Translating lyrics to chords with RNNs

Thomas Benchetrit, Matheus V. Bernat, Husam Jubran
thomas.benchetrit@epfl.ch, matheus.vieirabernat@epfl.ch, husam.jubran@epfl.ch
Computers and Music COM-418 - EPFL

***Abstract*—We human beings have a great power of creativity, and machines have a great power of extracting hidden patterns and replicating them, so why not join these two powers? In this project we aim to help uninspired songwriters who have the perfect lyrics but want a chord progression that goes well with the lyrics. To do this, we constructed a dataset of English songs with nearly 30 thousand songs from Ultimate Guitar, perform chord and lyrics analyses on it, and then construct a recurrent neural network (RNN) that takes in lyrics and outputs a 4-long chord progression. As the main results of this project, we have the dataset whose features can be further explored, and a Jupyter notebook with the RNN that takes in lyrics and outputs the 4-long chord progression.**

I. Introduction

As a songwriter, oftentimes you have great lyrics but you lack inspiration or the musical knowledge to find suitable chord progressions adapted to those lyrics. Depending on the happiness level of the song, it will have more or less major and minor chords; or maybe if it contains an expression of feelings rather than a story-telling (a characteristic of blues) then minor 7^{th} chords are a great match. To deal with this translation between lyrics to a chord progression, we designed a recurrent neural network (RNN) that receives arbitrarily long lyrics as input and outputs a 4-chords long progression.

II. Methods

A. Construction of the dataset

To our specific task of generating chord progressions from lyrics, we did not find any sufficiently large dataset containing lyrics and the chords. Therefore, we scraped songs from Ultimate Guitar [1]. As a result, the dataset has then the following columns: name, author, lyrics, chords, and link.

For the scraping of the dataset, the Beautiful soup [2] library was used and the Selenium driver for the Tor browser. The lyrics and the chords were extracted from the corresponding HTML-divs. Songs from five selected genres were extracted (Country, Rock, Rb, Pop, and Jazz), from all the available decades (1960 to 2020).

B. Analysis of the dataset

The songs in the dataset were analyzed by the emotion they conveyed among the finite set of emotions taken from the Russel model [3]: joy, sadness, anger, fear, neutrality, and calm. The model used for this emotion classification was the Hugging Face zero-shot classification [4]. It took around 10 hours to classify all the 29.7 thousand songs.

Moreover, emotion classification was also performed, where each song's lyrics were given a polarity sentiment between -1 and +1, where -1 represents a purely negative quote and +1 a purely positive one. This was done using the VADER sentiment analysis tool [5]. With an absolute threshold of 0.3, songs with a sentiment of less than -0.3 are set to *negative*, songs with a sentiment of more than +0.3 are set to *positive*, and songs with a sentiment between -0.3 and +0.3 are set to *neutral*. It took around 15 minutes to classify all the 29.7 thousand songs.

Finally, lemmatization and tokenization of the lyrics were done to explore the lyricism of all songs. The Python library Spacy [6] was used to realize among others: the counting of the number of unique words, pronouns, and verbs.

C. Modeling

Two models were used for the prediction of the chord progressions

- The Encoder which is a Recurrent Neural Network composed of a GRU cell and 2 Linear layers
- The Decoder which is also a RNN with an Embedding Layer, a GRU cell and a Linear Layer

First, the lyrics gets encoded as an hidden state vector, then this vector is pushed through the decoder to output the chord progression. Moreover, during the training phase, because of dimensionality, the first chord of the ground truth progression was fed to the model. Indeed, as several chords can be similar in term of harmonic functionnalities, this was used as a way to reduce the output dimensionality. On top of it, in order to adress class imbalance, the ”rarity” metric evoked earlier was use to ponder the loss in favor of lesser used chords progression. This bias in the loss prevented the model to output only the most common chord progressions. Finally, during prediction, the first chord is uniformly drown from the available chords in order to produce variety in the outputed progressions.

III. Dataset

A. Pre-processing

Pre-processing was made in the following ways:

- Removal of duplicates (i.e. songs with the same authors and titles)
- Removal of songs with non-English lyrics.
- Extraction for each song its most common 4-long chord progression
- Computed a ”rarity” metric in order to perform data-augmentation on rarer chord progressions

- Data augmentation was done by computing transposed versions of the chord progression into other keys.

B. Statistics

The initial dataset of 37 thousand songs in 36 different languages and five different genres: Rock, Pop, Country, Jazz, and Rb, Funk and Soul. After pre-processing, the dataset was left with 29 thousand songs. See Figure 1 for the genre distribution.

