Lyrical Patterns in the Country Music Genre: Distinctions across Genders and Decades

Jessica Pasik

COMM 313 – Spring 2021

Background & Research Question

- **Research Question**: How has the country music genre changed over time – from the 1990s to 2010s? Are there any prevalent changes between men and women both within and between these decades? What evidence can I find through lyrical analysis that could tell me more about where the country genre is headed in the future? What can be expected?
- **Hypothesis**: I hypothesize that while there may be diversity in topics over the decades, the lyrical analysis will reveal great similarity in lyrics from the 1990s and 2010s; I believe any substantial lyrical differences will be more prominent between male and female artists.

Organization / Data Collection

- **Data Collection:**
  - Excel chart of top country songs from 1990s & 2010s, separated by gender (songs pulled from online data citing top songs for the decades, award-winning songs, and nominated songs)
  - Chart includes assortment of artists (rare repetition); chart created to scrape decade, title, artist(s), gender into separate dictionaries → below are some of the beginning steps I took in organizing the data:

(1) Created mass dictionary to run through all files

(2) Created a function to tokenize *all* lyrics

(3) Processed all charts from different JSON files:

- all_songs, all_female, all_male, all_90s, all_2010s, female_90s, male_90s, female_2010s, male_2010s.
- Used Genius API to pull lyrics for every song

```
#creating mass dictionary that runs through all files
chart_dict = {}
for chart in chart_files:
      chart_dict[chart[:-5]] = json.load(open('data/charts/{}'.format(chart)))
```

(1)

```
def process_chart(songs):
    for song in songs:
        str_stripped = strip_section_markers(text=song['lyrics'])
        toks = tokenize(str_stripped, lowercase=True, strip_chars=char_to_strip)
        song['tokens'] = toks
```

(2)

```
# loading data from each JSON file (see below)
# processing all charts
for chart in chart_dict:
      process_chart(chart_dict[chart])
```

```
chart_dict['all_songs'][0].keys()
```

```
dict_keys(['Decade', 'Title', 'Artist', 'Gender', 'lyrics', 'tokens'])
```

(3)

- **Data Description:**
  - 80 songs total
    - 40 songs per decade
    - 20 songs per gender per decade

```
## finding length of each key in the dictionaries
for chart,value in chart_dict.items():
    print(chart,len(value))
```

```
all_2010s 40
all_90s 40
all_female 40
all_male 40
all_songs 80
female_2010s 20
female_90s 20
male_2010s 20
male_90s 20
```

Analysis: Tokenization, Frequencies, n-gram lists (bigrams/trigrams) & Word Comparisons

Frequencies

```
#Frequency List for all songs of the 2010s

word_freq_2010s = Counter() ##tally of # of times each type occurs in TOTAL
song_freq_2010s = Counter() ## tally of # of songs each type occurs in (each type only counted
bigrams_2010s_dist = Counter()
trigrams_2010s_dist = Counter()

# 1. loop over each song in chart
for song in chart_dict['all_2010s']:
    raw_lyrics = song['lyrics']
    song_toks = []
    temp = tokenize(raw_lyrics, lowercase=True,strip_chars=char_to_strip)
    for tok in temp:
        if tok not in stop_words:
            song_toks.append(tok)
    word_freq_2010s.update(song_toks)
    unique_type = set(song_toks)
    song_freq_2010s.update(unique_type) # 2b. update song_freq with the types (i.e. unique valu
    bigram_2010s = get_ngram_tokens(song_toks, n=2)
    bigrams_2010s_dist.update(bigram_2010s)
    trigrams_2010s = get_ngram_tokens(song_toks, n=3)
    trigrams_2010s_dist.update(trigrams_2010s)

print('Top 50 words in your `all_2010s` corpus\n', '='*34, sep='')
print(word_freq_2010s.most_common(50))

print('Top 50 # of songs each type occurs in your `all_2010s` corpus\n', '='*34, sep='')
print(song_freq_2010s.most_common(50))

print('Top 50 bigrams in your `all_2010s` corpus\n', '='*34, sep='')
print(bigrams_2010s_dist.most_common(50))

print('Top 50 trigrams in your `all_2010s` corpus\n', '='*34, sep='')
print(trigrams_2010s_dist.most_common(50))
```

Words to Compare

```
## comparing some key words present in BOTH the 90s and 2010s lyrics - key words found on various websites

words_to_compare = ['beer','boots','drink','drinking','drinks','truck','road','yeah','baby','girl','love','little','dr
print("{:<10} {:<10} {:<10}".format("word", "Tokens-1990s", "Tokens-2010s"))
print("="*30)

for word in words_to_compare:
    print("{:<10} {:<10} {:<10}".format(word,
                                        word_freq_90s.get(word,0),
                                        word_freq_2010s.get(word,0)))
```

```
word       Tokens-1990s Tokens-2010s
==============================
beer       5            9
boots      1            17
drink      3            16
drinking   0            0
drinks     0            6
truck      9            5
road       5            34
yeah       56           54
baby       27           47
girl       20           19
love       108          17
little     45           62
drunk      0            14
whiskey    6            20
```

^These are frequently used words in both old and modern country songs; comparing their frequencies in this chart

<-- This code block was carried out for every dictionary; finds tokens, removes stop words, finds n-grams and frequencies

