# Automatic Ontology Generation Based On Semantic Audio Analysis

Kolozali, Sefki

# Automatic Ontology Generation Based On Semantic Audio Analysis

**Şefki Kolozali**

Centre for Digital Music

School of Electronic Engineering and Computer Science

Queen Mary, University of London

Submitted to the University of London in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Queen Mary University of London

2014

# Abstract


Ontologies provide an explicit conceptualisation of a domain and a uniform framework that represents domain knowledge in a machine interpretable format. The Semantic Web heavily relies on ontologies to provide well-defined meaning and support for automated services based on the description of semantics. However, considering the open, evolving and decentralised nature of the Semantic Web – though many ontology engineering tools have been developed over the last decade – it can be a laborious and challenging task to deal with manual annotation, hierarchical structuring and organisation of data as well as maintenance of previously designed ontology structures. For these reasons, we investigate how to facilitate the process of ontology construction using semantic audio analysis.

The work presented in this thesis contributes to solving the problems of knowledge acquisition and manual construction of ontologies. We develop a hybrid system that involves a formal method of automatic ontology generation for web-based audio signal processing applications. The proposed system uses timbre features extracted from audio recordings of various musical instruments.

The proposed system is evaluated using a database of isolated notes and melodic phrases recorded in neutral conditions, and we make a detailed comparison between musical instrument recognition models to investigate their effects on the automatic ontology generation system. Finally, the automatically-generated musical instrument ontologies are evaluated in comparison with the terminology and hierarchical structure of the Hornbostel and Sachs organology system. We show that the proposed system is applicable in multi-disciplinary fields that deal with knowledge management and knowledge representation issues.

*Dedicated to my family*

# Acknowledgements

I would like to thank everyone in the Centre for Digital Music at QMUL. It has been a great privilege to spend several years in this research group, and its members will always remain dear to me. I would like to express the deepest appreciation to my supervisor, Professor Mark Sandler, for his guidance and support during my time here, and for always making funding available for travel to conferences. I must also acknowledge the support of fundings from EPSRC, OMRAS-2 and NEMA projects.

My special thanks goes to Mathieu Barthet and George Fazekas for not only reading drafts of this thesis, but also for the many fruitful collaborations we established. You both have been great friends and mentors to me. Thanks to Chris Cannam for helping me on almost every technical problem that I faced. I would also like to thank Dan Tidhar for sharing his extensive musical knowledge with me.

There are of course many more people I want to thank. In no particular order: Kurt Jacobson, Becky Stewart, Katy Noland, Matthias Mauch, Amelie Anglade, Daniele Barchiesi, Steven Hargreaves, Thomas Wilmering and all the people who have been part of C4DM while I've been here. Finally I wish to thank my parents for giving me every opportunity I could have hoped for. Everything I have done in the last few years has been underpinned by the loving support of my family.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

---

## 1.1 Research Aims

Ontologies have grown to be a dominant subject in Computer Science serving as explicit conceptual knowledge models to present available domain knowledge to information systems in a meaningful way. In an effort to annotate websites, they play a key role in the vision of the Semantic Web. In recent years, the World Wide Web has gone through rapid development both technologically and in its popularity. It became closely integrated with our lives. The Web exposes vast amounts of resources including music, photos, video and text contained in unstructured web documents.

However, these documents cannot be interpreted and used by machines directly, since standard Hyper Text Markup Language (HTML) documents, while being machine readable, do not provide machine readable information about their content. For this reason, the Web presents an important challenge in information management. The Semantic Web was conceived in order to resolve these issues by creating a machine interpretable web of data as an extension to the current Web.

The concept of the Semantic Web was initially proposed by Tim Berners Lee [Berners-Lee *et al.*, 2001] in order to enable search through explicit specifications of meaning in the content of web pages. Creating Ontologies to enable formalised description and linking of resources within a particular application domain is among the first steps towards building this new Web.

The semantic interpretation of music audio analysis relies heavily on the availability of formal structures that encode relevant domain knowledge. Many research groups built Ontologies manually to represent different types of data (e.g. music data, social data) within the formation of the Semantic Web. Some examples of Ontologies in the music domain are the music Ontology (MO) and the music performance Ontology, grounded in the MO [Raimond *et al.*, 2007; Fazekas *et al.*, 2010]. The use of Ontological models to access and integrate knowledge repositories is an important contribution, improving knowledge-based reasoning and music information retrieval (MIR) systems alike [Abulaish, 2008]. There are also significant benefits for the discovery of cultural heritage by exchanging data among diverse knowledge repositories, such as musical instrument museums, libraries, institutions or repositories. However, knowledge management in the domain of musical instruments is a complex issue, involving a wide range of instrument characteristics, for instance, physical aspects of instruments such as different types of sound initiation, resonators, as well as the player-instrument relationship. Representing every type of sound producing material or instrument is a very challenging task since musical instruments evolve with time and vary across cultures. The domain of musical instruments is broad and dynamic, including both folkloric instruments made out of non-manufactured items (e.g. blades of grass or conch shells), and new instruments relying on high technology (e.g. iPad app). Although much work has been done on instrument classification in organology, there is currently no classification scheme encompassing such complexity and diversity [Kartomi, 2001]. Thus, there is a need for automated systems that overcome these problems in knowledge management and Ontology engineering.

## 1.2 Research Questions

In order to solve these issues stated above, we developed a hybrid system for automatically generating ontologies relying on the acoustical analysis of isolated notes and solo performances played on various musical instruments. The utility of the system which combines information retrieval and semantic web technologies will be demonstrated through the following investigations:

- *What components are required and how these components should interact with each other to automatically obtain ontologies based on semantic audio analysis?* Most of the systems proposed to date benefit from semantic richness of textual data. Our system will address ways to exploit semantic audio for the purpose of automatic generation of ontologies.

- *How can we relate acoustic timbral descriptors and ontology definitions in order to represent them as ontologies?* The system should be able to identify musical concepts and their associations analysing audio content. We will undertake the musical instrument identification task using the cutting-edge music analysis approaches to extract conceptual metadata from musical sounds.

- *What are the effects of acoustic timbral descriptors and classifiers on the ontology outputs?* Cepstral analysis methods have been used extensively in speech and music analysis over the past few decades. Our aim will be to identify the effects of different number of cepstral features and classifier parameters on the ontology generation process and outputs.

- *What are the knowledge representation issues that exist in the traditional instrument taxonomies? How can we overcome these issues in our system?* Traditional designs based on taxonomy trees which lead to ill-defined knowledge representation, especially in the context of an ontology for the Semantic Web. We will investigate these issues to obtain well-defined musical instrument.

- *What are the evaluation techniques used for ontologies and which ones can be used to eval-*

*uate our system? How can we compare two given ontologies?* An essential element that makes a specific discipline or approach scientific is the capability to assess and compare outcomes in the area. We will explore the description of measures and methodologies for the evaluation of the generated ontologies.

## 1.3 Thesis Overview

We propose a hybrid system in order to automatically obtain ontologies for musical instruments based on their sound spectral structure. We describe and evaluate a supervised system composed of two main processing layers: $(i)$ content-based audio analysis involving feature extraction and machine learning sub-layers, and $(ii)$ conceptual analysis layer where we analyse and represent the obtained information in a graphical form. Most of the automatic ontology generation systems to date, such as the ones described in section 5.1, benefit from the semantic richness of the textual data by using the natural language processing techniques. However, it is a fact that the perception and interpretation of sound is an important aspect of the categorisation of musical instruments. Hence, there is a great deal of advantage of utilising audio data in an effort to ground musical knowledge on the Web and in the real world. This will likely to encompass a greater understanding of the complex characteristics of musical and natural sounds.

There is a range of levels of control or involvement that people can have in a system. It is worth to point out that the proposed system is an automatic system which carries out fixed functions on some available prior knowledge without the intervention of an ontology engineer. Nevertheless, it is not a dynamic nor an adaptive system to learn about new instruments and re-design ontology. The approach presented in this thesis provides a novel contribution by presenting a unified methodology for automatic ontology generation based on semantic audio analysis. Specifically, the focus is on preserving consistency between the acoustic descriptors computed from waveforms, the algorithms used for building instrument models, and conceptual analysis techniques used for automatically obtaining ontologies.