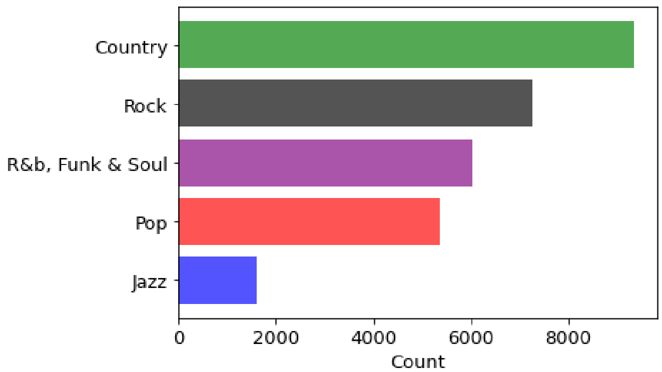


Fig. 1: Songs distribution per genre.

C. Chord analysis

Following the construction and pre-processing of the dataset, chord analysis of the dataset's chords was performed. First, we wanted to understand the number of amount of chords per song. See Figure 2 for this distribution. From this study of chord distribution, we discovered among others:

- That 99.5% of all songs have less than 250 chords;
- A bug in chord extraction: some of the chords were extracted wrongly due to the way the scraping was done following the HTML divisions. This is the case of some of the songs with only 4 chords;
- A bug in Ultimate Guitar: a considerable amount of pages were observed to contain an entire album of several songs, thus containing over 1000 chords.

Following this study, we also analyzed the sentiment of the songs. See Figure 3 for the distribution of sentiments.

From these sentiments, we wanted to understand whether the sentiment attributed to the lyrics of a song in the cwas reflectedhord progressions used in those songs. Therefore, we selected the most present 4-gram used in each song and analyzed these for the top 200 positive and negative songs. See Figure 4 for the top 20 4-grams in positive and negative songs.

To extract more easily information from Figure 4, we extracted the number of minor chords in positive and negative songs, as our intuition is that negative songs have more minor chords while positive songs have more major chords. We got then: 279 minor chords among the top 200 negative songs and 289 minor chords in the top 200 positive songs. Thus, we conclude that one can not say that (at least with the given

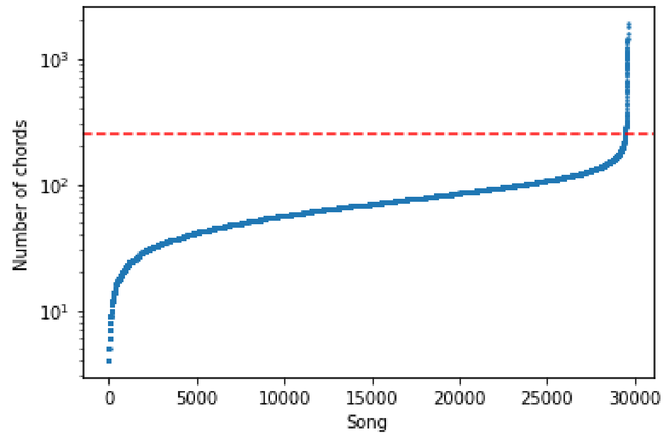


Fig. 2: Number of chords per song. The red dashed line is placed at the y-value of 250.

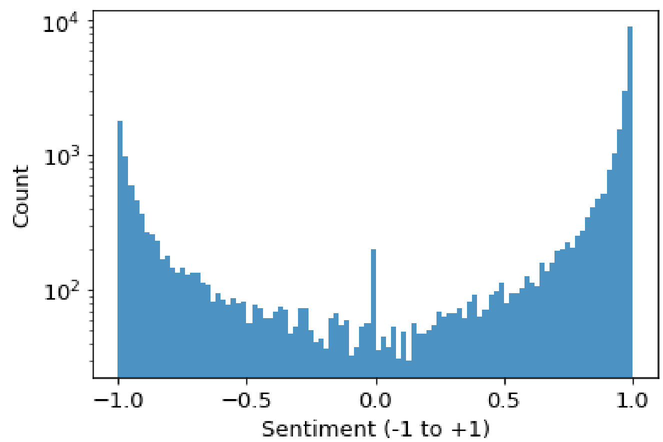


Fig. 3: Distribution of sentiments for all songs, log-scaled in the x-axis. With an absolute threshold of 0.3, there are 7558 negative songs, 1680 neutral songs and 20039 positive songs.

