
RHYME ANALYZER

RHYME SCHEMES

Position of rhymes

End rhymes

Internal rhymes

Perfect Rhymes

Near Rhymes

Assonance Rhymes

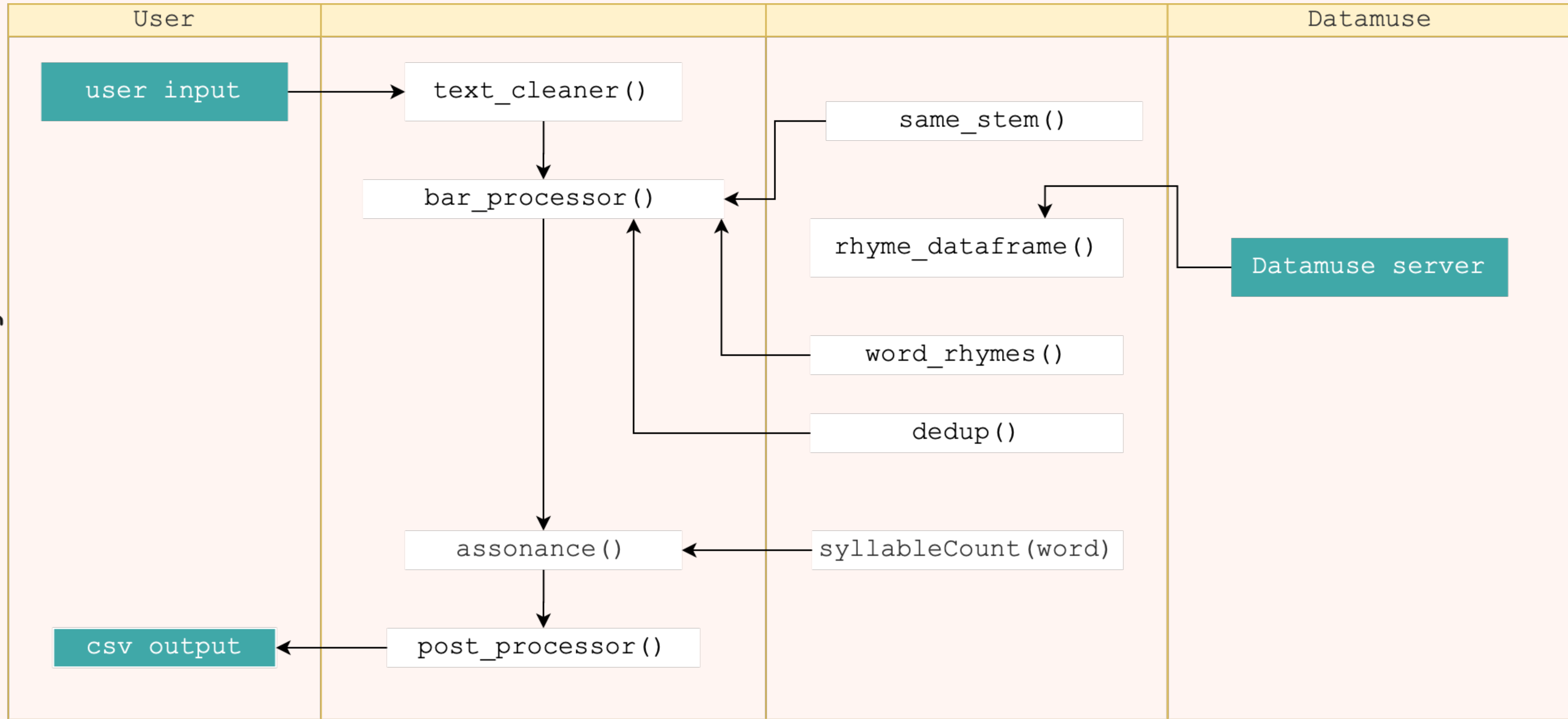
Compound Rhymes

Example

His palms are sweaty, knees weak, arms are heavy
There's vomit on his sweater already: Mom's spaghetti
He's nervous but on the surface he looks calm and ready
To drop bombs, but he keeps on forgetting

What he wrote down, the whole crowd goes so loud
He opens his mouth, but the words won't come out
He's choking how? Everybody's joking now
The clock's run out, time's up, over, blaow!