In chapter 7, we review ontology evaluation techniques, and using various parameters on timbral descriptors (i.e. MFCCs and LSFs) and classifiers (i.e. MLP and SVM) provide a comprehensive statistical analysis regarding the effects of the audio analysis component on the ontology generation process. We examine the obtained ontologies through a comparison to a gold-standard ontology that has been built by taking H-S system as a base.

In addition to the proposed system, we also argue that the traditional designs based on taxonomy trees, widely used to represent musical instruments, lead to ill-defined knowledge representation, especially in the context of an ontology for the Semantic Web. In chapter 4, we examine knowledge representation issues of musical instruments on the Semantic Web, by taking musical instrument classification schemes into account; and an assessment of the OWL representations of these classification schemes using SPARQL queries.

Publishing structured data in an open format that shares a common conceptual framework is among the important goals of the Semantic Web. In section 3.4.1, we contribute to the Web of Data by publishing a large set of music similarity features produced by the SoundBite playlist generator tool. We explain the process of collecting, organising and publishing the dataset which can be accessed via a SPARQL end-point and used in Linked Data services.

## 1.4 Thesis Outline

The core of this thesis focuses on development of an automatic ontology generation system based on semantic audio analysis. In the following, the thesis is outlined by summarising each chapter.

**Chapter 1: Introduction.** We introduce and state the problem studied in this thesis. The several tasks of Ontology engineering such as manual annotation, hierarchical structuring and organisation of data can be laborious and challenging. Therefore, we investigate how the process of creating

Ontologies can be made less dependent on human supervision by exploring concept analysis techniques in a Semantic Web environment.

**Chapter 2: Ontologies.** Beginning with details on philosophical and mathematical background by taking a cognitive science stance, and continuing with classification of Ontologies, this Chapter introduces foundational knowledge for Ontologies.

**Chapter 3: The Semantic Web.** We provide background knowledge on Ontologies from the Semantic Web standpoint. Therefore, we highlight the connection between the Semantic Web and Ontologies, and outline the semantic richness of Ontology representation models, the Semantic Web technologies for Ontology representation, and Ontology engineering on the Semantic Web.

**Chapter 4: The Semantic Web and Musical Instrument Ontologies.** We outline our investigation on knowledge representation issues of musical instruments on the Semantic Web, by taking musical instrument classification schemes into account; and an examination of the OWL representations of these classification schemes using SPARQL queries.

**Chapter 5: Fundamentals for Automatic Ontology Generation System.** We discuss the automatic Ontology generation frameworks based on the types of input data and the proposed Ontology generation tools to date. We also review basic methods for the content-based audio analysis, machine learning, and conceptual analysis techniques, which are relevant to both the previously existing and the original approaches to automatic Ontology generation framework discussed later on in this thesis.

**Chapter 6: A Framework for Automatic Ontology Generation System.** The general architecture of the proposed system is described in this Chapter. We also present the analysis of parameters for feature extraction and classification algorithms, and application tools that have been used in the experiments.

**Chapter 7: Evaluations.** We discuss our Ontology evaluation methodology, providing similarity metrics that have been used as a basis for evaluating the automatically generated Ontologies against a gold standard. We empirically support the major theoretical findings of the preceding sections in this Chapter. More precisely, we provide a thorough evaluation of the proposed system for measuring the musical instrument recognition using Analysis of Variance (ANOVA). For the evaluation of the generated Ontologies, Hornbostel and Sachs was considered as the basis for instrument terminology and initial hierarchical structure. Additionally, we also give a detailed evaluation for the conceptual analysis results of the generated Ontologies using Multivariate Analysis of Variance (MANOVA).

**Chapter 8: Conclusions.** We summarise, conclude and discuss directions for further research in terms of a set of open questions and possible technical improvements of the proposed solutions.

## 1.5 Published Work

### Conference and Workshop Proceedings

- Şefki Kolozali, György Fazekas, Mathieu Barthet, and Mark Sandler, A Framework for Automatic Ontology Generation Based on Semantic Audio Analysis, *In Proc. 53rd International Conference of the Audio Engineering Society on Semantic Audio*, January, 2014.

- Şefki Kolozali, Mathieu Barthet, and Mark Sandler, Knowledge Management On The Semantic Web: A Comparison of Neuro-Fuzzy and Multi-Layer Perceptron Methods For Automatic Music Tagging, *In Proc. 9th International Symposium on Computer Music Modeling and Retrieval*, pp. 220-231, 2012.

- Şefki Kolozali, György Fazekas, Mathieu Barthet, Mark Sandler, Knowledge Representation Issues in Musical Instrument Ontology Design, *In Proc. 12th International Society for Music Information Retrieval Conference*, pp. 465-470, 2011.

- Mathieu Barthet, Amélie Anglade, György Fazekas, Şefki Kolozali, Robert Macrae, Music recommendation for music learning: Hotttabs, a multimedia guitar tutor, *In Proc. Workshop on Music Recommendation and Discovery (WOMRAD), colocated with ACM RecSys*, pp. 7-14, 2011.

- Şefki Kolozali, Mathieu Barthet, György Fazekas, Mark Sandler, Towards the Automatic Generation of a Semantic Web Ontology for Musical Instruments, *Semantic Multimedia Lecture Notes in Computer Science*, Vol 6725, 2011, pp 186-187.

- Dan Tidhar, György Fazekas, Şefki Kolozali, Mark Sandler, Publishing Music Similarity Features on the Semantic Web, *In Proc. The 10th International Symposium on Music Information Retrieval*, pp. 447-452, 2009.

**Journal Papers**

- Şefki Kolozali, Mathieu Barthet, György Fazekas, Mark Sandler, Automatic Ontology Generation for Musical Instruments based on Audio Analysis, *IEEE, Transactions on Audio, Speech and Language Processing*. (Forthcoming)

# Chapter 2

# Ontologies

---

This chapter is organised in three parts. We provide an overview of the origins of ontologies from a philosophical point of view in Section 2.1. We then explain the formal definitions of mathematical modelling of Ontologies with an engineering point of view in Section 2.2. Finally, in Section 2.3 we discuss the classification of ontologies in the context of computer science.

## 2.1 Ontologies — Knowledge and Language

The term Ontology has its origin in philosophy and can be described as *science of existence* or “study of being”. Etymologically, the term “ontology” comes from the Greek, where the word “onto-” means “to be” and the word “logia” means “word, reason or thought”. The term of “ontology” was coined in the 17th century by the philosophers Jacob Lorhard in his Ogdoas Scholastica and Rudolf Göckel in his Lexicon philosophicum [Guizzardi, 2007].

An ontology is based on explicit specification of a conceptualisation of the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them. Thereby, a conceptualisation is an abstract simplified view of the world [Gruber, 1993]. The observation of the physical world is an important aspect of conceptualisation of knowledge. The ability to conceptualise objects and events in the world was one necessary building block for the evolution of human knowledge and language. However, with the increasing capacity of the human mind, languages became progressively more complex and ambiguous due to polysemic structure. This ancient discipline, ever since its inception some 2500 years ago, has led philosophers, to explorations of various ontological theories. For instance, the notions such as *categories*, as well as the super-concept and sub-concept referring as "genus" and "subspecies", was first introduced by Aristotle the Greek; and also presented the idea that ten basic categories could be used for classifying anything that may be said or predicted about anything: *substance, quality, quantity, relation, activity, passivity, having, situatedness, spatiality* and *temporality* [Ross *et al.*, 1924]. Another theory that involves categories for the logically possible ways of combining relationships in a proposition or judgement – was introduced by Kant. The theory aims to classify anything using only twelve categories. Later, a meta-level principle, that was based on these twelve categories, had been proposed by Peirce to generate new categories repeatedly. The principle was based on three parts, namely, Firstness, Secondness, and Thirdness: Firstness is determined by qualities inherent in something, Secondness by a relation or reaction directed toward something else, and Thirdness by some mediation that brings multiple entities into relationship (see [Sowa, 1999; Smith, 2004] for a thorough review).

