

COAL Metadata Extraction for Audio

podcast.mp3 (binary)

...ÓÖç,éá<lá yWRQ@gÂ][2 ~S II○-T FÍ Y†
VøΔHGΩñI „,l sH,íUm9_æg
μ̄:RÁâß÷€Vg U% 3°FU Ó Δ
VÉU¢?O.óùmñ|7áå Á-ÁgòúMñÁé≤kkC
yEŠÅD&„!�ådFéyæjzCJ ÅçààBUlðçÜ
ð\$g_ „Åns[†]-t- ä Aæþpaß=úð „%≤y E‘fózð
h_ Gç_@ð@ „!CE_ t“c[†]çOð „!fløj_ ï
„!ð.ÁÖ_ -_!lx|xE!ÚØ#‰çvñÍð „!x.~AZ5ÆA...
jet:œ\$ññoÖç,éá<lá yWRQ@gÂ][2 ~S II○-T
FÍ Y†,VøΔHGΩñI „,l sH,íUm9_æg...
.....μ̄:RÁâß÷€Vg U% 3°FU Ó Δ
VÉU¢?O.óùmñ|7áå Á-ÁgòúMñÁé≤kkC

Audio-Dateien sind nicht maschinenverstehbar. COAL ändert das.

(COAL = “Web Media Content Analysis Framework”)

- 1 curl -H "accept:text/turtle" "http://example.org/coal/resource?url=http://example.com/testpodcast.mp3" Descriptive RDF for the audio file requested from the server



2

MediaInfo



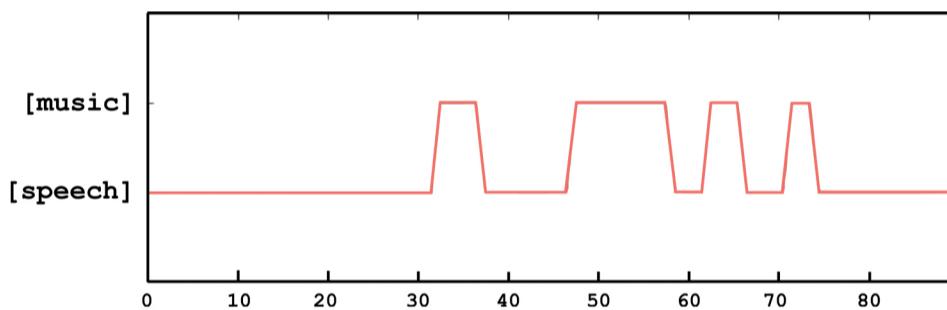
Nachdem die Audio-Datei vom **DownloadWorker** heruntergeladen wurde, extrahiert der **MediaInfoWorker** Informationen wie **Dateigröße und -format, Länge, BPM, Channels, etc..**
Hierzu wird die *MediaInfo-Library* genutzt.

Hierzu wird die *MediaInfo-Library* genutzt.

ID3-Tag



Parallel zum **MediaInfoWorker** wird auch der **ID3TagWorker** gestartet. Dieser nutzt die *mutagen-Library* um in der mp3-Datei gespeicherte Informationen wie **Interpret**, **Titel**, **Album**, **Genre**, **Veröffentlichungsjahr**, etc. zu extrahieren.



Um die Inhalte des Podcast zu segmentieren - also in “speech”- und “music”-Segmente zu unterteilen - muss die Datei als .wav vorliegen. Darum kümmert sich der **ConverterWorker**.

Das Segmentieren selber wird dann vom **SegmentationWorker** durchgeführt. Hierzu wird die *pvAudioAnalysis-Library* verwendet.

Speech Recognition



Die als "speech" erkannten Segmente werden vom **SpeechRecognitionWorker** an die Google Speech API gesendet. Dieses erkennt den **gesprochenen Text** und sendet diesen in Textform zurück. Die empfangenen Inhalte werden dann in die rdf-Datei geschrieben.

Music Recognition

Parallel dazu verarbeitet der **MusicRecognitionWorker** die als "music" erkannten Segmente. Dazu erzeugt er für jedes "music"-Segment einen Fingerprint, welcher an *AcoustID* gesendet wird. Als Ergebnis erhalten wir die *MusicBrainz-ID* des erkannten Songs. Diese können wir nutzen um Informationen wie [Titel, Interpret, etc. des Songs](#) von *MusicBrainz* abzurufen.

SpeechRecWorker

MusicRecWorker

3 Erzeugtes RDF an den Client zurücksenden

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix dc: <http://purl.org/dc/elements/1.1/>
```

```
<http://coal.s16a.org/resource?url=http://static.nico.is/testpodcast.mp3>
    a foaf:Document ;
    dc:identifier "cache/2dc2ca1ca9f8a330b7e9e3aa2d04be46/data.ttl^^xsd:string ;
    foaf:topic "http://static.nico.is/testpodcast.mp3"^^xsd:string .
```

```
<http://static.nico.is/testpodcast.mp3>
dc:format      "audio/mpeg"^^xsd:string
dcterms:extent "1446600"^^xsd:int
dcterms:modified "Thu, 16 Jun 2016 10:28:00"^^xsd:dateTime
ebu:audioChannelNumber "2"^^xsd:int
nie:averageBitrate "128000"^^xsd:int
nie:bitRateMode "CBR"^^xsd:string
nico:duration "00:00:00"^^xsd:string
```

David Kreidler, Nico Böckhoff, Jonathan Schneider, Nico Knoll, Lukas Heilmann

Betreut von: Magnus Knuth, Jörg Waitelonis, Harald Sack