Nick Steelman

Flow Reader

At the start of the project, my vision was to create a software that would be able to grab some song lyrics from a specified website and then analyze those lyrics for rhyme schemes and return some data about what it finds.

In order to tackle this problem I would first need to find out if two words rhyme. To do this I downloaded a toolbox that contains the CMU phonetic spelling dictionary, which allows me to input a word and get the broken down pronunciation. From there I wrote my own function to determine the extent of which two word rhyme.

The function essentially counts back from the end of each word and if the phonetic sound is the same as the phonetic sound at the other index, I add 100 to the total Rhyme Score. Additionally if any of the phonetic sounds match at the index directly above or below, I add 50 to the score, so even if they somewhat rhyme but the syllables are a little off, points are still awarded.

From there I wrote analyze_text that will take a string and parse it into an array of words, cutting out periods, commas and question marks. It will then run through the array and compare each word to the 10 previous words, if the total Rhyme Score is >= 100 it will add 1 to the Rhyme Count. An added affect of this is that more rhyming words will have a snowball effect. So:

Bat, Cat -------------->+ 1
Bat,Cat,Sat --------->+ 3
Bat,Cat,Sat,Hat----->+6
etc.

At the end of each verse of the song, it will total the Rhyme Count and divide by the number of words and finally multiply by 100 to achieve a total Rhyme Rate. At the end of the song it will total all Counts and Words to compute the total Rhyme Rate for a song.

Finally, I wrote some simple html searching code to grab and analyze the verses from any song given the Metrolyrics webpage.

Extensions:

After this implementation, the code worked decently well, but there was still a large amount to improve on. Firstly, the CMU dictionary is not all inclusive and often does not understand words. In some rap songs it would miss over 30 words per verse. This is due to the presence of some slang such as “ ‘cause ” or names such as “Loius Vuitton”. To combat this, when the program comes across a word it does not know, it will lop off the first letter until it does. So ‘cause will turn to cause, which has the same pronunciation. This greatly improved readability with only slightly reducing accuracy, since there is a rare case where eliminating off the beginning of a word changes how the end is pronounced.

Additionally, the program would rhyme words with themselves and add repeat rhymes to the total rhyme count. I did not feel like this was fair so I added a list to make sure that the same rhyme would not be counted twice in one verse and an if statement that checks if the two words being checked are the same.

I also played around with how conservative it is in determining rhymes. Right now it is very liberal but it can easily be changed to only detect words that really rhyme, not just sound alike.

After these changes, the program seems to be much more accurate and will recognize the differences between rhymes with high rhyme content and low rhyme content.

Results:

Artist	Song	Rating
Alexander Hamilton	My Shot	98
Drake	One Dance	51
Wu-Tang	Da Mystery of ChessBoxin	131
Pokemon	Pokemon Theme	68
21 Pilots	Heathens	99
Ed Sheeran	Shape of You	79
Busta Rhymes	Break Ya Neck	98
Eminem	Lose Yourself	169
Eminem	Rap God	135
Biggie Smalls	Hypnotize	116
Queen	Bohemian Rhapsody	82
MF Doom	Doomsday	133
Frank Sinatra	Under My Skin	101