Terms must be explicitly defined to be understood and categorised into corresponding concepts. However, to deal with the inevitable ambiguities of language caused by an implicit exchange of different meaning is a very complicated task. Understanding such ambiguities of knowledge and language was a topic that concerned Gottlob Frege (1848-1925), a German philosopher and logician. Frege introduced the distinction between thought content and referent; his study was later popularised by Ogden and Richard's diagram (see Fig 2.1) knowns as the "meaning triangle" or "semiotic triangle" [Campbell *et al.*, 1998; Lycan, 2006; Richard, 2006].

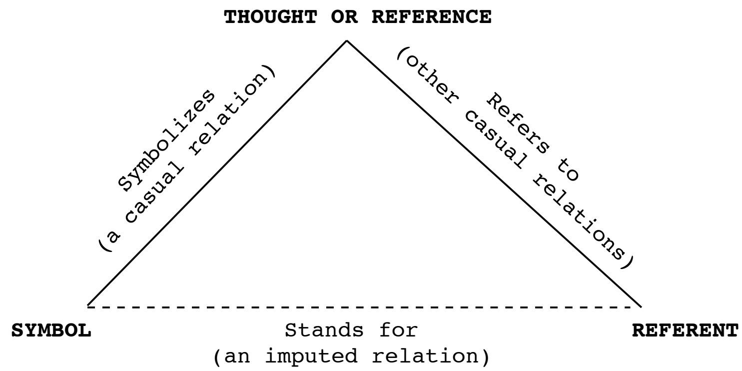



Figure 2.1: The meaning triangle is a model that depicts how linguistic symbols are related to the objects they represent. The diagram is adapted from [Ogden & Richards, 1923]

The meaning triangle illustrates that although a thought (concept) can be inspired by a referent (thing or object), there is no direct relationship between a symbol and a referent. An interpreter, therefore, needed to utilise a symbol to invoke a corresponding thought regarding a referent. For people with the appropriate contextual information who know that the name of a cat is "Yojo", for instance, it is apparent that the name "Yojo" (i.e. symbol) evokes the concept of a cat (i.e. reference) and furthermore denotes a specific cat (i.e. referent) in the world [Sowa, 1999]. It is essential to explicitly define concepts to avoid this sort of issue in communication among interpreters; for instance, two interpreters use different symbols or words for the same thought without knowing that they both refer to the same object. This kind of confusion might cause a serious complication if machines are in charge of any critical task.

The adoption of ontologies was stimulated by the requirement for knowledge representation and management in various fields, such as software engineering, database design and artificial intelligence (AI). This formalised approach to information codification permits abstract representation of the world by referring to real objects. Thus, an agent can use this knowledge to perform any task autonomously and systematically by making decisions and incorporating knowledge in a computational environment. In the next section, we will present the mathematical definitions of ontology modelling primitives as a mathematical foundation of the automatic ontology generation techniques.

## 2.2 Ontologies — Mathematical Definitions

Ontologies have been used in a variety of areas in computer science. The necessity for knowledge sharing and reuse in AI triggered the first noteworthy rise in the formal representation of domain knowledge. The proposed aim of the Advanced Research Projects Agency (ARPA) — Knowledge Sharing Effort project was to enable knowledge to be shared between agents with a common syntax and semantics to reach an agreement on the communicated subject. Ontologies, therefore, have been developed along the same lines as a language, providing axiomatic and textual definitions representing concepts, objects, and other entities along with the relationships among them that are presumed to be present in a certain area of interest. When agents decide to use an ontology, they commit to communicate in this language in accordance with these definitions which specify use of a specifically developed vocabulary. The problem in constructing such a communication language is to match abstract theories to the physical world. To describe the formal ontology structure, we will be using the mathematical definitions in line with the definitions in [Bloehdorn *et al.*, 2005] throughout this thesis.

The mathematical definition of an Ontology is given below which can easily be mapped onto existing ontology representation languages.

**Definition 1.** (Ontology). An *ontology* is a structure $\mathcal{O} := (\mathcal{C}, \leq_{\mathcal{C}}, \mathcal{R}, \sigma_{\mathcal{R}}, \leq_{\mathcal{R}}, \mathcal{A}, \sigma_{\mathcal{A}}, \mathcal{T})$

consisting of

- four disjoint sets $\mathcal{C}, \mathcal{R}, \mathcal{A}$ and $\mathcal{T}$ whose elements are called concept identifiers, relation identifiers, attribute identifiers and data types, respectively,
- a partial order, $\leq_{\mathcal{C}}$, on $\mathcal{C}$, is called concept hierarchy or taxonomy,

- a function $\sigma : \mathcal{R} \to \mathcal{C} \times \mathcal{C}$, is called signature,
- a partial order, $\leq_{\mathcal{R}}$, on $\mathcal{R}$, is called relation hierarchy,
- a function $\sigma_A : \mathcal{A} \to \mathcal{C} \times \mathcal{C}$, is called attribute signature.

Ontologies can be designed in a hierarchical structure, that links entities, such as concepts and relationships, either horizontally or vertically. In mathematics, especially in order theory, a partially ordered set formalises the sequencing and arrangement of the nodes of a graph where the parent node is at the top level and the children of a corresponding node are at the same level, below their parent. The mathematical definition of this hierarchical structure is given by:

**Definition 2.** (Subconcepts and relations). If $c_1 \leq_{\mathcal{C}} c_2$, for $c_1, c_2 \in \mathcal{C}$, then $c_1$ is a subconcept of $c_2$, and $c_2$ is a superconcept of $c_1$. If $r_1 \leq_R r_2$, for $r_1, r_2 \in \mathcal{R}$, then $r_1$ is a sub relation of $r_2$, and $r_2$ is a super relation of $r_1$.

**Definition 3.** (Direct subconcepts and relations). If $c_1 \leq_{\mathcal{C}} c_2$ and there is no $c_3 \in \mathcal{C}$ with $c_1 \leq_{\mathcal{C}} c_3 \leq_{\mathcal{C}} c_2$. Then $c_1$ is a direct subconcept of $c_2$, and $c_2$ is a direct superconcept of $c_1$. Direct superrelations and subrelations are defined analogously.

Given the identifiers of concepts, a natural step is to attempt to identify and describe relationships between concepts. Defining relationships among concepts is a central feature of almost any knowledge extraction and management task. For binary relations, the relationships of *domain* and *range* is defined as follows:

**Definition 4.** (Domain and Range). For a relation $r \in \mathcal{R}$ with $|\sigma(r)| = 2$, we define its *domain* and its *range* by $dom(r) := \pi_1(\sigma(r))$ and $ran(r) := \pi_2(\sigma(r))$. For two relations $r_1, r_2 \in \mathcal{R}, r_1 \leq_{\mathcal{R}} r_2$ implies $\pi_i(\sigma(r_1)) \leq_{\mathcal{C}} \pi_i(\sigma(r_2))$, where $\pi(\sigma(.))$ denotes the i-th argument specified by $\sigma(.)$.

In what follows, we provide an explicit representation of a lexical level of ontology structure, $\mathcal{O}$, which can be defined as follows:

**Definition 5.** A lexicon for an ontology $\mathcal{O}$ is a structure

$$\mathcal{L} := (\mathcal{L}_C, \mathcal{L}_R, \mathcal{L}_A) \tag{2.1}$$

three sets $\mathcal{L}_C$, $\mathcal{L}_R$, and $\mathcal{L}_A$ whose elements are called lexical entries for concepts, relations, and attributes, respectively. An ontology is pair with lexicon;

$$(\mathcal{O}, \mathcal{L}) \tag{2.2}$$

where $\mathcal{O}$ is an ontology and $\mathcal{L}$ is a lexicon for $\mathcal{O}$. Ontologies formalise the intensional aspects of a domain. The extensional aspects are provided by knowledge bases, which contain assertions about instances of the concepts and relations.

