Deliverable D4.7

Second prototype tool for the automatic semantic description of music samples

Grant agreement nr	688382
Project full title	Audio Commons: An Ecosystem for Creative Reuse of Audio Content
Project acronym	AudioCommons
Project duration	36 Months (February 2016 - January 2019)
Work package	WP4
Due date	31st July 2017 (M30)
Submission date	31st July 2017 (M30)
Report availability	Public (X), Confidential ( )
Deliverable type	Report ( ), Demonstrator (X), Other ( )
Task leader	MTG-UPF
Authors	Frederic Font
Document status	Draft (), Final (X)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Table of contents

Table of contents 2

Executive Summary 3

1 Description of the annotation tool 4

1.1 Goals 4

1.2 Implementation 7

1.3 Usage instructions 7

2 Conclusion and future improvements 9

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Executive Summary

As part of the Audio Commons Ecosystem, a number of tools are provided for the automatic analysis of audio content without the need for human intervention. These tools are designed for extracting i) musical audio properties for music pieces and music samples, and ii) non-musical audio properties for any kind of sounds. Work-in-progress versions of these tools have been released in parallel for the first prototype in the Audio Commons Ecosystem.

The current deliverable addresses the automatic extraction of musical audio properties for sound samples and demonstrates the second prototype of the tool for extracting such properties. The tool consists of an Essentia audio extractor which extracts a number of audio properties and a Python script which post-processes some of these properties and renames it to the commonly agreed descriptor names of the Audio Commons Ontology. The current version of the tools also integrates the timbral descriptors developed in deliverable D5.2 and a linked data compatible output format (JSON-LD). The whole ensemble is provided as a Docker container which makes it really easy to be installed and executed on any platform.

The musical properties included in this audio extractor have been selected according to a draft sound schema for the Audio Commons Ontology (which has not yet been included in the ontology itself). The current version of the annotation tool adds some more descriptors with respect to the previous version. Future versions of the tool will include extra descriptors coming from deliverables D4.3 and D4.8 as well as updated versions of the descriptors presented in D5.2 and D5.7 (i.e. the tools for the annotation of musical properties for music pieces and the tools for the annotation of non-musical properties).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

1 Description of the annotation tool

1.1 Goals

The Audio Commons Ecosystem will provide tools for the automatic annotation of different kinds of audio content. These tools are developed and evaluated in in different tasks across work packages 4 and 5. In particular, we will focus on the analysis of:

- Musical properties for music samples
- Musical properties for music pieces
- Non-musical audio properties (timbral models)

The differentiation between music samples and music pieces has been done to tailor some descriptors to the particular context in which they're used. We define music pieces as audio recordings typically corresponding to *complete* songs, while we define music samples as individual or simpler music elements such as single notes, percussive hits, chords, melodies or loops. Note that loops (or short music fragments) can also be classified under music pieces in case they're complex enough.

As part of the development of the Audio Commons Ontology, a sound schema is being developed in which we specify, among others, a number of musical and non-musical properties which could be included in the automatic annotation tools. The following table shows the musical properties that are considered for both music pieces and music samples. Highlighted in orange are the descriptors that were already included in the first version of this deliverable (some of them have been updated and improved for this version). Highlighted in red are the descriptors that were initially planned to be added to the extractor but that have been excluded because are not relevant enough (ac:bitdepth, ac:attack, ac:decay, ac:sustain, ac:release, ac:chord) or can be easily guessed from the file name (ac:format). Marked in green are the descriptors that have been added for the second version of this deliverable, including the non-musical audio properties (also relevant for music samples) from D5.2 First prototype of timbral characterisation tools for semantically annotating non-musical content. Note that new descriptors also include confidence measures for tempo, tonality and pitch estimations. This was suggested as an improvement in the evaluation report in D4.4. The remaining descriptors (not highlighted) are planned to be incorporated in a future release of the tool as they get further developed in task T4.3 and all the annotation tools are finally integrated into a single one.

Name	Description	Relevant for...
`ac:duration`	Duration of audio	Music pieces + Music samples
`ac:format`	Audio format (e.g. wav, mp3)	Music pieces + Music samples
`ac:lossless`	Whether audio file is in lossless codec (1 or 0)	Music pieces + Music samples
`ac:codec`	Codec used for encoding the audio (e.g. pcm_s16le)	Music pieces + Music samples