Snap back to reality, oh, there goes gravity
Oh, there goes Rabbit, he choked, he's so mad but he won't
Give up that easy, nope, he won't have it, he knows
His whole back's to these ropes, it don't matter, he's dope
He knows that, but he's broke, he's so stagnant, he knows
When he goes back to this mobile home that's when it's
Back to the lab again, yo! This whole rhapsody



```

import inflect
import pandas as pd
import string
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.snowball import SnowballStemmer

p = inflect.engine()

RHYME_DICT = {} # global rhyming dict w/ key: word & value: rhymes in lyrics
VOWELS = ['AA1', 'AE1', 'AH1', 'AO1', 'AW1', 'AY1', 'EH1', 'ER1',
          'EY1', 'IH1', 'IY1', 'OW1', 'OY1', 'UH1', 'UW1'] # stressed vowel

def txt_Dict():
    filepath = "phodict.txt"
    dict1 = dict()
    with open(filepath) as file1:
        for line in file1:
            words = line.split() # splits line on whitespace
            dict1[words[0]] = tuple(words[1:])
    return dict1

PHODICT = txt_Dict() # phonetic dictionary

def rhyme_generator():
    "Takes user inputted lyrics and stores into list and dictionary"
    lyrics = input("Input lyrics to rhyme scheme analyzer: ")
    BAR_LIST = text_cleaner(lyrics)
    bar_processor(BAR_LIST)
    assonance()
    post_processor()
    df = pd.DataFrame.from_dict(RHYME_DICT, orient='index')
    df.to_excel('output_rhymes.xls', sheet_name = 'Rhymes')

```

```

def text_cleaner(lyrics):
    "Helper function that cleans up lyrics for list and rhyme dictionary"
    tokens = word_tokenize(lyrics)
    # converts words to lowercase
    tokens = [element.lower() for element in tokens]
    table = str.maketrans('', '', string.punctuation)
    stripped = [w.translate(table) for w in tokens] # strips punctuation
    BAR_LIST = [word for word in stripped if word.isalpha()] # keep alpha words
    stop_words = set(stopwords.words('english')) # set of words to ignore
    BAR_LIST = [word for word in BAR_LIST if not word in stop_words] # filter
    return BAR_LIST

```

nltk.tokenize: Splits words based on any punctuation and strips punctuation

nltk.corpus: Allows import of stopwords, so can filter out unimportant words like “he,” “and,” etc.

nltk.stem.snowball: Supports checking if words have the same stem (i.e. run, runs, running), so they may be put into the same rhyme group

nltk: Natural language processing library/module for Python

PHODICT: text file from CMU dictionary where most common words of the English language and their phonetic breakdowns are listed


```

def bar_processor(BAR_LIST):
    "Function iterates through the list and searches each word for its rhymes"
    bar_set = set(BAR_LIST)
    bool_dict = dict.fromkeys(bar_set, False)
    for i in bar_set:
        if (bool_dict[i]): # if visited, skip]
            continue
        intersection = same_stem(i, bool_dict)
        # find intersection bw rhymes & words in bar set
        intersection2 = bar_set.intersection(word_rhymes(BAR_LIST,i))
        intersection3 = dedup(intersection2, bool_dict)
        intersection |= intersection3
        for j in intersection: # mark as True b/c visited now
            bool_dict[j] = True
        list_rhyme = list(intersection)
        RHYME_DICT[i] = list_rhyme

def same_stem(word, bool_dict):
    '''Checks which words have the same stem and place them into the same rhyme group'''
    stemmer = SnowballStemmer("english")
    stem = stemmer.stem(word)
    intersection = set()
    intersection = {w for w in bool_dict.keys() if (not bool_dict[w] and (stemmer.stem(w) == stem))}
    return intersection

def word_rhymes(BAR_LIST, word):
    "Calls scraper for perfect and near rhymes of argument word from lyrics"
    perfect_rhymes = rhyme_dataframe(word, 'rhy') # perfect rhymes
    if (len(BAR_LIST) > 500):
        return(perfect_rhymes)
    near_rhymes = rhyme_dataframe(word, 'nry')
    df_rhy = perfect_rhymes | near_rhymes
    return(df_rhy)

def rhyme_dataframe(word, rhyme_type):
    "Scrapes rhymes into pandas df and outputs set"
    base_url = 'https://api.datamuse.com/words?rel_'
    url = base_url + rhyme_type + '=' + word + "&max=50"
    df_rhy = (pd.read_json(url)) # url provides data in json format
    if (df_rhy.empty):
        return(set())
    df_rhy = df_rhy.loc[:, ['word']]['word']
    df_rhy = set(df_rhy.unique())
    return df_rhy