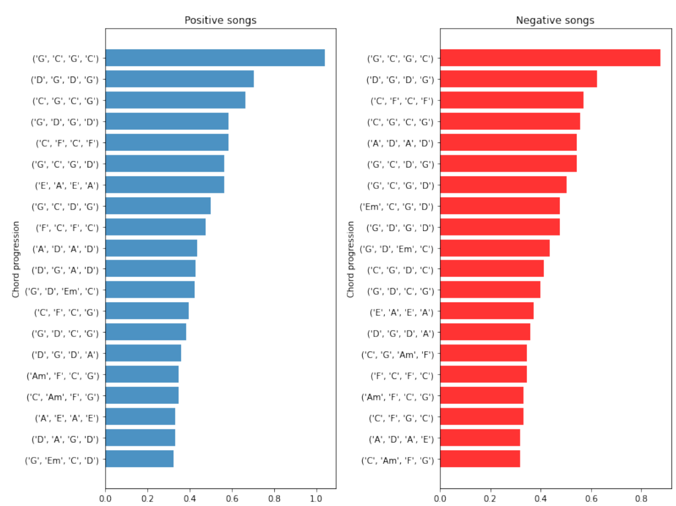
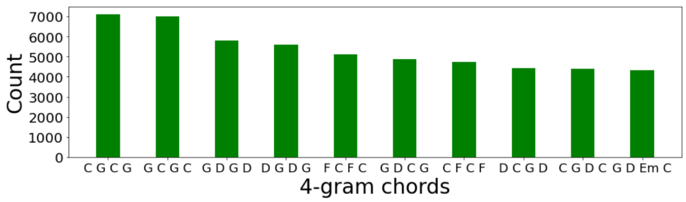
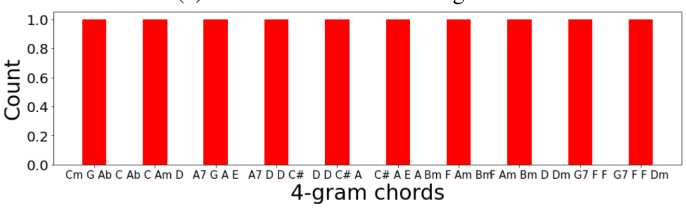


Fig. 4: Caption



(a) Most common chords 4-grams



(b) Least common chords 4-grams

Fig. 5: These histograms show the most/least common chords 4-grams in the whole dataset.

dataset and method for sentiment classification), we cannot state that minor chords are more present in negative songs.

In addition to all the mentioned above, a study on the chord's n-grams was performed, specifically 4-grams, and for the whole songs in the dataset. First, we looked at the most common 4-gram chords in our dataset, we can see the result in Figure 5a,for example, the chord progression C G is know to be very popular in western music, and we can see that it appears in many of the most common n-grams. On the other side, we looked at the least common 4-grams, we got the result that appears in Figure 5b

Furthermore, we looked at the most common chord progression lengths in the whole dataset, the results are in Figure 6 , we can see that the most common length are between between 45 and 70.

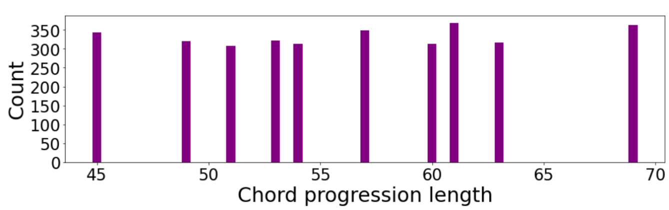


Fig. 6: This histograms show the most common chord progression length in the whole dataset.

D. Lyrics analysis.

In the analysis of the lyrics, we counted the number of unique words, and also the most present nouns, verbs, and pronouns. From this study we got the following insights:

- Rb, Funk and Soul is the music genre with the least number of unique words with 758 words, while Country is the one with the most unique words used with 1074;
- The word 'love' is the most present word in all styles;
- The word 'baby' is in the top 10 of most used nouns in all styles but in Country.
- The verbs through all genres all pretty much all similar (*get*, *let*, *know*, *go*, *come*, *tell*) and don't convey much interesting information.

TABLE I: Emotions table.

Genre	neutral	calm	sad	joy	fear	anger
Jazz	0.2495	**0.2793**	0.1569	**0.1427**	0.0971	0.0745
Rock	0.2465	0.2553	0.1670	0.1102	**0.1234**	**0.0976**
Pop	0.2451	0.2656	0.1526	0.1288	0.1213	0.0865
Rb Funk Soul	0.2401	0.2593	0.1564	0.1328	0.1184	0.0929
Country	**0.2584**	0.2698	**0.1698**	0.1201	0.0969	0.0850

Finally, from the quantified values of emotions in Table I we draw the following conclusions:

- Jazz has the maximum emotions of calmness and joy;
- Country has the maximum emotion of sadness and neutrality;
- Rock has the maximum emotions of fear and anger.

References

[1] "Ultimate guitar," https://www.ultimate-guitar.com/, 2022.

[2] "Beautiful soup," https://beautiful-soup-4.readthedocs.io/en/latest/, 2022.

[3] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *Journal of Research in Personality*, vol. 11, no. 3, pp. 273–294, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/009265667790037X

[4] "Hugging face," https://huggingface.co/, 2022.

[5] "Vader sentiment analysis," https://github.com/cjhutto/vaderSentiment#:~:text=by%20Katie%20Roehrick-,About,on%20texts%20from%20other%20domains., 2022.

[6] "Spacy," https://spacy.io/, 2022.