Yonatan Lipman - 305629750
Amit Nelinger - 203906508

15.08.17

Meaning and Computation

Introduction-

As two music and linguistic enthusiasts, it was clear that our project would be a hybrid of those fields. The relation between language and music is an historically strong one, yet, it was also very vague always. It's clear we have some rules of thumb that most people can agree upon. Usually you won't hear any major scales, up-beat drums or cheerful melodies in a song that dealing with breakups, death or some other depressing subject. It's also common to see that when a singer repeats a phrase, the chord progression is repetitive as well. The musical arrangement, harmony and melody is used to intensify the emotion expressed by the lyrics and vica-versa.

Still, we need to remember music is an art form. And as one, we expect it to be different between artists, and affected by mood swings, influences and intention of the artist itself. The same lyrics is expected to yield a very different result between various artists. One great example, is the takes on Avraham Halfi's lyrics "Kasha Ba'layla Bli Adam" by Alon Eder and Shai Tzabari and the big difference that the harmony induces.
(1. https://www.youtube.com/watch?v=jidZHUoHhDY
2. https://www.youtube.com/watch?v=RQgGJbct58M)

With these conflicting matters in hand, being aware of the complexity and non-deterministic nature of the relationship, we've decided to go on a journey to better understanding of how music and words interact, willing to accept the answer that there might not be one.

Our Project -

We thought we should use machine learning and language analysis algorithms taught in class to create music with AI. Creating melody from lyrics was our first option, but because melodies grows exponentially complex and yet it's more intuitive for non-musicians, we found harmony more appealing. Our motivation was making music more approachable for everyone. To create a chord progression you would need theoretical and practical background in music, which most people don't posses. We decided to create a tool that can help lyricists or any song writer creating harmonies, that would appear in the general known format of how chords for songs are written on the internet. We took inspiration from the classic NLP problem of Parts of Speech

Tagging. Further in this report we would go into details of how we created our tool, problems we encountered, relaxations we had to make and how we evaluated our results.

Procedures-

Gathering Data (Web Scraping):

We wanted to gather as many songs as possible out of the huge collection of 1,100,000 songs available in www.ultimate-guitar.com. To achieve this, we used the "BeautifulSoup" package.

The format in which the lyrics and chords are registered in this site, and many other sites we encountered, is very loose and inconsistent. After examining the data manually and experimenting with different scrapers we decided on how to retrive the data. Our criteria is gathering songs that were of type "Chords", and had a line of chords - which we could recognize by format using BeautifulSoup, that followed with the next line that is an english sentence, which we recognized by using the "enchant" package.

We were able to gather more than a 83,000 songs, resulting in a great amount of words and chords, before being blocked by ultimate-guitar servers. After analyzing the situation, we decided the data is sufficient and it might even be too big for us to learn.

Cleaning Data:

Cleaning the data was the toughest part, mainly because when scraping we opted to getting as much data as we could. We had to assert that the data that we were about to process was really in the format that we agreed on (discussed above). Additionally, we had to deal with throwing replicates, empty songs, unfamiliar chords and non-english songs. After the data was 'clean', we needed to decide on the labeling mechanism. Ideally we would've created an alignment pattern that would've caught n-grams of words per chord, but since the data was so diverse, having long sentences with a single chord, or many chords over few words, chords that don't align to any word and other data diversity problems. For the scope of the project, we decided that we should align a chord for each word, truncating chords that we couldn't map in this one-to-one manner. Eventually, we've collected over ~1,280,000 chords and their aligned lyrics, spanning more than ~15,000 songs. We were able to avoid songs in unrecognized format, different languages and made up chords.

Learning:

Initial Plan and relaxation-

Ideally we wanted to treat a chord progression for his scale functions and not his actual chords. In C scale, the chord progression of "**F**->**G**->**C**" would be analyzed as IV->V->I.