**Definition 6.** A knowledge base is a structure for an ontology;

$$\mathcal{KB}_O := (\mathcal{I}, i_{\mathcal{C}}, i_{\mathcal{R}}, i_{\mathcal{A}}) \tag{2.3}$$

consisting of

- a set $\mathcal{I}$ whose elements are called instance identifiers.
- a function $i_{\mathcal{C}} : \mathcal{C} \to \mathfrak{B}(\mathcal{I})$ is called concept instantiation,
- a function $i_{\mathcal{R}} : \mathcal{R} \to \mathfrak{B}(\mathcal{I}^+)$ is called relation instantiation with $i_{\mathcal{R}}(r) := i_C(dom(r)) \times i_C(ran(r))$, for all $r \in \mathcal{R}$. The function $i_R$ is called relation instantiation.

**Example 1.** To clarify the definitions, this section contains an ontology example shown in Figure 2.2. The ontology describes a musical item using the Music Ontology [Raimond, 2008]. It is a simple example, but noticeably helps to explain the basic ontology elements. The concepts are depicted as ellipses, relations as arrows lines, and instances as round boxes. A relation has an

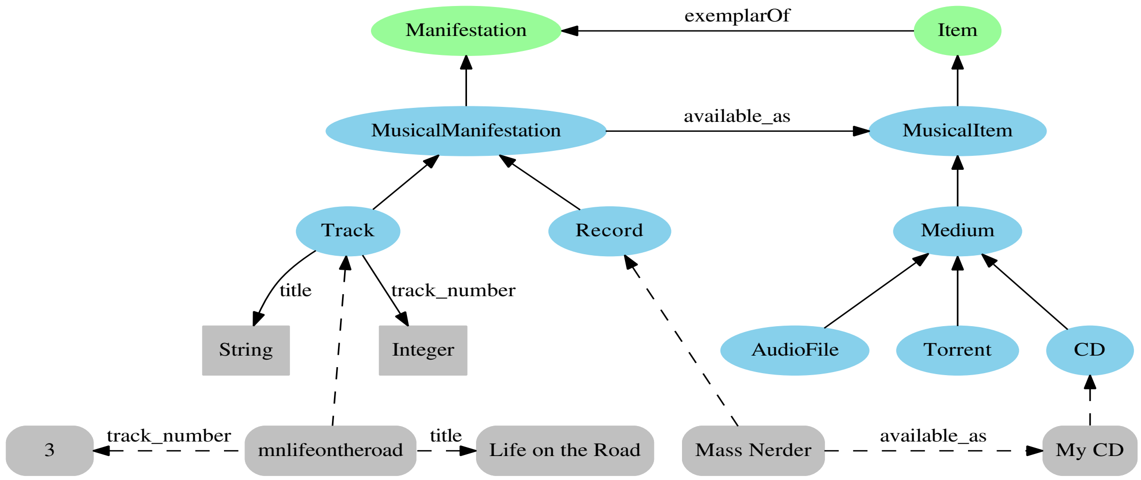



Figure 2.2: An ontology sample for a musical item based on the Music Ontology.

incoming arrow from its domain an outgoing arrow to its range. The instantiations of concepts and relations are depicted as dotted arrow. The subsumption relations are depicted as upwards arrows. The example contains eight concept: `Manifestation, MusicalManifestation, Track, Record, Item, MusicalItem, Medium, CD`; two relations: `examplarOf, available as`; and two attributes: `track_number and title`.

- $\mathcal{C}$={`Manifestation, MusicalManifestation, Track, Record, Release, Item, MusicalItem, Medium, CD`}
- $\mathcal{R}$:={`available as, examplarOf`}
- $\mathcal{A}$:={`track_number, title`}

According to the **direct superconcept relation** we have, from left to right: `Track` $\subset$ `Musical Manifestation, Record` $\subset$ `Musical Manifestation, MusicalManifestation` $\subset$ `Manifestation, CD` $\subset$ `Medium, Medium` $\subset$ `MusicalItem`, and `MusicalItem` $\subset$ `Item`. For the relations and attributes in the example ontology we have the following signatures:

- $\sigma_R(\texttt{available\_as})$=$\{\texttt{MusicalManifestation, MusicalItem}\}$

- $\sigma_R(\texttt{examplarOf}) = \{\texttt{Item, Manifestation}\}$

- $\sigma_A(\texttt{title}) = \{\textit{Life on Road}, \textit{String}\}$

- $\sigma_A(\texttt{track\_number}) = \{3, \textit{Integer}\}$

Further, the axiom that every CD needs to have at least one record is defined. Axioms are not depicted in the graph, but represented in the paragraphs.

$$\forall x\, \texttt{CD}(x)\, \exists\, y\, \texttt{available\_as}(y,x) \wedge \texttt{Record}(y) \tag{2.4}$$

In our example, we have three instances $\mathcal{I} := \{\textit{My CD}, \textit{Mass Nerder}, \textit{mnlifeontheroad}\}$. Further, we have the following instantiation relations:

- $\mathcal{I}_C(\texttt{CD}) := \{\text{My CD}\}$

- $\mathcal{I}_C(\texttt{Record}) := \{\text{Mass Nerder}\}$

- $\mathcal{I}_C(\texttt{Track}) := \{\text{mnlifeontheroad}\}$

- $\mathcal{I}_R(\texttt{available\_as}) := \{\text{Mass Nerder, My CD}\}$

- $\mathcal{I}_A(\texttt{track\_number}) := \{\text{mnlifeontheroad}, 3\}$

- $\mathcal{I}_A(\texttt{title}) := \{\text{mnlifeontheroad, Life on the Road}\}$

In this section the ontology and knowledge base structure have been introduced. Ontologies are provided a well-defined semantics by improving these definitions with an actual ontology language, such as OWL in Section 3.1. Axioms allows us to formalise a wide range of associations between objects. Particularly the expressive semantics completely set ontologies apart from other schema structures such as XML trees[1], database schemas, or UML[2]. The semantics of ontologies allow inferring supplemental knowledge.

## 2.3 Classification of Ontologies

Ontologies can be classified into four categories: Top-level Ontologies, Core Ontologies, Domain Ontologies and Application Ontologies.

### 2.3.1 Top-level Ontologies

Top-level Ontologies provide broad semantic interoperability among diverse knowledge domains by describing general concepts, such as time, space, events and processes, that are common across all knowledge domains. The main purpose of ontologies is to support communication among artificial agents and human being. As one of the main principles of the communication is having a common syntax and terminology among artificial agents, the top-level ontologies plays an important role for establishing consensus in independent ontology population that exists on the Semantic Web. The concepts of top-level ontologies must be independent from any particular problem or domain, it should be as general, reusable, and widely applicable as possible. In this sense, top-level ontologies stay closest to philosophical ontologies. As an example, the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)[3], Upper Mapping and Binding Exchange Layer (UMBEL)[4] and Standart Upper Ontology (SUO)[5], and Cyc[6] can be given as comprehensive foundational knowledge representation.

[1] http://www.w3.org/XML/
[2] http://www.uml.org/
[3] DOLCE:http://www.loa-cnr.it/DOLCE.html
[4] UMBEL:http://umbel.org/
[5] SUO:http://suo.ieee.org/
[6] Cyc:http://www.opencyc.org/

### 2.3.2 Core Ontologies

Core Ontologies maybe seen as a more general domain ontologies, involving concepts and relationships that allow to subsume any specific domain ontologies under fundamental concepts. Typically core ontologies define concepts, such as state, event, process, action, component, etc. The Music Ontology, the Core Ontology for Multimedia (COMM) [Arndt *et al.*, 2007] and the CIDOC Conceptual Reference Model (CIDOC CRM) [Crofts *et al.*, 2010] are examples of core ontologies.

