TROMPA: Towards Richer Online Music Public-domain Archives

Deliverable 5.5
Annotation Tools

Grant Agreement nr	770376
Project runtime	May 2018 - April 2021
Document Reference	TR-D5.5-Annotation Tools v1
Work Package	WP5 - TROMPA Contributor Environment
Deliverable Type	R - Report
Dissemination Level	PU - Public
Document due date	30 April 2019
Date of submission	30 April 2019
Leader	UPF
Contact Person	Alastair Porter (alastair.porter@upf.edu)
Authors	Alastair Porter (UPF), Julian Urbano (TUD)
Reviewers	David Weigl (MDW), Emilia Gomez (UPF)

Executive Summary

In this deliverable we describe the requirements and development plan to make tools to collect and store annotations about music content that we work with in TROMPA. These annotations will be used in the project to enrich the music that we are studying in TROMPA, and help us to improve algorithms for the automatic analysis and annotation of this music. We present the design for a graphical interface for creating the types of annotations that are required in the use cases being produced in WP 6. The individual use cases can embed the annotation tools in the applications that they are developing to provide annotation functionality. The annotations will be stored in the Contributor Environment (CE), being built as part of Deliverable 5.1. We will interact with the crowd planning strategies (Deliverable 4.4) to determine the best way of soliciting annotations from users depending on the skills of the users and requirements of a particular annotation task.

In TROMPA, we plan to collect user annotations in the use cases for performers, singers, music enthusiasts, and music scholars. In this deliverable we focus on annotations of music recordings for the first three user groups, but we will also design our annotation storage system to support other kinds of annotations such as annotations of music scores, video recordings, or images so that other use cases can also utilise the framework that we are developing. We will focus on a closed set of annotation types, including free-form annotation labels, annotation with a closed vocabulary of labels, and ratings of items in the TROMPA corpus. Annotations can apply to a single item, or to a segment of an item (e.g., between two timestamps). Some annotations can apply to more than one item or segment (e.g., the similarity between two recordings). We discuss these requirements in Section 2.

The annotation interface will be a web-based interface that can be embedded in web applications. It will provide users with the ability to view audio recordings, listen to them, select segments of recordings, and apply an annotation to a selection. Where possible we plan to reuse existing tools that have been developed within the Music Information Retrieval field. Annotations will be stored in the CE. We will use the Web annotations model to store annotations, an existing open standard for annotations which has been used in previous research by some TROMPA consortium members. Web annotations will give us a standardised way of applying annotations by a certain person and of a certain type to an item that exists in the CE. We will extend the data model and API of the CE to support these annotations. The annotation interfaces and storage are described in Section 3.

In collaboration with Deliverable 4.4 (Hybrid annotation workflows) we will develop strategies to determine which types of annotations are the most important to solicit from users for the tasks that we want to perform in TROMPA. Different types of annotations will be used for different purposes, from the training and evaluation of automatic music description algorithms, to the enrichment of musical content, allowing people to learn more about the music recordings that we store in the CE. These strategies will allow us to choose what is the most important type of annotation that should solicit from a user in order to maximise the benefit to other users and algorithms, and request that those annotations are performed first. We outline this collaboration in Section 4.

Version Log		
#	Date	Description
v0.1	26 April 2019	Initial version submitted for internal review
v0.2	30 April 2019	Revised version after internal review
v1.0	30 April 2019	Final version submitted to EU

Table of Contents

1. Introduction

This deliverable will provide functionality for Task 5.5 of the TROMPA project (Annotation tools). In Task 5.5 we consider "annotation acquisition, feedback to the user, and crowd planning strategies to dynamically adapt the behaviour of the systems as response to the quantity and quality of completed tasks or to the availability and reliability of crowd contributors". Annotations are important for TROMPA because they help us to enhance existing collections of music, adding feedback and insights from the different types of audiences who we target. For example, we see music enthusiasts sharing feedback about performance tips on certain works, or enthusiasts who are interested in seeing the kind of notes and markings that professional orchestra players make on their own copies of scores. Music scholars share notes and insights among themselves to improve their understanding of compositions and performances. From a technical perspective, annotations help us to improve the functionality of automatic systems that we are developing in TROMPA to automatically annotate and analyse music works. Automatic systems may need annotations to use as *ground truth* for building automatic classifiers, requiring a number of examples of each type of annotation such as performance difficulty.