Analysis: KWIC Concordances & Keyness Analysis

KWIC Concordance

```
#create list of KWIC concordance lines of "HEART"
kwic_heart_2010s = []
for song in chart_dict['all_2010s']:
    if song.get('tokens'):
        kwic_rel_heart2010s = make_kwic('heart', song['tokens'])
        kwic_heart_2010s.extend(kwic_rel_heart2010s)

print(f'"heart" occurs {len(kwic_heart_2010s)} times in your lyrics')

#sort and view them
heart_sorted_2010s = sort_kwic(kwic_heart_2010s, ['L1','R1'])
print_kwic(heart_sorted_2010s)
```

Keyness Analysis

```
## keyness analysis: key words in the 90s subset vs those in 2010s
calculate_keyness(word_freq_90s, word_freq_2010s, top = 20)
```

WORD	Corpus A Freq.	Corpus B Freq.	Keyness
love	108	17	82.091
boy	32	7	19.623
let	46	18	15.250
tell	36	14	12.043
want	24	7	11.406
i've	35	14	11.232
really	19	5	9.980
ya	34	16	8.301
maybe	19	6	8.290
know	71	46	7.798
say	30	14	7.440
would	22	9	6.822

```
## keyness analysis: key words in the 90s subset vs those in 2010s
calculate_keyness(word_freq_2010s, word_freq_90s, top = 20)
```

WORD	Corpus A Freq.	Corpus B Freq.	Keyness
back	74	15	37.647
every	51	10	26.660
'em	39	6	24.238
road	34	5	21.728
like	107	55	12.745
need	31	10	9.523
good	33	12	8.437
free	22	6	8.372
always	19	5	7.527
whiskey	20	6	6.776

Analysis: Part-of-Speech Tagging

```
# POS all 2010s

all_2010s_tagged = []

for song in range(len(chart_dict['all_2010s'])):
    txt = tokenize((chart_dict['all_2010s'][song]['lyrics']), lowercase=True, strip_chars=char_to_strip)
    all_2010s_tagged.append(nltk.pos_tag(txt))
```

```
# picking VERBS
verb_2010s = []
for song in range(len(all_2010s_tagged)):
    for tags in range(len(all_2010s_tagged[song])):
        if all_2010s_tagged[song][tags][1].startswith('V'):
            verb_2010s.append(all_2010s_tagged[song][tags][0])
```

```
# all 2010s verb frequency
verb_2010s_freq = Counter(verb_2010s)
verb_2010s_freq.most_common(20)
```

```
top_25_verb_90s = verb_90s_freq.most_common(25)
```

```
top_25_verb_2010s = verb_2010s_freq.most_common(25)
```

```
### displaying 25 most frequent VERBS in 90s songs and 2010s songs, respectively
for idx,pair in enumerate(top_25_verb_90s):
    print('{:<10}{:<10}{:<15}{:<10}'.format(pair[0], pair[1], top_25_verb_2010s[idx][0], top_25_verb_2010s[idx][1]))
```

```
## what percentage of all tokens in 90s corpus are verbs/adjectives/nouns
## divide sum of each counter by sum of total tokens counter

verb_90s_perc = (sum(verb_90s_freq.values()) / sum(word_freq_90s.values()))* 100
adj_90s_perc = (sum(adj_90s_freq.values()) / sum(word_freq_90s.values()))* 100
noun_90s_perc = (sum(noun_90s_freq.values()) / sum(word_freq_90s.values()))* 100

print('{} percent of the tokens in all 90s songs were verbs'.format(verb_90s_perc))
print('{} percent of the tokens in all 90s songs were adjectives'.format(adj_90s_perc))
print('{} percent of the tokens in all 90s songs were nouns'.format(noun_90s_perc))
```

```
41.93388429752066 percent of the tokens in all 90s songs were verbs
18.694214876033058 percent of the tokens in all 90s songs were adjectives
48.84297520661157 percent of the tokens in all 90s songs were nouns
```

```
## what percentage of all tokens in 2010s corpus are verbs/adjectives/nouns
## divide sum of each counter by sum of total tokens counter

verb_2010s_perc = (sum(verb_2010s_freq.values()) / sum(word_freq_2010s.values()))* 100
adj_2010s_perc = (sum(adj_2010s_freq.values()) / sum(word_freq_2010s.values()))* 100
noun_2010s_perc = (sum(noun_2010s_freq.values()) / sum(word_freq_2010s.values()))* 100

print('{} percent of the tokens in all 2010s songs were verbs'.format(verb_2010s_perc ))
print('{} percent of the tokens in all 2010s songs were adjectives'.format(adj_2010s_perc))
print('{} percent of the tokens in all 2010s songs were nouns'.format(noun_2010s_perc))
```

```
39.20351302241066 percent of the tokens in all 2010s songs were verbs
17.867958812840705 percent of the tokens in all 2010s songs were adjectives
46.577831617201696 percent of the tokens in all 2010s songs were nouns
```

Findings

- Running through the various frequencies, there is a lot of *similarity* in lyrics from the 1990s and 2010s
- ‘love’ appeared significantly more in 1990s country songs than in those of the 2010s
- ‘baby’ occurred almost 2x in 2010s compared to 1990s
- ‘yeah’ and ‘girl’ occurred almost equally between the decades

word	Tokens-1990s	Tokens-2010s
beer	5	9
boots	1	17
drink	3	16
drinking	0	0
drinks	0	6
truck	9	5
road	5	34
yeah	56	54
baby	27	47
girl	20	19
love	108	17
little	45	62
drunk	0	14
whiskey	6	20

Further Analysis / Next Steps

- Collocation
- Sentiment Analysis
- Organizing frequencies into charts for easier comparison
- Closer look at differences in lyrics between genders – i.e., do women sing more about X than men do?