### 2.3.3 Domain Ontologies

Domain Ontologies are the core of any semantic modelling effort. The main purpose of the domain ontology is defining concepts and their relationships by providing a clear description for a domain of interest. Thus, the particular meaning of terms applied to the corresponding domain are provided by a domain ontology. For instance, we may think of an ontology of musical instruments or audio features as examples in the music domain. More concrete examples include the OWL-Time[7] that describes the temporal concepts (e.g. instants, intervals, duration and date-time) [Hobbs & Pan] or device ontology describing devices and their services (e.g. printing service, scanning service) [Bandara *et al.*, 2004].

### 2.3.4 Application Ontologies

Top-level, core ontologies and domain ontologies are emerging ontology types having a potential to provide a conceptual foundation for linking diverse application ontologies on the Semantic Web. Their scope is, however, so large and thus inadequate to cover the specific use case of an application. Thereby, an application ontology describes concepts depending on a particular domain or task by combining, interlinking and enriching knowledge from various sources. Application ontologies can be developed either by extracting the entire ontology as a subset of one or augmenting the corresponding application terminology with ontological elements from diverse domain ontologies.

[7] `OWL-TimeOntology:http://http://www.w3.org/TR/owl-time/`

Contrary to the domain ontologies, which are broad and shallow, application ontologies are narrow and deep, and most of the application ontologies are developed by domain experts for use in specific types of applications.

## 2.4 Summary

In this chapter an introduction and persuasion why computer science needs the "concept" of ontology has been presented. The roots of conceptualisation and ontology in philosophy as well as its core paradigm in line with the meaning triangle have been introduced. Subsequently, it has been shown how ontologies may support communication and explicit representation of knowledge. Therefore, the notion that underlies the meaning triangle has been combined with a "semiotic view" on ontologies resulting in an ontology and knowledge base structure.

A formal mathematical definitions of these structures describing their core elements and their interaction have been given in Section 2.2 with an engineering standpoint. Finally, we have also reviewed the classification of ontologies in order to have a detailed information with the purposes of distinctive kind of ontologies in Section 2.3.

# Chapter 3

# The Semantic Web

In this chapter we will consider the concepts of the Semantic Web and Ontology in tandem. We will begin by presenting the significance of these notions for the distribution of knowledge in section 3.1. We will then highlight the progress that has been made in the semantic richness of ontologies through our review of ontology representation models, such as taxonomies, thesauruses, conceptual models, and logical theories, in section 3.2.

The Semantic Web is an initiative of the World Wide Web Consortium (W3C) which proposes standards underlying the technologies of the Web. Therefore, we will present W3C technologies that are relevant in our examination of issues in ontology design and generation in section 3.3. We will look at the Semantic Web standards used in this study; namely, RDF, RDFS, SKOS, OWL, and SPARQL.

Finally, we will outline ontology engineering on the Semantic Web by giving instances from various application domain studies in section 3.4.

## 3.1 Ontology Engineering on the Semantic Web

As more people obtain access to better and cheaper digital technologies, most companies and institutions have begun to reach a wider range of users and reduce cost using internet services and intelligent agent systems. Ontologies have been applied to a very wide range of fields (e.g. music, geology, agriculture, defence, robotics, and intelligence agent systems) to enhance communication among these web agents, reducing content heterogeneity. In this section we introduce the general problem of information sharing in the presence of heterogenous data. Therefore, we explain the Semantic Web and Ontologies as a means of dealing with the semantic heterogeneity and identify open problems that will be addressed in the remainder of this thesis.

### 3.1.1 The Semantic Web

Considering the fact that language was the earliest instrument to communicate and share human knowledge, ever since the development of language, there have been attempts to improve the distribution of knowledge. For example, printing, television and radio are significant inventions in the history of distribution of knowledge. Though these inventions have their own distinct ways of sharing information, the World Wide Web (www) brought all these features together, whereby one can read (e.g. books, articles, or newspapers), listen (e.g. music and audio books), watch (e.g. documentaries, movies, or videos), and share thoughts easily with other people using social media such as Twitter[1] and Facebook[2]. The enormous increase in the amount of published data and the demand for access to different types of information have, however, led to a knowledge management issue on the Web.

Traditional query engines are not capable of answering any complex queries, such as finding information about the records of a specific music group released after 1984 or detailed information about a song — e.g. "how was a song produced?" or "what effects were used to achieve that

[1] http://www.twitter.com
[2] http://www.facebook.com

particular instrumental timbre?". This seems to be beyond the capabilities of any web query engine. Web 2.0-based applications obtain metadata, exploiting the power of user communities by allowing them to share and annotate data, such as last.fm[3], youtube[4] and flickr[5]. The annotations, however, take the form of simple tags, such as "rock", "guitar", "fun", "80s". The meanings of tags are typically not well defined, and in some cases it maybe even difficult for human users to understand them. Though these approaches seem to be practically very convenient to accomplish small-or large-scale tasks, they don't provide any solution to the problem of how to locate and integrate information. The aim of the Semantic Web is to solve these issues.

The Semantic Web is the new WWW infrastructure that will enable describing the formal semantics and machine processing of current web content. The scope of the Semantic Web has increased in gradual prevalence as a research area since the original idea was introduced by Sir Tim Berners-Lee. Contrary to the structure of the current Web, Semantic Web technologies include tools for interlinking heterogeneous data sources including content and metadata, providing the Web with Linked Data. The Semantic Web is, therefore, considered as an integrator across different content, information applications and systems. In the future, consequently, the Semantic Web will be an excellent domain in which to develop Artificial Intelligence (AI) applications. The machine processibility that may be achieved on the Semantic Web relies to a great extent on the availability and proliferation of ontologies.

### 3.1.2 Development of Ontologies

The development of Ontologies is a formalisation process to represent acquired knowledge in an organised and structured form that both computers and humans can understand. Most of the current ontology development methods still require tremendous effort and subjective judgments from the

[3] www.last.fm
[4] www.youtube.com
[5] www.flickr.com

ontology developers to acquire and maintain the ontology. The ability to design and maintain ontologies requires expertise in both the domain of the application and the ontology language used for modelling. However, with their growing utilisation, not only has the number of available ontologies increased considerably, but they are also becoming larger and more complex to manage. A formal ontologist has to deal with two different tasks: *i*) metaphysics and *ii*) system engineering. The former deals with highly abstract domain information such as Top-level ontologies. The latter task is largely an empirical endeavour whereby information is gathered and tested over a specific domain of interest, such as core ontologies, domain ontologies and application ontologies. Here are the fundamental steps in the ontology construction progress [Morris, 2007]:

- Enumerate all relevant terms that are expected to appear in the ontology. Identify as many 'things' and 'concepts' in the domain as possible.
- For each concept, enumerate its attributes, properties and specify any restricted values. The hierarchical relationship of concepts is a very important aspect of ontology design. For instance, if a concept named "**A**" is a subclass of another concept called "**B**" it means that every property statement that holds for an instance of **B** must also apply for instances of **A**.
- For each pair of things or concepts, one needs to decide if there is some relationship between them, and look for hierarchy, composition, co-operation, and dependence. These can be defined by using basic set operations on boolean combinations of classes such as Union ($\cup$), Intersection ($\cap$), and Complement ($\complement$). These steps can be enriched using OWL primitives such as cardinality, required values, and relational characteristics.

Ontology engineering has a rich literature. More detailed account on the process of ontology development can be found in [Staab & Studer, 2009] for an overview. The rest of the section outlines the semantic richness of ontology representation models, and provides the basis for our examination of the knowledge representation issues in musical instrument ontology design in Chapter

4.

## 3.2 Semantic Richness of Ontology Representation Models

Computers need to use ontologies to facilitate automated reasoning, so we have to adapt the communication theory defining a machine readable syntax, to make it interpretable and accessible. To establish required tasks, thus, every computer/machine can use and interpret a common language with the same terminology to establish required tasks.