This deliverable, D5.5 is concerned with implementing the features described in Task 5.5. We will focus on three aspects concerned with the implementation of an annotations platform:

- ❖ The development of an interactive tool that runs in a web browser for annotating audio files
- ❖ The development of a schema and storage format for annotations
- ❖ The integration of crowd planning strategies for soliciting annotations from users based on work performed in WP4

This deliverable presents the requirements of an annotation tool that can be used in the TROMPA project to collect annotations from users for a number of the end-user pilots that are being developed in the project. We will also present a proposal for storing the collected annotations in a way that integrates with the existing tools that have already been built to support the research taking part in TROMPA, for example the Contributor Environment (CE). In this deliverable we focus on annotations of audio, however we plan to work closely with other project partners who require the annotation of other modalities (for example, Task 3.4, analysis and annotation of music scores) in order to ensure that our work in the design of the annotation schema can also support annotations for these other types of data. We will also present a preliminary outline of crowd planning strategies that will be used in order to more efficiently obtain annotations from users, and ensure that users that are more suited to providing certain types of annotations can provide them. This first version of the deliverable will present our design plans for the annotation systems. A subsequent version of this deliverable will be published in Month 34, describing the software that was developed and how it was integrated into TROMPA use-cases.

2. Requirements

The software and processes produced as a part of this deliverable are designed to be used by many software demonstrators that will be developed as part of TROMPA in WP 6. This software is designed to be reusable, both by the use cases for which we plan to develop software, and also as an open software library that can be used freely by anyone who has a need for similar annotation tasks. As the result of this deliverable will be used by other members within the TROMPA consortium we

have to ensure that what we produce will fulfil the requirements of the members who need to integrate it into their demonstrators. We will provide functionality for the following end user pilots:

- ❖ Performers
- ❖ Singers
- ❖ Music enthusiasts

In discussion with the task leaders of the relevant end user pilots we have determined requirements for the following types of annotation tasks:

Performers (MDW)

- ❖ Free-form text labels that can be applied to a point in time in a musical score or recording
- ❖ closed-vocabulary annotations around performative aspects (e.g., "wrong note", "missing note")

Singers (UPF, VL)

- ❖ The rating of the difficulty to sing an entire musical score or recording (on a difficult scale)
- ❖ The performance rating of an aspect of audio segment (on a scale, or a binary yes/no annotation)
- ❖ The similarity between two sections of recordings

Music enthusiasts (UPF)

- ❖ Folksonomy tagging of audio recordings or segments
- ❖ Free-form text labels applied to audio recordings or segment
- ❖ The similarity between two sections of recordings
- ❖ The rating of an audio segment

2.1. General requirements

The tools that are generated as a part of this task must integrate with the rest of the TROMPA project, including the Contributor Environment. In order to integrate with other parts of the overall project we include these additional requirements:

- ❖ End user pilots will be developed as web applications, and so the annotation tools that we will build must use standard web technologies (Javascript and HTML).
- ❖ Annotations need to be related to items that exist in the TROMPA CE
- ❖ Annotations need to be assigned to a user, different users can annotate the same item

2.2. Requirements of other pilots

In discussion with task leaders of other end user pilots not directly related to our work in this task, we are aware of other types of annotations that are required by these pilots. This includes the ability to provide annotations for media other than music recordings, such as music scores, images, or videos.

3. Annotation tools

This section outlines the graphical annotation tools that we will develop, as well as the system that we will use to store annotations.

3.1. Web-based annotation tool

We will develop an annotation tool that can be used in the web based user pilots outlined in Section 2. We will use React[1], an industry standard Javascript library for developing interactive web applications. This tool will allow users to create different types of annotations related to audio content stored in the TROMPA Contributor Environment. We will support the following types of annotations:

❖ Free-form labels of any length
❖ Labels chosen from a fixed vocabulary relevant to the task
❖ Ratings (e.g. difficulty or similarity)

