Learning / research algorithms

We started with researching different machine learning solutions for predicting the next item in a sequence. Finite-state machine was mentioned for algorithm design and implementation for deliverable 3 and other machine learning applications such as Recurrent Neural Network (RNN), LSTM NN's, Recommender systems, and language model computation by faculty and industry experts. However, since we do not have a large enough dataset to accurately train traditional machine learning models, we chose a simple model we could that could go further in predicting the future.

We considered evaluating the distance, hierarchies, patterns between Roman numerals, and levels of harmonic and theoretic complexity in the beginning stage of building a prototype of the algorithm.

References:

Introduction to Pattern Recognition (Tutorial) and Machine Learning:

Great Learning. (2021, November 24). Pattern recognition (tutorial) and Machine Learning: An introduction. Retrieved from https://www.mygreatlearning.com/blog/pattern-recognition-machine-learning/

Introduction to Sequence Modeling Problems:

Kumar, N. (2019, December 17). Introduction to sequence modeling problems. Retrieved December 03, 2021, from https://towardsdatascience.com/introduction-to-sequence-modeling-problems-665817b7e583

Recurrent Neural Network (RNN)

In the beginning part of the research process, Recurrent Neural Networks (RNN) was one of the methods that we kept seeing. We have decided to look for other machine learning solutions as we did not have a large set of data.

The Basics of RNN:

Phi, M. (2020, June 28). Illustrated Guide to Recurrent Neural Networks. Retrieved December 03, 2021, from https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

General Tensorflow tutorials:

https://www.tensorflow.org/tutorials

Recurrent Neural Network in Tensorflow:

https://www.tensorflow.org/guide/keras/rnn

Recommender Systems

We were introduced with the concept of recommender systems during the midterm presentations by faculty as it could predict recommended progression to add or build on top of. We decided to pass on recommender systems as the language modeling seemed more suitable for building chord progressions over time based on deliverable3 (corpus of digitally represented songs).

Introduction to recommender systems:

Rocca, B. (2019, June 12). Introduction to recommender systems. Retrieved December 03, 2021, from https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

N-gram model using Markov chain

We chose n-gram model using Markov chain because it was the simplest model that could give us probabilities of the sequences of words, and it has the fundamental and ground-level concepts of language modeling.

Language Models N-Gram:

Kapadia, S. (2019, August 19). *Language Models: N-Gram*. Medium. http://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9

Introduction to N-Grams:

Kumar, Prachi, et al. (2017, October 21). An Introduction to N-Grams: What Are They and Why Do We Need Them? https://blog.xrds.acm.org/2017/10/introduction-n-grams-need/