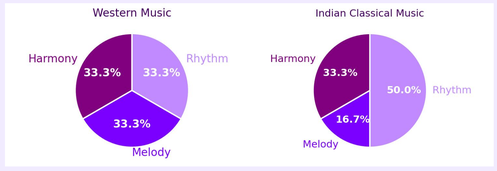

Fig. 1: An empirical distribution of Western Music and ICM

A CUSTOM ORTHOGRAPHY

- The first step to building our orthography was to establish a vocabulary. We did this by identifying the 16 most common words, such as *dha, ta, dhin* etc.
- Additionally, we added a space token, represented by the syllable '*b*', as well as an <EoS> (end of sentence) token, represented by '*s*'
- Each composition is made of $yx+1$ tokens, where $1 \leq x \leq 4$ is the number of measures and is the size of each measure in beats. For our experiments, all of our compositions follow a $y=16$ beat length measure
- Every composition ends with the <EoS> token
- With the help of ICM experts, we converted over a hundred compositions to this new notation
- Finally, we developed a short parser to convert our notation into TaalMaata[4], a semi-sequential notation used to digitally play written compositions to audibly analyze input/output compositions

ICMSLM

- ICMSLM (Small Language Model) is a grassroots level recurrent sequence architecture that uses our own propagation and optimization functions
- The model's parameters (i.e. size and state) are initialized, and the data is loaded and tokenized. In each training epoch we propagate forward and backward, clip the gradients, and update the RNN cell's weights.
- A grid search for hyperparameter tuning yielded an optimal learning rate of 0.01 and an epoch count of 100.
- The model successfully learned popular word phrases, such as "*Dha b ghe b na b*" and "*ti ra ki ta*"
- However, the model did not perform well joining phrases into compositions, often resulting in awkward pauses or speedups. In some cases, the model abused the gap token by repeated generation.
- Sample Compositions:

ta dha b ti ra ki ta dha b ti ra ki ta dha b ghe b tin b na ra ki ta dha b ti ra ki ta dha b ti s

dha b ti b ta b dha b ti b ta b dha b ke b ti b ta b dha b ti ra ki ta dha b ti b ta b s

IMAGE OF MODEL ARCHITECTURE HERE

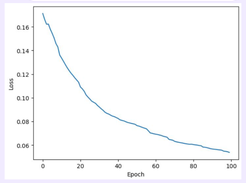

Fig. 2: SLM Loss over Time

ICMMLM

- ICMMLM (Medium Language Model) is a more complex LSTM (Long Short-Term Memory Model), using PyTorch to build a transformer with multiheaded attention[5]
- The data is loaded, tokenized, and left-padded. The data goes through embedding, encoding,and attention layers.
- We compute the loss with the cross-entropy function, backpropagate, and finally update the LSTM with an Adam optimizer. This process is repeated for all epochs
- We performed a grid search, yielding a learning rate of 0.001, epoch count of 120, batch size of 16, 4 attention heads, 3 fully-connected layers, and a 0.1 dropout.
- The compositions generated by the model displayed a greatly improved understanding of phrase structuring, as it was able to join phrases without disrupting the flow of the composition
- One drawback we noticed is that the model sometimes repeated chunks from training data
- We attribute this to the small size of the dataset, and we hope to address this in the future
- Sample Compositions:

dha b ti b b b dha b ti b b b dha b b b dha b dha b ti ra ki ta dha b dha b ti ra ki ta dha b ghe b tin b na b ki b na b ghe b ghe b dha b ghe b dhin b ghe b na b na b ghe b na b s

IMAGE OF MODEL ARCHITECTURE HERE

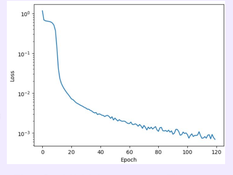

Fig. 3: MLM Loss over Time

FOUNDATION MODELS

- In addition to in-house LSTM architectures, we tested foundation models on their ability to generate ICM compositions. We asked GPT3.5 and LLAMA 2 to generate any ICM composition.
- GPT gave very generic and poorly written compositions, while LLAMA provided nothing
- We set up a finetuning pipeline for both foundation models. With GPT, we achieved a desirable loss threshold after about 50 epochs
- LLAMA was more complicated. Some tokens in our dictionary already existed in LLAMA's tokenizer, while others didn't, causing possible imbalances
- We came up with two approaches to address this
- 1) We randomly generate tokens that aren't in LLAMA's dictionary, and then map our tokens that do exist in LLAMA to a non-existent token. This gives all of our tokens a fresh start
- 2) We tokenize the dataset by letters instead of words, since all 26 letters are in LLAMA's tokenizer. This way, all tokens have pre-trained embeddings

FUTURE WORK

- We aim to expand the models to include taals, or time signatures, (other than the 16 beat taal, called *teentaal*, used in our experiments) such as *Roopak* (7), *Ektaal* (12), and *Jhaaptaal* (10)
- This would require an additional semantic layer to understand the correlation between taals and size

REFERENCES

[1] Agostinelli, Andrea, et al. "Musiclm: Generating music from text." arXiv preprint arXiv:2301.11325 (2023).
[2] Peter Yarrow "The Basics of Harmony, Melody, and Rhythm." Issue, 9 Mar. 2023, issuu.com/peteryarrow/docs/the_basics_of_harmony
[3] "Notating Indian Classical Music." Notating Indian Classical Music - Raag Hindustani, raag-hindustani.com/Notation.html.
[4] TaalMala.com, "Tabla, Pakhawaj, Manjeera, Tanpura, Swarmandal, Harmonium and Santoor Accompaniment and Composition." TaalMala, www.taalmala.com/help.shtml.
[5] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

ACKNOWLEDGEMENTS

- Special thanks to my research mentor Akshaj Kumar Veldanda, my research advisor Dr. Siddharth Garg, and CAS Assistant Dean Brendan Sullivan
- Funding and Support from NYU's ENSURE group, NYU DURF Grant, and Tarana Dance Academy of NJ