```

```

def dedup(set1, bool_dict):
    "Prevents addition of duplicate words"
    set2 = {w for w in set1 if (not bool_dict[w])}
    return set2

def assonance():
    "Refines dictionary items by combining assonance rhymes"
    syl_dict = {}
    for i in RHYME_DICT:
        try:
            if (syllableCount(i.upper()) == 1):
                syl_dict[i] = PHODICT[i.upper()]
        except KeyError:
            pass
    for vow in VOWELS:
        vow_list = [word for word in syl_dict.keys() if vow in syl_dict[word]]
        length = len(vow_list)
        vow_set = set()
        for index in range(length):
            word = vow_list[index]
            vow_set |= set(RHYME_DICT[word])
            if (index == length - 1):
                RHYME_DICT[word] = list(vow_set)
                break
        RHYME_DICT.pop(word)

def syllableCount(word):
    "Calculates number of syllables in argument word"
    word = word.upper()
    syllables = PHODICT[word]
    return sum(int(syllable[0] in "AEIOU") for syllable in syllables)

def post_processor():
    "Removes single length rhyme groups"
    RHYME_DICT_COPY = RHYME_DICT.copy()
    for word in RHYME_DICT_COPY.keys():
        if (len(RHYME_DICT_COPY[word]) == 1):
            RHYME_DICT_COPY.pop(word)

rhyme_generator()

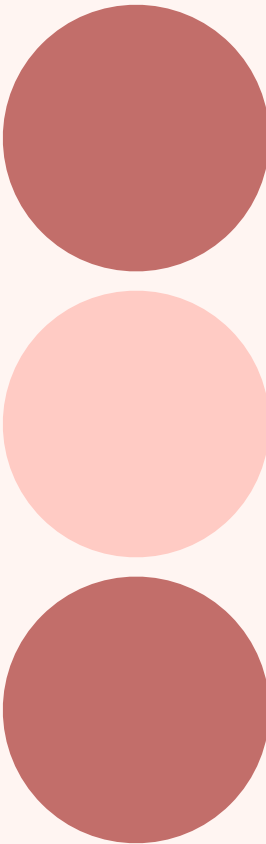
```

