Marc Siquier - Dataset Creation

```
In [1]: from freesound import freesound
import random
import utils
from utils.run_entity_linking import spotlight
from IPython.core.display import display, HTML

# Get the API key from http://www.freesound.org/apiv2/apply/ (you'll nee
d Freesound user account)
API_KEY='9GUfpnadafRpnB3v76G7M5x9i7X3tNHtZ2a4FrBb'
c = freesound.FreesoundClient()
c.set_token(API_KEY,"token")
```

1) Define a number of audio categories and find audio examples from Freesound for each category

```
# Configure dataset parameters and audio categories
DATASET NAME = 'broad' # Dataset will be saved in a .json file with this
name
DATASET CLASSES = {
    # Must be dictionary with structure like {'class name': 'query terms
   'class name 2': 'query terms 2',... }
    'Acoustic Guitar': 'acoustic guitar',
'Electric Guitar': 'electric guitar',
'Distorted Guitar': 'distorted guitar'
N = 100 # Number of sounds per class
N SOUNDS PER USER = 3 # Do not get more than 3 sounds per user
# Get sound examples from Freesound
dataset = dict()
for name, target_query in DATASET_CLASSES.items():
    print 'Getting sounds for class %s...' % name,
    # Get first page of results
    PAGE_SIZE = 150 # Page size for fs requests
    N_PAGES = int((N * 1) / PAGE_SIZE) # Number of pages to retrieve
    fields = "id, tags, description, username"
    results_pager = c.text_search(
        query=target_query,
        page_size=PAGE_SIZE,
        group_by_pack=1,
        fields = "id, tags, description, username, analysis",
        #descriptors = "lowlevel.mfcc.mean,lowlevel.barkbands.mean",
    all_results = results_pager.results
    # TIP ON AUDIO FEATURES: you can get also audio features extracted i
n freesound by passing a 'descriptors'
    # parameter in the text search function and including 'analysis' in
the fields list
    # (see http://www.freesound.org/docs/api/resources_apiv2.html#respon
se-sound-list):
    # fields = "id, tags, description, username, analysis"
    # descriptors = "lowlevel.spectral centroid,lowlevel.barkbands.mean"
    # e.g.: results_page = c.text_search(query=target_query, ..., fields
=fields, descriptors=descriptors)
    # ...
    # Get extra pages
    for i in range(0, N_PAGES):
        if results_pager.count > (i+1) * PAGE_SIZE:
            results_pager = results_pager.next_page()
            all_results += results_pager.results
    print len(all results)
    # Get only N sounds max per user
    user_sounds_count = dict()
    filtered_results = list()
    random.shuffle(all_results) # Shuffle list of sounds (randomise ord
er)
    for result in all results:
        if result["username"] in user sounds count:
            user_sounds_count[result["username"]] += 1
        else:
            user_sounds_count[result["username"]] = 1
        if user_sounds_count[result["username"]] <= N_SOUNDS_PER_USER:</pre>
            filtered results.append(result)
    # Randomly select N sounds from al results obtained
    if len(filtered_results) >= N:
        selected sounds = random.sample(filtered results, N)
        dataset[name] = selected sounds
```

Getting sounds for class Distorted Guitar... 150 selected 100 sounds out of 106! Getting sounds for class Electric Guitar... 150 selected 100 sounds out of 103! Getting sounds for class Acoustic Guitar... 150 selected 100 sounds out of 137!

2) Explore the dataset (know your data!)

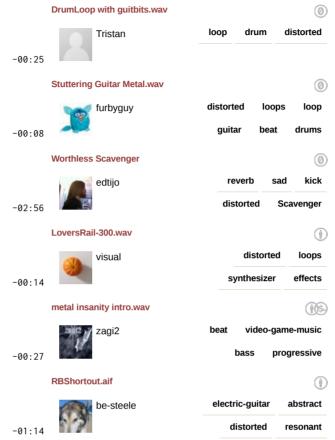
```
In [3]: # Load dataset from saved file
DATASET_NAME = 'broad'
dataset = utils.load_from_json('%s.json' % DATASET_NAME)
N = len(dataset[dataset.keys()[0]]) # Number of sounds per class
print 'Loaded dataet "%s" (%i classes, %i sounds per class)' % (DATASET_NAME, len(dataset.keys()), N)
```