Semantic Web ontologies encompass the open world assumption. This assumption leads to a partial and modular structure in which an ontology can be merged with another ontology, or adapted to continuously changing knowledge. Figure 3.1 depicts the semantic richness of various kinds of ontology representation models, starting from left bottom where the semantics are simple and weak, and continuing towards to the top right where the semantics are strong and capable of representing more complex meaning.

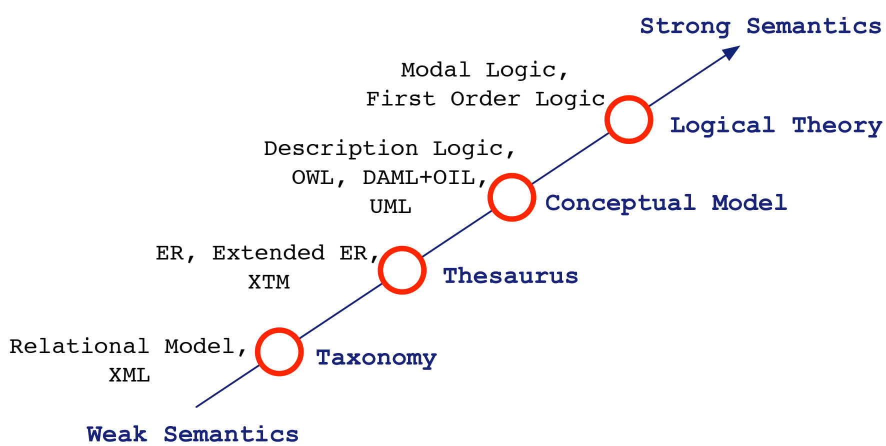



Figure 3.1: The ontology spectrum, adapted from [Daconta *et al.*, 2003]

### 3.2.1 Taxonomies

To date, taxonomies that allow us to organise data into a hierarchical structure are very efficiently and widely used in classification problems, such as search engines and e-library systems. Since taxonomic design is based on hierarchical structure and supports syntactic interoperability, this model can search and find the necessary information much faster and compared to other classification models. This method has also been used a lot by ethno-musicologists in order to classify musical instruments. In a taxonomy, however, the relationship between a parent node and a child node is ill-defined, so taxonomies involve semantically weak structures for expressing knowledge. Taxonomic relations have usually been defined by using "`sub-class`" axiom without defining the detailed relationships among concepts. We will provide more details along with examples of instrument taxonomy classification studies from ethno-musicology in section 4.1.

### 3.2.2 Thesauruses

Compared to taxonomies, a thesaurus model provides a different kind of relationship between concepts such as `synonyms`, `homonyms`, the `narrower`, the `broader` and `associated` relations. These relationships improve the semantic richness of this model to a certain extent: for instance, `synonyms` describe terms with similar meanings, whereas `homonyms` describe terms which are written in the same way but have different meanings. The `broader` and `narrower` relations describe the parent-of and child-of relationships among entities, respectively. Finally, the `associated` relation defines the related entities in an ambiguous way. The Entity Relationship (ER) model, Extended Entity Relationship (EER) model and XML Topic Map (XTM) can be categorised as thesaurus models, since they only support structural interoperability.

### 3.2.3 Conceptual Models

A conceptual model allows the effective characterisation of knowledge in a particular domain, thereby providing well-defined meta-model through concept subsumptions, properties and attributes of domain classes. For instance, OWL or UML diagrams may be used in such contexts for domain modelling [Domingue *et al.*, 2011].

### 3.2.4 Logic Theories

When we move towards to the top right, there are formal approaches in which the level of explicit meaning and the degree of formality grows along with increasing support for automated reasoning. Logical languages are semantically interpretable and allow us to specify formalised logical theories. The open world assumption is crucial for logic theories to infer a new knowledge from another integrated or heterogeneous ontology by using the common components. Some of the well-known logic languages are $(i)$ Description Logics (DL) which are strict subsets of first-order logic, and $(ii)$ traditional logic programming such as F-Logic [Staab & Studer, 2009].

## 3.3 Semantic Web Technologies for Ontology Representation

The Semantic Web is an initiative of the World-Wide Web Consortium (W3C) which sets underlying standards for the technologies of the World-Wide Web [Matthews, 2005]. The W3C is a forum that provides information infrastructure between people and organisations. It was set up to prevent Web standards from being dominated by commercial interests. The W3C investigates how to maintain interoperability and universality of the Web using open standards. Some of the well-known standards and web-based ontology languages are presented in this section. The technologies used throughout this thesis are listed below:

- RDF: Resource Description Framework,
- RDF schema,
- SKOS: Simple Knowledge Organization System,
- OWL: Ontology Web Language,

- SPARQL: Simple Protocol and RDF Query Language.

### 3.3.1 Resource Description Framework (RDF)

RDF[6] is a standard for describing content available on the internet. The idea of RDF originates from the Platform for Internet Content Selection and has been developed by two consecutive working groups within the World-Wide Web Consortium. The fundamental data structure underlying the RDF model consists of statements formed by *subject*, *predicate* and *object* terms. These statements are also called *triples*. As depicted in Fig 3.2, the representation of a triple is a single edge, labelled with predicate, connecting two nodes, which are labelled with the subject and object. This describes a binary relationship between the subject and object via the predicate.

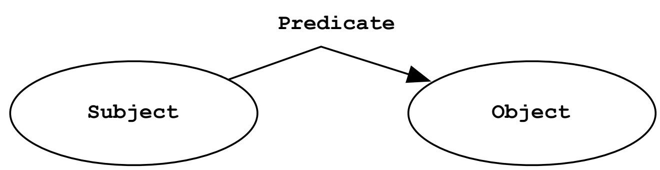



Figure 3.2: The fundamental data structure underlying the RDF model: *subject,* predicate, and *object.*

A set of triples is called an RDF graph. In order to facilitate the sharing and exchanging of graphs on the Web, the RDF specification includes Turtle and Notation 3 (N3) serialisations. An example representing the details of a track on a band's music album is given in Listing 3.1. The band concept is captured by `mo:MusicGroup` where the namespace prefix **mo** is associated with the namespace name `http://purl.org/ontology/mo/`. This Turtle resource provides information about the home page, images, and name of the band.

[6] http://www.w3.org/RDF/

```
@prefix io: <http://example.org/io/taxonomyN#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix mo: <http://purl.org/ontology/owl/> .
@prefix ex: <http://example.com/> .

ex:mnlifeontheroad
   a mo:Track ;
   mo:title "Life on the Road" ;
   mo:track_number "3" ;
   foaf:maker [
      a mo:MusicArtist;
      foaf:name "All";
      owl:sameAs
   <http://dbtune.org/musicbrainz/page/artist/e92547b5-a134-4149-
      af9a-c944af9e476f>
   ] .
```

Listing 3.1: A Turtle sample based on Music Ontology [Raimond, 2008]

### 3.3.2 RDF Schema

The RDF vocabulary definition language is called RDF Schema[7]. RDFS is an extension of RDF in which a small vocabulary is been included that defines (for instance) `rdf:Property` or `rdf:type`. The core difference is that the terms included in RDF are not sufficient for describing an ontology. RDFS, however, is a simple ontology language. For example, an RDFS sample given in Listing 3.2 illustrates two classes – namely, `Record` and `Track` – denoting a property link between them, where "**a**" is a property representing `is-a` relationship. The domain and range properties are specified using `rdfs:domain` and `rdfs:range` to represent the property with `mo:track` linking the `Record` class and `Track` classes.

Although RDFS extends RDF by including basic features needed to define ontologies, it lacks some important features to describe an ontology in more detail. These include local scope properties to declare range restriction that apply to only some classes; boolean combination of classes

[7] http://www.w3.org/TR/rdf-schema/

```
mo:Record
      a rdfs:Class ;
      rdfs:label "Record" .
mo:Track
      a rdfs:Class ;
      rdfs:label "Track" .
mo:track
      a rdf:Property ;
      rdfs:label "track";
      rdfs:domain mo:Record;
      rdfs:range mo:Track .
```

Listing 3.2: RDF Schema based on Music Ontology [Raimond, 2008]

(e.g. union, intersection, and complement); cardinality restrictions (e.g. how many distinct values a property may or must take) — and special characteristics of properties (e.g. transitive, unique and inverse).