1545 WORDS

Eminem's *Rap God*

LOOK, I WAS GONNA GO EASY ON YOU AND NOT TO HURT YOUR FEELINGS
BUT I'M ONLY GOING TO GET THIS ONE CHANCE
SOMETHING'S WRONG, I CAN FEEL IT (SIX MINUTES, SLIM SHADY, YOU'RE ON)
JUST A FEELING I'VE GOT, LIKE SOMETHING'S ABOUT TO HAPPEN, BUT I DON'T KNOW WHAT
IF THAT MEANS, WHAT I THINK IT MEANS, WE'RE IN TROUBLE, BIG TROUBLE,
AND IF HE IS AS BANANAS AS A SAY, I'M NOT TAKING ANY CHANCES
YOU WERE JUST WHAT THE DOCTOR ORDERED
I'M BEGINNING TO FEEL LIKE A RAP GOD, RAP GOD
ALL MY PEOPLE FROM THE FRONT TO THE BACK NOD, BACK NOD
NOW WHO THINKS THEIR ARMS ARE LONG ENOUGH TO SLAP BOX, SLAP BOX?
THEY SAID I RAP LIKE A ROBOT, SO CALL ME RAPBOT
BUT FOR ME TO RAP LIKE A COMPUTER MUST BE IN MY GENES
I GOT A LAPTOP IN MY BACK POCKET
MY PEN'LL GO OFF WHEN I HALF-COCK IT
GOT A FAT KNOT FROM THAT RAP PROFIT
MADE A LIVING AND A KILLING OFF IT
EVER SINCE BILL CLINTON WAS STILL IN OFFICE
WITH MONICA LEWINSKY FEELING ON HIS NUT-SACK
I'M AN MC STILL AS HONEST
BUT AS RUDE AND INDECENT AS ALL HELL SYLLABLES, KILLAHOLIC (KILL 'EM ALL WITH)
THIS SLICKETY, GIBBEDY, HIBBEDY HIP HOP
YOU DON'T REALLY WANNA GET INTO A PISSING MATCH WITH THIS RAPPIDY BRAT
PACKING A MAC IN THE BACK OF THE AC, BACKPACK RAP CRAP, YEP, YACKIDY-YAC
AND AT THE EXACT SAME TIME I ATTEMPT THESE LYRICAL ACROBAT STUNTS WHILE I'M PRACTICING
THAT I'LL STILL BE ABLE TO BREAK A MOTHERFUCKIN' TABLE
OVER THE BACK OF A COUPLE OF FAGGOTS AND CRACK IT IN HALF
ONLY REALIZED IT WAS IRONIC I WAS SIGNED TO AFTERMATH AFTER THE FACT
HOW COULD I NOT BLOW? ALL I DO IS DROP F-BOMBS, FEEL MY WRATH OF ATTACK
RAPPERS ARE HAVING A ROUGH TIME PERIOD, HERE'S A MAXIPAD
IT'S ACTUALLY DISASTROUSLY BAD
FOR THE WACK WHILE I'M MASTERFULLY CONSTRUCTING THIS MASTERPIECE AS
I'M BEGINNING TO FEEL LIKE A RAP GOD, RAP GOD
ALL MY PEOPLE FROM THE FRONT TO THE BACK NOD, BACK NOD
NOW WHO THINKS THEIR ARMS ARE LONG ENOUGH TO SLAP BOX, SLAP BOX?
LET ME SHOW YOU MAINTAINING THIS SHIT AIN'T THAT HARD, THAT HARD
EVERYBODY WANT THE KEY AND THE SECRET TO RAP IMMORTALITY LIKE I HAVE GOT
WELL, TO BE TRUTHFUL THE BLUEPRINT'S SIMPLY RAGE AND YOUTHFUL EXUBERANCE
EVERYBODY LOVES TO ROOT FOR A NUISANCE
HIT THE EARTH LIKE AN ASTEROID, DID NOTHING BUT SHOOT FOR THE MOON SINCE
MC'S GET TAKEN TO SCHOOL WITH THIS MUSIC
'CAUSE I USE IT AS A VEHICLE TO BUS THE RHYME
NOW I LEAD A NEW SCHOOL FULL OF STUDENTS