Loaded dataet "broad" (3 classes, 100 sounds per class)

```
In [4]: # Pick some sounds from each category and show players (Freesound embeds
         ) to listen to them
        for class_name, sounds in dataset.items():
            html = "<h3>%s</h3>" % class name
            html += "<h4>Example sounds:</h4>"
            html += utils.generate_html_with_sound_examples([sound['id'] for sou
        nd in sounds][:6])
            class_tags = utils.get_all_tags_from_class(class_name, dataset)
            class keywords = utils.get all keywords from class(class name, datas
        et)
            html += "<h4>Most commons tags tagcloud:</h4>"
            html += utils.generate_html_tagcloud(class_tags, N=100, max_px=50, m
        in_px=10, pow_scale=1.2)
            html += "<h4>Most commons keywords tagcloud:</h4>"
            html += utils.generate html tagcloud(class keywords, N=100, max px=5
        0, min_px=10, pow_scale=1.\overline{2})
            html += "<br>"
            display(HTML(html)) # <- This is pure jupyter notebook AWESOMENESS m</pre>
        agic which renders the HTML in the output of the cell
```

Distorted Guitar

Example sounds:



Most commons tags tagcloud:

reese beat mangled overdriven harmonics Noise n slide loops grit music drums remix guitar-chords muted power-chord drumloop mosh-pit melodic heavy-metal electric distortion Harmonics soundtrack Loop drone distort overdrive hardcore metal Distorted bpm riff slash bass power drum hoover noise delay rhythmic lead-guitar string rock 120bpm synth heavy wild blues rhythm ambient gabber Rock resonant Harmonic indie Guitar experimental hard sound pareidolia

Most commons keywords tagcloud:

guitar chord distortion

Effects Hardstyle Gabber Boss chord overdrive drums microphone reverb riff drum Guitar rock playing kicks Cubase loop tascam style distorted sound signal Amp This guitar amp share electric guitar bpm string Logic Pro acoustic guitar href recorded raw delay Edition unit An bit computer created My sample rel="nofollow" level pitch nice song keyboard simple hard dance processed Recorded Modelling Sound looped http Distorted Ibanez links Here Electro-Harmonix fret short access virus Guitar high https plugins amp Sounds dedicated The Enjoy 4- played July strings noise bass live wave remix Orange Made M-Audio de BPM lot amplifier Series EQ crunchy Check All Riff stuff FX

Electric Guitar

Example sounds:



Most commons tags tagcloud:

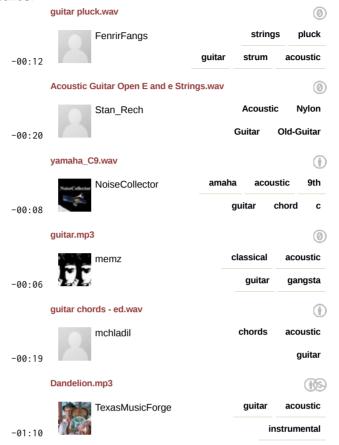
effect instrument 115bpm loud A Electric pan power reverb fuzz stereo processed outer beat Drum 4x4-Records ambient overdrive Electric-Guitar a-string experimental psychedelic chord Lead c time distorsion string stretch palm-mute Clean EDM Loop electric-guitar bending distorted funk clean Power soundscape heavy ■ lick riff e amplifier drums D 90Bpm Drums echo Funk acoustic Snare wah distortion Guitar Kick power-chord environmental-sounds-research noise atmosphere bass bow Instrument sound bpm

Most commons keywords tagcloud:

Electric guitar Two It muted amp disk Distorted sliding style overdrive played audacity Fender reversed Clean music The sid delay Zoom lick pedal Recorded Tube distortion reverb Electric Guitar Guitar chord Orange Ibanez key direct link projects good Simple Boss minor preamp electric forget riff plugging href Logic Pro clean la Hendrix heavy metal processed acoustic playing guitar chord strummed single GT Modelling This looped BOSS amplifier rel="nofollow" recorded Electric distorted bpm Les Paul

Acoustic Guitar

Example sounds:



Most commons tags tagcloud:

quantized film Drum a 6 reverb dry sample-pack ambient Faux pop delay roland-oetter nylon-guitar piano Funk sound Loop acoustic-guitar notes Classical Music instrumental sounds minor atmosphere summer tap flstudio Bass sample chords Nylen Finger-picking Lead banjo short sustain noise folk-quitar note musical-instruments strings folk Electric chord String arpeggio classical picking tuning twang Folk blues strumming base original recording Guitar riff nylon major emotional synth Snare stereo

Most commons keywords tagcloud:

Me Sounds gt playing recording download Logic Pro chords classical guitar chord Synths /12 sounds Acoustic guitar strings https Kontakt mics Lethbridge accompanied nylon Recorded tuning stereo bit licks cmd record This created Tascam D.Major ST bit.ly Sound Tuning Some strummed instrument riff Acoustic Guitar sound music strumming High microphone Guitar Try 2016 reverb The tuned projects These project pop Sennheiser banjo rel="nofollow" sample kind http string acoustic recorded microphones It Folk mic electric noises Made played Reaper