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

ac:filesize	Size of the file	Music pieces + Music samples
ac:bitrate	Number of bits per second	Music pieces + Music samples
ac:bitdepth	Number of bits per sample	Music pieces + Music samples
ac:samplerate	Number of samples per second	Music pieces + Music samples
ac:channels	Number of channels	Music pieces + Music samples
ac:audio_md5	MD5 checksum of raw undecoded audio payload. It can be used as a unique identifier of audio content.	Music pieces + Music samples
ac:genres	Music genre(s) of the musical content	Music pieces
ac:instruments	Musical instruments present in the recording	Music pieces + Music samples
ac:mood	Mood conveyed by the musical content	Music pieces
ac:tonality	Key and scale (e.g. A minor)	Music pieces + Music samples
ac:tonality_confidence	Confidence of the tonality estimation	Music pieces + Music samples
ac:tempo	Tempo in BPM of the audio signal	Music pieces + Music samples
ac:tempo_confidence	Confidence of the tempo estimation	Music pieces + Music samples
ac:loop	Whether audio file is *loopable* (Boolean)	Music samples
ac:chord	Played chord (e.g. G#m)	Music samples
ac:note_name	Played note name (e.g. C4)	Music samples
ac:note_midi	Played note midi number (e.g. 60)	Music samples
ac:note_frequency	Played note frequency (e.g. 440Hz)	Music samples
ac:note_confidence	Confidence of the note estimation	Music samples
ac:loudness	Loudness value	Music pieces + Music samples
ac:dynamic_range	Dynamic range of audio recording	Music pieces + Music samples
ac:temporal_centroid	Temporal centroid	Music samples

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

ac:attack	Attack length	Music samples
ac:decay	Decay length	Music samples
ac:sustain	Sustain amount	Music samples
ac:release	Release length	Music samples
ac:log_attack_time	Logarithm of the time it takes to reach maximum amplitude of audio signal (good for perceptual attack)	Music samples
ac:single_event	Whether the audio file contains one single *audio event* or more than one (Boolean). This computation is based on the loudness of the signal and does not do any frequency analysis.	Music samples
ac:brightness	Brightness of the analyzed audio in a scale from [0-100]. A *bright* sound is one that is clear/vibrant and/or contains significant high-pitched elements.	Music samples
ac:hardness	Hardness of the analyzed audio in a scale from [0-100]. A *hard* sound is one that conveys the sense of having been made (i) by something solid, firm or rigid; or (ii) with a great deal of force.	Music samples
ac:depth	Depth of the analyzed audio in a scale from [0-100]. A *deep* sound is one that conveys the sense of having been made far down below the surface of its source.	Music samples
ac:roughness	Roughness of the analyzed audio in an undetermined scale. A *rough* sound is one that has an uneven or irregular sonic texture.	Music samples

1.2 Implementation

The annotation tool in particular can be found in the following public Github repository: https://github.com/AudioCommons/ac-audio-extractor. It consists of a "dockerized"[1] Python script which uses the Essentia Python bindings to compute audio descriptors as well as computes some other audio descriptors (the timbral models) in pure Python code. Then it reformats the output to match the property names defined in the Audio Commons sound schema and to generate both standard JSON and JSON-LD (linked data compatible) output analysis files. Essentia (http://essentia.upf.edu) is an audio analysis library developed and maintained at the Music Technology Group in Universitat Pompeu Fabra, coordinators of the Audio Commons Project. More

[1] Docker (http://docker.com) is a multi-platform software that allows the deployment of software into containers. Containers work like lightweight virtual machines which include all the dependencies needed for the software to run. Therefore, "dockerized" programs are easy to run in different platforms as the Docker layer handles dependencies and other configuration parameters.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

information about the Essentia framework is provided in deliverable D4.3 First prototype tool for the automatic semantic description of music pieces.

The tool provides a command line utility to analyze a single audio file and write the output analysis into a JSON file (see next section for instructions). It does not manage the analysis of collections of audio files. This is expected to be done by another script that calls the command line utility and is run in the host machine. That other script is out of the scope of this deliverable.

1.3 Usage instructions

In order to use the analysis tool, Docker will need to be installed in the host system. If docker is installed the steps are quite simple:

1. Clone repository https://github.com/AudioCommons/ac-audio-extractor.
2. `cd` to repository folder and run `docker build -t audiocommons/ac-audio-extractor .`
3. Place the audio files that need to be analyzed in the same repository folder and run `docker run -it --rm -v \`pwd\`:/essentia audiocommons/ac-audio-extractor -i filename.wav -o analysis_output.json`

This will analyze the file `filename.wav` and place the results in `analysis_output.json`. In order to analyze files from other directories, one can either create a symbolic link in the same directory as the repository, or mount a different directory in the Docker container by replacing `` `pwd` `` with the path to that directory.