ME? I'M A PRODUCT OF RAKIM, LAKIM SHABAZZ, 2PAC N-
-W.A, CUBE, HEY, DOC, REN, YELLA, EAZY, THANK YOU, THEY GOT SLIM
INSPIRED ENOUGH TO ONE DAY GROW UP, BLOW UP AND BE IN A POSITION
TO MEET RUN DMC AND INDUCT THEM INTO THE MOTHERFUCKIN' ROCK N'
ROLL HALL OF FAME
EVEN THOUGH I WALK IN THE CHURCH AND BURST IN A BALL OF FLAMES
ONLY HALL OF FAME I BE INDUCTED IN IS THE ALCOHOL OF FAME
ON THE WALL OF SHAME
YOU FAGS THINK IT'S ALL A GAME 'TIL I WALK A FLOCK OF FLAMES
OFF OF PLANKING, TELL ME WHAT IN THE FUCK ARE YOU THINKING?
LITTLE GAY LOOKING BOY
SO GAY I CAN BARELY SAY IT WITH A STRAIGHT FACE LOOKING BOY
YOU WITNESSING A MASS OCCUR
LIKE YOU WATCHING A CHURCH GATHERING TAKE PLACE LOOKING BOY
OY VEY, THAT BOY'S GAY, THAT'S ALL THEY SAY LOOKING BOY
YOU GET A THUMBS UP, PAT ON THE BACK
AND A "WAY TO GO" FROM YOUR LABEL EVERYDAY LOOKING BOY
HEY, LOOKING BOY, WHAT YOU SAY LOOKING BOY?
I GOT A "HELL YEAH" FROM DRE LOOKING BOY
I'MMA WORK FOR EVERYTHING I HAVE
NEVER ASK NOBODY FOR SHIT, GET OUTTA MY FACE LOOKING BOY
BASICALLY BOY YOU'RE NEVER GONNA BE CAPABLE
TO KEEP UP WITH THE SAME PACE LOOKING BOY
'CAUSE I'M BEGINNING TO FEEL LIKE A RAP GOD, RAP GOD
ALL MY PEOPLE FROM THE FRONT TO THE BACK NOD, BACK NOD
THE WAY I'M RACING AROUND THE TRACK, CALL ME NASCAR, NASCAR
DALE EARNHARDT OF THE TRAILER PARK, THE WHITE TRASH GOD
KNEEL BEFORE GENERAL ZOD THIS PLANET'S KRYPTON, NO ASGARD, ASGARD
SO YOU BE THOR AND I'LL BE ODIN, YOU RODENT, I'M OMNIPOTENT
LET OFF THEN I'M RELOADING IMMEDIATELY WITH THESE BOMBS I'M TOTIN'
AND I SHOULD NOT BE WOKEN
I'M THE WALKING DEAD, BUT I'M JUST A TALKING HEAD, A ZOMBIE FLOATING
BUT I GOT YOUR MOM DEEP THROATING
I'M OUT MY RAMEN NOODLE, WE HAVE NOTHING IN COMMON, POODLE
I'M A DOBERMAN, PINCH YOURSELF IN THE ARM AND PAY HOMAGE, PUPIL
IT'S ME, MY HONESTY'S BRUTAL
BUT IT'S HONESTLY FUTILE IF I DON'T UTILIZE WHAT I DO THOUGH
FOR GOOD AT LEAST ONCE IN A WHILE
SO I WANNA MAKE SURE SOMEWHERE IN THIS CHICKEN SCRATCH I SCRIBBLE AND DOODLE
ENOUGH RHYMES TO MAYBE TO TRY AND HELP GET SOME PEOPLE THROUGH TOUGH TIMES
BUT I GOTTA KEEP A FEW PUNCHLINES JUST IN CASE 'CAUSE EVEN YOU UNSIGNED
RAPPERS ARE HUNGRY LOOKING AT ME LIKE IT'S LUNCHTIME



44 X 30

(CATEGORIES X MAX SIZE)

Eminem's Rap God

[illegible]

FUTURE IMPROVEMENTS

(Immediately possible refinements)

- **Refine assonance rhyme because not all same vowel sound words rhyme (i.e. “rap,” “tack,” “rat” rhyme, but “rap” and “ham” are too much of a stretch)**
 - **Refine assonance rhyme function to combine not only single syllable categories but also multi-syllabic categories (i.e. from previous slide: “attack” and “half”)**
 - **Host the program on a website server for others to use**
-

FUTURE IMPROVEMENTS

(More difficult refinements)

- **Have the program guess the pronunciation of proper nouns and slang (rap contains a plethora of references from pop culture, etc., which will not be found in the CMU pronunciation dictionary)**
 - **Have the program match compound rhymes, rhymes that span multiple words**
 - **Change from outputting excel file to highlighting words (currently not supported because too many categories and thus too many colors)**
-