Annotations can be applied to one or more music recordings, either as a whole or only a segment of a recording. The annotation tool will provide graphical interfaces to allow users to make an annotation. This will include an interface to show graphical representations of audio recordings, and input interfaces to select an option, choose a rating, or enter text. The tool will also include functionality to retrieve data from the Contributor Environment and to submit annotations back, building on the work started in Task 5.3. Use cases developers will be able to embed the graphical components of this annotation tool into their software in order to allow users to create or view annotations.
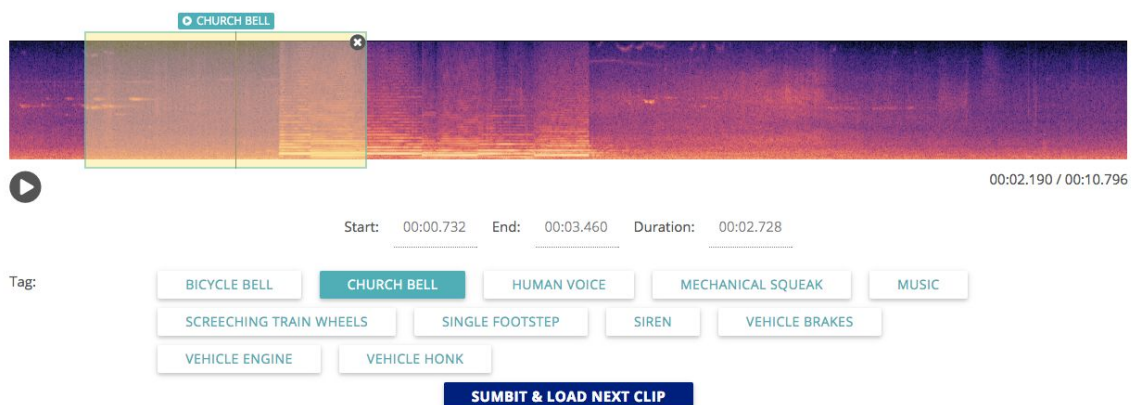
Figure 3.1. An example of the CrowdCurio audio annotator[2]

A key goal of the TROMPA project is to contribute content back to online music public domain archives. As a parallel to this we see the value of contributing to existing software projects where possible. In the Music Information Retrieval field there have been a number of projects developed that are used for the annotation of audio. We hope to build on existing tools where possible to improve their functionality. The Crowd Curio audio annotator (Figure 3.1, M. Cartwright et al. 2017) is a Javascript-based interface for viewing audio recordings and making annotations. This annotation interface can be used to send annotation information to any arbitrary storage system. We see this as a useful resource to use to start building the annotator interface. UPF has experience in modifying this library for use in a similar annotation project[3].

---

[1] https://reactjs.org/
[2] image from https://github.com/CrowdCurio/audio-annotator
[3] https://github.com/MTG/similarity-annotator

3.2. Storage of annotations

Annotations that are produced from the use cases should be stored in a way that makes it easy to relate them to items in the Contributor Environment, and to make it easy for other use cases or researchers to obtain these annotations for further research or development.

In order to facilitate this we will store annotations within the Contributor Environment, being developed in TROMPA as part of WP5. This infrastructure, described in more detail in Deliverable 5.1 consists of a graph database which stores relations between items on the Web, such as musical scores or audio recordings, as well as URI references to these resources, and additional metadata obtained from external sources and metadata created in TROMPA. An application programming interface (API) allows use cases to access this data over the Internet.

Within the TROMPA consortium we have experience storing similar types of annotations using the Web Annotations Data Model (*Web annotations*) standard (Weigl and Page 2017). This model defines a structured format for expressing annotations of connected resources.

Key concepts in web annotations can be linked to content that we are working with in TROMPA:
- ❖ An **ID** representing the annotation
- ❖ One or more **Targets** which refer to items being annotated, represented as URIs. These would represent items in the Contributor Environment, which can already be identified by a URI. Sections of items such as a time range of a recording can be represented by the media fragments model[4]
- ❖ Zero or more **Bodies** which contain information that the annotation associates with its target(s). For example, a tag, description, or rating
- ❖ An optional **Motivation** describing the purpose for which the annotation was made. The web annotation specification defines a number of fixed motivations[5], but this list can be extended by use of the Simple Knowledge Organization System model (SKOS)[6]
- ❖ An optional intended **Audience** to whom the annotation is targeted, for example music scholars or software algorithms

```
{
  "@context": "http://www.w3.org/ns/anno.jsonld",
  "id": "http://trompamusic.eu/3f319024-1891-4c68-8a1c-5814800dffee",
  "type": "Annotation",
  "motivation": "commenting",
  "bodyValue": "aSemanticTag",
  "target": {
    "id": "http://trompamusic.eu/dba9f66a-51fe-4254-a006-4501118beca3",
    "type": "Sound"
  }
}
```