This would've been good for several reasons :

1. Our algorithm could be scale invariant.
2. Musicologists would say same chords in different scales have a different "meaning", but if you play the same functionality in different scales it would sound very similar.
3. Using this analysis we could assess our results in a better way. V and VII are known to be similar yet, V (dominant) and I (tonic) are very different in nature, the first supplying a feeling of tension, while the latter is considered as a feeling of "home".

Unfortunately, after cracking our heads for a long period of time, we understood that we were aiming for a project which is trying to answer a very difficult problem with computational implications that we couldn't even run. Creating algorithms to determine the song scale (which might not be the original), and analyzing the chords functions might be another highly complicated problem with non-deterministic solutions. We've decided to stay with the chords in hand, and do the best we can.

As a middle ground, we looked to using POS tagging solutions for obtaining results that we could compute with the resources available to us. We used the "TextBlob" framework, which relies on "NLTK" in order to train a Naive Bayes classifier.
The classifier learned the data in form of one-to-one word-chord tagging, since as explained in the last section, this was the most computationally reasonable method of alignment and producing outputs for the input text that we classified and tried produce a harmony to.

<u>Evaluation-</u>

Since we've come across a highly complex computational problem, we relaxed our evaluation to a simple counting measurement, knowing that this would most likely not produce results that we could evaluate musically, but we've felt that we should at least see the product of our efforts. Still, since we've come up with a concept for evaluating our initial problem that we feel should be explained in this article.
In order to evaluate how good are algorithms that generate chords from lyrics is, we came up with this estimation function:

$$\textbf{Score = f(song) = (}\Sigma\textbf{ R(ACi, RCi)) / N}$$

Where function f receives a **song** with chord labels for n-grams of words and outputs chords. It iterates over each word in the song and send the Algorithm-Chord (**ACi**) and

the Real-Chord of the song (**RCi**) to the function **R.** this function counts how many corresponding notes are between the two chords and divides it by the number of notes in the chord that has more notes. For example - R(Am7, C) = ¾ (Am is composed of A,C,E,G and C is composed of C,E,G). **N** is the number of words in the songs.
The results will always be between 0-1. 1 being a superb result, and 0 showing our algorithm handled very poorly with this song.
If our algorithm gives the exact same chords as the other song, in each iteration R will return 1, and then our total would be N/N = 1 (perfect score).

The question of which chords are more alike is a very difficult one. Is C major more similar to G major because they are both “happy” chords, or is it more similar to Am which has ⅔ notes in common with him. In our opinion, in this case, Am would be more similar to C than G, that’s because of his functionality, which we mentioned earlier. For this reason we’ve decided to go with our algorithm which is simple and we believe it provides valid assessment for the algorithms result.

Discussion-
In the scope of this course, for two students the problem is highly complex and infeasible to solve with the available resources. Yet, we believe we’ve discovered some valuable lessons that might be helpful when tackling the task in hand. First, after encountering large amounts of diverse data, we were impressed that there is no “correct” way to generate chords from lyrics and that style and beat is difficult to quantify, especially in purely textual formats that don't comply with the classical forms of musical notations. More than before, we believe that chord progression for specific lyrics could wear many forms. Secondly, there is no “golden standard” to how songs should be written in tablature websites and there is no enforcement, making the data contain many typo’s and chord mistakes. Third, learning this sort of problem has to take many factors in consideration that initially we weren't aware of: Frequencies of chord progression, relations between notes inside the scale, emotion expressed in the lyrics that implies a different sense of harmony, style of music, etc’.
In conclusion, we’ve met a problem that we didn’t expect would be so difficult computationally and conceptually, where there are factors that don’t have an obvious quantification that could only be explained in subjective feelings, in our sense, and needs expertise and knowledge out of our reach. Nevertheless, we enjoyed tackling these problems and mapping out the different ways we could approach for producing such a solution for something that is heavily rooted in senses that don't end in textual format,. We’ve enjoyed working on something we love and wanted to try to enrich these fields.