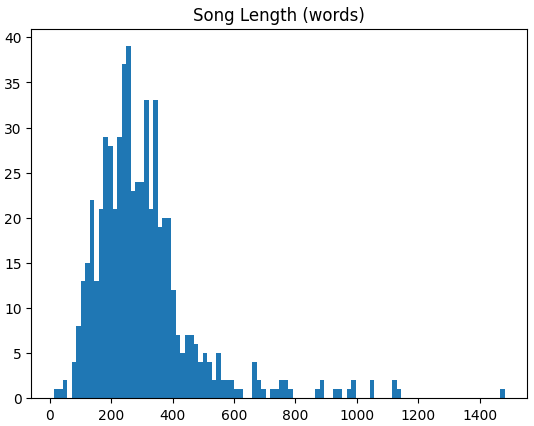
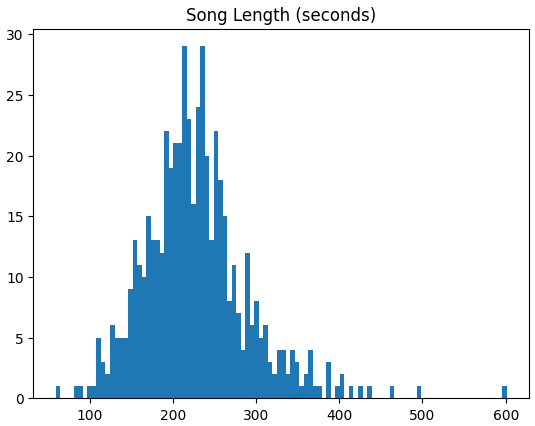
# Deep Learning Assignment 3

Yonatan Amaru – 316365998  
Noam Munz – 207042292

## Data Analysis

* The data in this assignment includes 599 songs, and their respective melodies in MID files.
* Data came divided into 594 train samples and 5 test samples.





* We can see that there’s some correlation between song length in words and in seconds. We’ll be using these when vectorizing melodies.

## Preprocessing

Lyrics preprocessing:

* For the lyrics preprocessing we used GoogleNews word2vec model with vocabulary size of 3,000,000.
* Each song was divided into sequences (dynamic size) and each word in sequence was converted into a vector (punctuation, stop words and words not in vocabulary were removed).
* For targets, each sequence word was designated the class of its following word according to the vocabulary index.

Melody preprocessing:

* For melodies, we grouped words in lyrics (dynamic size) and took that relative part of the melody:
* For each group we assigned the same melody vector under the assumption that the melody is similar in segments of words/time.
* The vectorization of the melody considered the following features (size=143):
  + Average pitch.
  + Average velocity.
  + Instruments played.
  + Chroma features.
* Train data was divided into 534 train samples and 60 validation samples to fine tune training parameters.

## model

For our model we used LSTM that supports two training approaches:

* The first approach concatenated the lyrics vector and the melody vector and treats it as a single input. It uses that input to predict the next word.
* The second approach runs the two vectors in two separate LSTM layers and concatenates them at the end before running it through the fully connected layer.

We hope to use the first approach as a baseline and the second approach as a way to improve upon it to give more predictive power to the melody LSTM layer.

Model parameters:

* Text vector size (300).
* Melody vector size (143).
* Hidden layer size – size of LSTM layer hidden layers.
* Output size – the size of the word2vec model vocabulary.
* Number of layers – number of hidden layers in LSTM.
* Dropout – our chosen regularization method.
* Concatenate – a flag indicates the approach to train by.

## training

For training, initially we’ll run the model with different hyperparameters using the first approach. We’ll document training loss and validation loss throughout training.

Firstly, we’ll test the following hyperparameters:

* Segment sizes (grouping lyrics into melodies): [1,4,8]
* Sequence lengths (the context in which to train LSTM model): [50,100,200]

Due to computational costs, we’ll train using epochs=30, batch size=1, learning rate=0.001, dropout=0 and number of LSTM hidden layers=2. After finding the optimal hyperparameters we’ll further tune one:

* Epochs: [30,50]
* Batch size: [1,4]
* Learning rate: [0.0005, 0.001]
* Dropout: [0,0.2]
* LSTM hidden layers: [2,6]

Next, we’ll test the second approach.

## experiments