Dataset creation using Last.fm tags

AMP Lab - AcousticBrainz

Roman Tsukanov

SMC, 2016

Idea

Create dataset(s) based on tags scraped from Last.fm.

These tags are assigned by users to tracks. Each tag is associated with a count, which indicates how many times it was assigned to a specific track.

Dataset contains a list of MusicBrainz IDs for recordings (tracks) in AcousticBrainz database. It includes list of tags and counts (normalized) associated with a specific recording.

Step 1

First step is to analyze what kind of tags are there.

The easiest way is to go through each of them and count occurrences.

```
tags = defaultdict(list)  # each tag is mapped to a list of (mbid, normalized count) tuples

def pairwise(iterable):
    i = iter(iterable)
    return izip(i, i)

with open(LASTFM_TAGS_FILE) as tags_file:
    for line in tags_file:
        line_list = line.strip().split(',')
        mbid = line_list[0]
        for tag, count_norm in pairwise(line_list[1:]):
            tags[tag.strip().lower()].append((mbid, count_norm))
```

Step 1

```
tag_occurences = defaultdict(int)
for tag, recordings in tags.iteritems():
    tag_occurences[tag] = len(recordings)
sorted(tag_occurences.items(), key=operator.itemgetter(1), reverse=True)  # by occurrences
```

> rock, alternative, pop, favorites, electronic, alternative rock, metal, indie, classic rock, **female vocalists**, beautiful, love, **awesome**, american, hard rock, **90s**, instrumental, **male vocalists**, **00s**, soundtrack, british, **80s**, singer-songwriter, chillout, mellow, folk, chill, dance, experimental, punk, jazz, seen live, indie rock, favourites, heavy metal, progressive rock, electronica, guitar, favorite, ambient, **70s**, **cool**, oldies, blues, acoustic, classic, favourite, **female vocalist**, epic, favorite songs, **male vocalist**, psychedelic, soul, punk rock, **sad**, loved, melancholy, **8 of 10 stars**, easy listening, pop rock, catchy, hip-hop, piano, party, fun, melancholic, amazing, **60s**, sexy, **6 of 10 stars**, **german**, **happy**, cover, ballad, fip, downtempo, atmospheric, **10 of 10 stars**, funk, favourite songs, dark, soft rock, progressive metal, progressive, **uk**, new wave, hip hop, industrial, death metal, fucking awesome, relaxing, relax, **usa**, rock n roll, **female**, gothic, upbeat, **7 of 10 stars**, rap, live, hardcore, electro, psychedelic rock, blues rock, indie pop, folk rock, friendsofthekingofrummelpop, lounge, 77davez-all-tracks, **2000s**, romantic, love at first listen, world, trip-hop, **good**, **female vocals**, japanese, **male vocals**, **great**, summer, britpop, funky, best, dreamy, country, love it, english, emo, classical, heard on pandora, memories, drjazzmrfunkmusic, synthpop, post-punk, rnb, thrash metal, **deutsch**, energetic,

Step 2

Now we can create some datasets.

- **Mood**: [“happy”], [“sad”]
- **Female/Male**: [“female vocalists”, “female vocalist”, “female vocals”, “female”], [“male vocalists”, “male vocalist”, “male vocals”, “male”]
- **Quality of content**: [“good”, “awesome”, “amazing”, “great”], [“bad”, “awful”, “terrible”, “garbage”]
- **Origin**: [“american”, “usa”], [“british”, “uk”], [“german”, “deutsch”, “germany”], [“spanish”, “spain”]
- **Rating (out of 10)**: [“0 of 10 stars”], ["1 of 10 stars"], …["10 of 10 stars"]

Step 2

Will use recordings that were assigned the same tag the most.

For each dataset that we want to create:

1. Get list of recordings associated with a specific tag
2. Sort recordings by normalized count associated with them
3. Write recording and class that it is associated with into a CSV file

Each class gets (roughly) the same number of recordings to prevent bias.

After that datasets are ready to be imported into AcousticBrainz.

Step 3

Now we can import datasets into AcousticBrainz and start evaluation.

All datasets that I created are at http://acousticbrainz.org/user/Gentlecat. Some are still being evaluated, but I already got some results:

Mood:

Accuracy: 82.37%

Predicted (%)					Actual (%)
	happy	sad		Proportion	
happy	82.70	17.30	happy	50.38	
sad	17.97	82.03	sad	49.62	

Quality of content:

Accuracy: 71.11%

Predicted (%)					Actual (%)
	bad	good		Proportion	
bad	48.51	51.49	bad	39.30	
good	14.26	85.74	good	60.70	