Figure 3.1. An example of a web annotation in JSON-LD format

---

[4] https://www.w3.org/TR/media-frags/
[5] https://www.w3.org/TR/annotation-vocab/#motivation
[6] https://www.w3.org/2009/08/skos-reference/skos.html#broader

The TROMPA Contributor Environment storage system uses neo4j[7], a graph database which is suited to storing information represented by RDF triples, like Web Annotations. We will extend the CE data model to include concepts that allow us to store web annotations. We will extend the GraphQL interface that is used to communicate with the CE to allow our annotation tools and other external software to read and write annotations. Because the web annotations model defines a generic annotations system, the implementation that we will develop in the CE can also be used by other tasks within TROMPA, such as digital score edition (T5.2), which requires the functionality to create annotations of digital music scores.

4. Crowd planning strategies

The goal for involving the crowd in annotation activities is twofold:
- ❖ The continuous evaluation and improvement of technologies developed within TROMPA.
- ❖ The combination of humans and algorithms for the multimodal description of music.

Both objectives imply the participation of humans at various levels of expertise, for various annotation tasks, and for various music artifacts. An important aspect of soliciting annotations is to decide who to ask, to perform what task, and for which music items. We can identify three strategies to satisfy the above goals:
- ❖ Evaluation. Both humans and automatic algorithms may be evaluated in some way to assess how well they perform for some task. Of special importance is the evaluation of algorithms, so that we quantify their expected quality in the automatic description of music artifacts at scale. Such evaluations typically involve comparisons with reference annotations or ground truth, which is scarce or non-existent for rare or new tasks. The evaluation problem therefore becomes an estimation problem, bearing some degree of uncertainty that can be narrowed with the incorporation of more annotations. Therefore, one strategy to allocate crowds to tasks is to do it such that the uncertainty in the evaluation of technology is minimized.

- ❖ Training. Once evaluations are carried out, one may decide that some specific technology is still not mature enough or appears to produce biased results. While sometimes the way to correct for this requires manual interventions in computer code, for machine learning technology one is often more interested in providing algorithms with more data to train their underlying models. Therefore, a second strategy to allocate crowds to tasks is to do it such that the new annotations will help develop better technology by targeting critical points.

- ❖ Enrichment. After the performance of algorithms is estimated, they may be used for large-scale description of music artifacts. These algorithms are of course fallible, so their output similarly bears some degree of uncertainty. In particular, some items will prove particularly difficult to describe if for instance a diverse set of algorithms produce very different outputs. Therefore, a third strategy to allocate crowds to tasks is to do it such that the new annotations will reduce the uncertainty in the target multimodal description that algorithms may bring if they disagree with each other.

---

[7] https://neo4j.com/

The ultimate strategy would be that processes that solicit annotations for items in the CE decide which of the three strategies above would be the most beneficial for the system, such that it could switch among them when necessary. Section 3 of deliverable D4.4 provides more details.

5. Planning

This deliverable will be updated again in Month 34, containing a detailed overview of the software components and crowd planning strategies that will have been developed for this task. However, the software components developed in this deliverable will also need to be integrated into other deliverables from Work Package 6 due in Month 24. Therefore, we set ourselves an additional deadline of Month 20 to provide functional software to the other deliverables. This will enable us to solicit feedback from the partners responsible for these deliverables and make any required changes in the software and processes that we build. We will evaluate in the final version of this deliverable if we were successful in delivering functional software to the partners responsible for those deliverables.

6. References

6.1 Written references

M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacConnell, E. Law, J. Bello, and O. Nov. "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations." In Proceedings of the ACM on Human-Computer Interaction, 1(1), 2017.

Weigl, D., & Page, K. (2017). A framework for distributed semantic annotation of musical score: "Take it to the bridge!". International Society for Music Information Retrieval.

6.2 List of abbreviations

Abbreviation	Description
CE	Trompa Contributor Environment
JSON-LD	JSON-Linked Data, an extension to the JSON structured data format to allow semantic-web annotations to be embedded in the data
MDW	Universität für Musik und darstellende Kunst Wien, Trompa Partner
SKOS	Simple Knowledge Organization System
UPF	University Pompeu Fabra, Trompa Partner
VL	Voctro Labs, Trompa Partner