### 3.3.3 Simple Knowledge Organisation System (SKOS):

SKOS[8] is a semi-formal model for expressing controlled vocabularies (classification schemes, thesauruses, taxonomies) in RDF. It defines `skos:Concept`, whose individuals may be associated with one or more lexical labels (`skos:prefLabel, skos:altLabel`) and placed within a hierarchy using `skos:broader`, `skos:narrower`, or `skos:related` properties, exhibiting a thesaurus model [Allemang & Hendler, 2008].

SKOS is defined as an OWL Full ontology: that is, it uses a sub-vocabulary of OWL Full to define a vocabulary for simple resource descriptions based on controlled structured vocabularies. It provides a fairly simple set of model constructs that allow the creation of extensible, distributed information networks. Therefore, information represented in a different language can be easily transformed into SKOS. Publishing data in SKOS also enables the concepts defined to be referenced on a global scale.

[8] http://www.w3.org/TR/skos-reference

SKOS also describes associate relations, for instance `skos:related`, in a semi-formal way, thereby providing limited support for other kinds of relationships and more explicit definitions, even though it is suitable for hierarchical classification schemes. Consequently, it is difficult to retrieve information without additional knowledge. This point will be discussed further in Chapter 4.

### 3.3.4 Ontology Web Language (OWL)

In the late 90s, the necessity for a much more expressive ontology language was commonly accepted within the Semantic Web research community and led to several proposals for new Web ontology languages, such as Simple HTML Ontological Extensions (SHOE), the Ontology Inference Layer (OIL), and DAML + OIL. Certain requirements – for instance, interoperability between disparate data repositories – led to the realisation that a standard ontology language is crucial for the development of the Semantic Web. The W3C, therefore, established a standardisation research group to develop an ontology language in 2001. By taking into account earlier proposals (e.g. OIL, DAML + OIL and RDF), the outcome of this work was the Ontology Web Language (OWL) standard[9]. The foundation of OWL is based on open-world semantics, where missing information is treated as unknown rather than as false. Additionally, axioms may represent inference rules along with a mixed set of Terminology Box (TBox) and Assertion Box (ABox) axioms. The former plays the role of a conceptual schema to describe the constraints on the structure of a domain and the latter asserts facts about concrete situations. In comparison with earlier ontology languages, OWL allows one to explicitly define and instantiate a web ontology along with a richer vocabulary description language describing properties and classes.

*OWL-Full :* allows free mixing of OWL with RDF Schema and, like RDF Schema, does not enforce a strict separation of classes, properties, individuals and data values.

[9] http://www.w3.org/TR/owl-features/

*OWL-DL :* is based on Description Logics. It is a sublanguage of OWL Full that restricts how the constructors from OWL and RDF may be used. The advantage of this is that it permits efficient reasoning support. The disadvantage is that we lose full compatibility with RDF [Antoniou & can Harmelen, 2008]. The data type properties such as `owl:inverseOf`, `owl:FunctionalProperty`, `owl:InverseFunctionalProperty`, and `owl:SymmetricProperty` cannot be specified.

*OWL-Lite :* excludes enumerated classes, disjointness statements, and arbitrary cardinality. The advantage of this as a language is that is easier to understand (for users) and easier to implement (for tool builders). The disadvantage is the restricted expressivity. These constraints are that the constructors `owl:oneOf`, `owl:disjointWith`, `owl:unionOf`, `owl:complementOf` and `owl:hasValue` are not allowed. Cardinality statements (both minimal, maximal and exact cardinality) can only be made on the values 0 or 1, and no longer on arbitrary non-negative integers. `owl:equivalentClass` statements can no longer be made between anonymous classes, but only between class identifiers.

### 3.3.5 Simple Protocol and RDF Query Language (SPARQL):

SPARQL[10] defines a standard access protocol for RDF that provides Semantic Web developers with a powerful tool to extract information from large data sets. A query consists of several graph patterns, which can be combined recursively to form complex queries. It may be used for any data source that can be mapped to RDF.

The syntax of SPARQL is based on the same notation for universal quantification that is used in Turtle and N3. The keywords SELECT identify the variables to appear in the query results and WHERE indicates a question pattern. In listing 3.3 we construct a simple query to retrieve a list of tracks of a music artist, called “ALL”.