Running the above commands will produce an output like the following:


```
{
    "duration": 9.241541862487793,
    "lossless": 1.0,
    "codec": "pcm_s16le",
    "bitrate": 705600.0,
    "samplerate": 44100.0,
    "channels": 1.0,
    "audio_md5": "2722ac23a142ce727e0642b0a63c7347",
    "loudness": -28.64586639404297,
    "dynamic_range": 3.432065963745117,
    "temporal_centroid": 0.5782503485679626,
    "log_attack_time": 0.6950863599777222,
    "filesize": 815294,
    "single_event": false,
    "tonality": "G# major",
    "tonality_confidence": 0.5119080543518066,
    "loop": false,
    "tempo": 84,
    "tempo_confidence": 0.42026047706604003,
    "note_midi": 74,
    "note_name": "D5",
    "note_frequency": 608.390625,
    "note_confidence": 0.0,
    "brightness": 60.313207479409286,
    "depth": 16.728879931862544,
    "hardness": 82.90738501480826,
    "roughness": 6.646583836789146
}
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Alternatively, using the JSON-LD output format it would look like:


```
{
    "@context": {
        "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
        "ac": "https://w3id.org/ac-ontology/aco#",
        "afo": "https://w3id.org/afo/onto/1.1#",
        "afv": "https://w3id.org/afo/vocab/1.1#",
        "ebucore": "http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#",
        "nfo": "http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#"
    },
    "@type": "ac:AudioFile",
    "ebucore:bitrate": 705600.0,
    "ebucore:filesize": 529278,
    "ebucore:hasCodec": {
        "@type": "ebucore:AudioCodec",
        "ebucore:codecId": "pcm_s16le"
    },
    "nfo:compressionType": "nfo:losslessCompressionType",
    "ac:audioMd5": "8da67c9c2acbd13998c9002aa0f60466",
    "ac:availableItemOf": {
        "@type": "ac:AudioClip"
    },
    "ac:signalAudioFeature": [
        {
            "@type": "afv:Loop",
            "afo:value": true
        },
        {
            "@type": "afv:Tempo",
            "afo:confidence": 1.0,
            "afo:value": 120
        },
        {
            "@type": "afv:Key",
            "afo:confidence": 0.2868785858154297,
            "afo:value": "G# minor"
        },
        {
            "@type": "afv:TemporalCentroid",
            "afo:value": 0.5078766345977783
        },
        {
            "@type": "afv:MIDINote",
            "afo:confidence": 0.0,
            "afo:value": 74
        },
        {
            "@type": "afv:Pitch",
            "afo:confidence": 0.0,
            "afo:value": 592.681884765625
        },
        {
            "@type": "afv:Loudness",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
            "afo:value": -28.207069396972656
        },
        {
            "@type": "afv:Note",
            "afo:confidence": 0.0,
            "afo:value": "D5"
        },
        {
            "@type": "afv:LogAttackTime",
            "afo:value": 0.30115795135498047
        }
    ],
    "ac:signalChannels": 1,
    "ac:signalDuration": 6.0,
    "ac:singalSamplerate": 44100.0
}
```


We have written a blog post in the Audio Commons website which describes the tool and provides instructions on how to used. It can be accessed here:
https://www.audiocommons.org/2018/07/15/audio-commons-audio-extractor.html

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

2 Conclusion and future improvements

In this demonstrator deliverable we described the second version of an automatic tool for annotating musical properties of music samples. We present the tool as a command line utility that can be given an input audio file and extract relevant musical (and non-musical) properties according to a work-in-progress Audio Commons sound schema. The tool can run on different platforms as it is provided in a Docker container.

A follow-up deliverable to come later this year (D4.10, in October), will report on the evaluation of the prototype presented here, but preliminary experiments already showed and improvement over the previous version. Even if this version of the tool includes significant improvements over the previous one, is is still planned to be updated and incorporate more descriptors from other deliverables in WP4 and WP5. This is a tentative list of improvements that are expected to be included in future iterations of the tool:

- **Adding more descriptors from the sound schema**: incorporate the instrument identification tool that has already been developed as part of task T4.3, as well as trained models for (at least) music genre and potentially some other high-level music properties.
- **Updates in the JSON-LD output**: adding new descriptors will require to update the JSON-LD output to provide support for them. Also, some of the descriptors currently included in the tool (i.e. the timbral models from task 5) are only outputted when using standard JSON format as these have not yet been included in the Audio Commons Ontology and can't be represented in the JSON-LD.
- **Integration with other tools described in D4.7 and D5.7**: the current version of the analysis tool already incorporates some descriptors from WP5, but there are further descriptors described in D4.7 and D5.7 which can also be added to the current tool so that all Audio Commons descriptors are unified in a single place. This will require some changes in the code structure of the tool presented in this deliverable, but will be possibly thanks to the efforts of *dockerizing* all the tools developed so far.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382