[10] `http://www.w3.org/TR/rdf-sparql-protocol`

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?group ?track ?title
WHERE {
      ?track foaf:maker mo:MusicArtist;
                            foaf:name "ALL" .
}
```

Listing 3.3: A SPARQL query for retrieving a list tracks for a music group.

### 3.3.6 Linked Data

Linked Data is a recent movement that focuses on creating a Web of Data in order to facilitate the process of publishing structured data in an open format that shares a common conceptual framework, that is, RDF. The principles of Linked Data were first outlined by Berners-Lee in [Berners-Lee], and extended by technical documents such as [Bizer *et al.*; Sauermann & Cyganiak] that provide guidance on Linked Data. Based on these principles, Linked Data URIs must be dereferencable, such as HTML for humans and RDF for machines, thus the Web will be used by humans as well as by machines. This is achieved employing an HTTP mechanism called content negotiation. The basic idea of the content negotiation is to enable servers to identify clients requested resource format, such as HTML or RDF, through examination of HTML headers, and respond in appropriate format. The resource content should include links to additional HTTP URIs that also dereference the additional information. Hence, the Web of Data provides a meaningful navigation path for users as well as machines, and allows them to access different data resources, that are connected with RDF links.

The Linking Open Data (LOD) project aims to publish open data sets on the semantic web following linked data recommendations and appropriate web standards. Figure 3.3 depicts the Linked Data cloud where each note represents a distinct data set published as Linked Data.

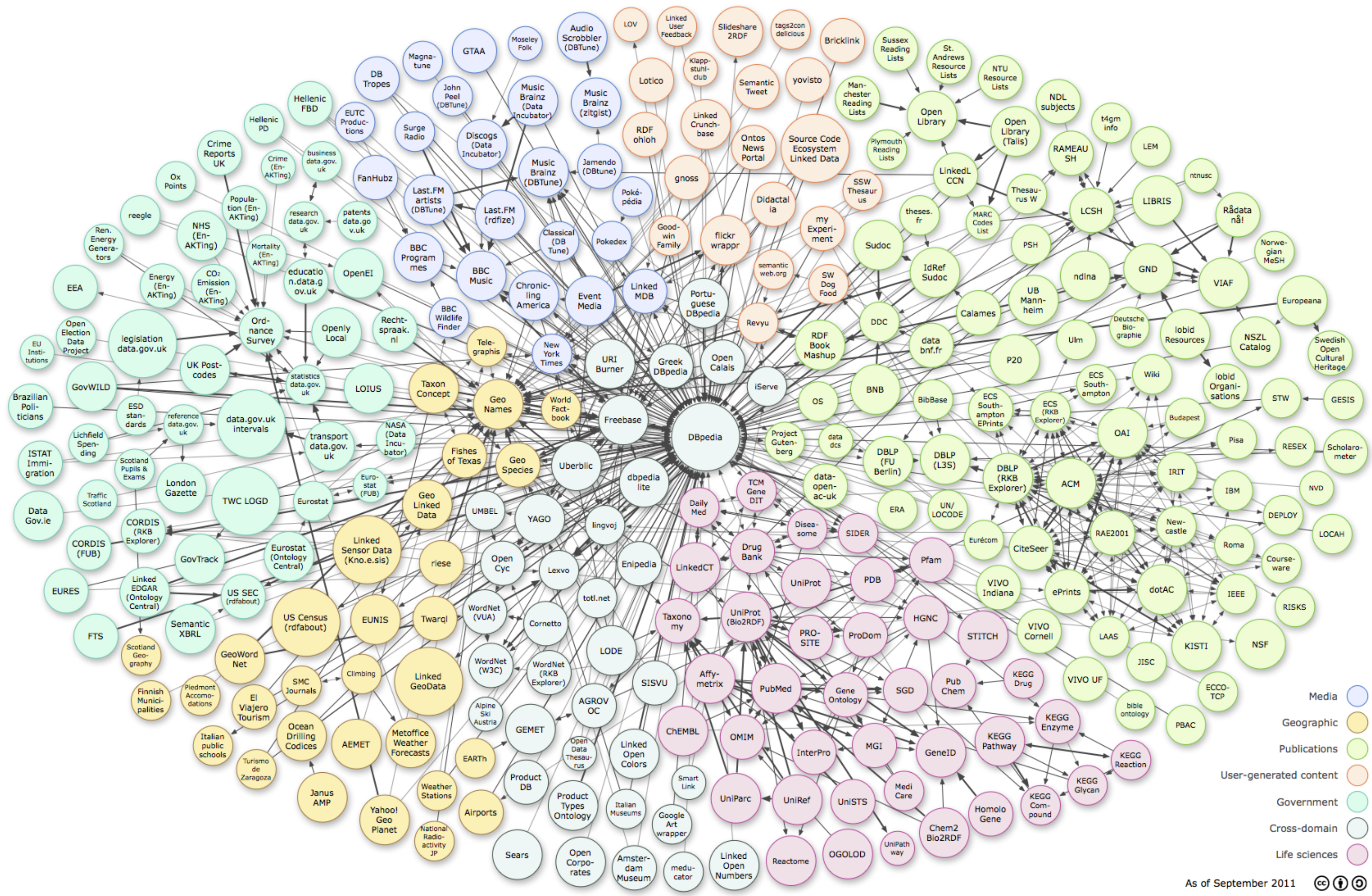



Figure 3.3: Linked Open Data cloud as of September 2011 where the pale violet circles indicate data sets related to music. Diagram by Richard Cyganiak, and available at http://richard.cyganiak.de/2007/10/lod/

## 3.4 Music Information Retrieval on the Semantic Web

The World Wide Web is an excellent resource that enables users to publishing and accessing documents within a global information space. With the increasing quantity of music content on the Web, however, there is a need to find ways of improving music retrieval effectiveness. While search engines index the documents and analyse the structure of links between them to infer potential relevance to users' search queries, navigating music content is a significantly different information retrieval task with some unique challenges. The most apparent distinction being that musical data comprised of not only textual data but also digitally encoded audio signals and symbolic musical representations. This led to development of Music Information Retrieval (MIR) as a response to user needs, such as managing, searching or accessing musical audio content on the Web.

Semantic descriptors extracted from audio recordings, such as chords, rhythm and instrumentation, can be applied to a variety of applications including audio collection management or music recommendation. These descriptors can also be a valuable resource, when interlinked with cultural or editorial metadata, for instance concerts, tour dates, or artist relations. Linked Data provides such environment that involves structured data connections from diverse domains and enables navigation between data sources to discover additional data.

In this section we will first overview the present music-related linked data, and describe our work on collecting, organising and publishing a large set of music similarity features coupled with valuable editorial meta-data. Secondly, we will review the knowledge based approaches for MIR systems.

### 3.4.1 Music-related Linked Data

A wide range of music-related data sources have been interlinked within the Linked Open Data initiative. The *DBTune*[11] is one of the important project that hosts many music-related linked data sets,

[11] http://dbtune.org/

such as the *Jamendo*[12], the *Magnatune*[13], the *John Peel sessions*[14], the *AudioScrobbler wrapper*[15], a MusicBrainz translation, and the *BBC playcounts*[16]. For instance, *Jamendo* is linked to Geonames, therefore it provides geolocation-based mash-up for musical data. Enabling music artists to publish their works under a creative commons license, the *Jamendo* and *Magnatune* datasets allow users to retrieve information with regards to the music artists, their geographic locations and works. While *Jamendo* is more of an open platform, *Magnatune* is more like a traditional record label that hand-picks artists and content for publication. BBC is one of the big organisations that contributes to the open linked data. For instance, BBC released some metadata about the *John Peel sessions* that describes metadata related to the various recordings associated with the long running John Peel BBC 1 radio show in RDF. An additional music-related datasets is the *BBC playcount* dataset, that can be used to retrieve information about music artists, which are played on BBC programmes, as RDF with links to the *DBTune* and *Musicbrainz* dataset.

With a straightforward hashing method for generating unique identifiers for music artists, albums, and tracks, the *MusicBrainz*[17] project has constructed one of the most comprehensive music metadata repositories on the Semantic Web. Recently, MusicBrainz repository's content has been published as Linked Data by converting music metadata into RDF and providing it as TripleStore/SPARQL [Dixon *et al.*, 2011]. This allows users to easily search through available data and integrate information from other repositories. *DBpedia*[18] is another online database containing information that possesses structured data extracted from Wikipedia. These datasets are linked to each other. As a contribution to the Linked Open Data project, we published a data set involving music similarity features produced by the SoundBite playlist generator tool [Tidhar *et al.*, 2009].

[12] http://www.jamendo.com/
[13] http://magnatune.com/
[14] http://dbtune.org/bbc/peel/
[15] http://dbtune.org/last-fm/
[16] http://dbtune.org/bbc/playcount/
[17] http://www.musicbrainz.org/
[18] http://dbpedia.org/

The details of the process of collecting, cleaning and publishing of music similarity features are explained below.

*The SoundBite dataset*

In this section, we describe the process of collecting, cleaning and publishing music similarity features from a large user base coupled with valuable editorial metadata. Metadata are verified against MusicBrainz, a large public database of editorial information on the Web, and published together with the matching similarity features on the Semantic Web [Tidhar *et al.*, 2009]. The heart of the data collection system is SoundBite [Levy & Sandler, 2006, 2007], a tool for similarity-based automatic playlist generation. SoundBite is available as an iTunes plugin. Once installed, it extracts features from the users' entire audio collection and stores them for future similarity calculations. It can then generate playlists consisting of the *n* most similar tracks to any given seed track specified by the user. The similarity data currently consist of 40 values per track based on the distribution of Mel-Fequency Cepstral Coefficients (MFCC) as described in [Levy & Sandler, 2006]. The extracted features are also reported to a central server, where they become part of the so called Isophone database. This database is used for aggregating information from SoundBite clients, consisting of editorial metadata and similarity features for each audio track. The entire system may therefore be regarded as a distributed framework for similarity feature extraction. The SoundBite dataset consists of MFCC features and MusicBrainz identifiers for a cleaned-up subset of the data reported back to the central server by the different instances of the SoundBite client application. Currently, the database includes metadata for 152,410 tracks produced by 6,938 unique artists.

Prior to publishing, the data needed to undergo a clean-up process. In the first stage of the clean-up process: title, artist, and album are matched against the MusicBrainz database. The durations of tracks are used for resolving ambiguities, as well as for sanity check (a large difference between the reported duration value and the duration retrieved from MusicBrainz may indicate that the other fields are erroneously or maliciously wrong). Each matching track is been assigned an ID provided by the MusicBrainz database, which serves as unique identifier. We found that about 28% of the entries in our database had exact matches (artist, title, album, and approximate duration) in the MusicBrainz database. The remaining 72% are stored for possible future use, but do not currently qualify for publishing. The relatively small proportion of tracks that do qualify can be regarded as an indication of the poor reliability of textual metadata in end users' audio collection. As indicated in [Sigurdsson *et al.*, 2006], MFCC features are more robust at higher bit rates. Therefore, in the second stage the data is further filtered according to maximum bit rate and best quality audio file type (e.g. keeping AACs as opposed to MP3s), in order to preserve the highest quality features for each track. Since these parameters are included in the metadata reported to the server, this doesn't require access to the audio files themselves. Once cleaned-up and filtered as described above, the MFCC features and the obtained MusicBrainz IDs are exported from the database as RDFs using the D2R Mapping [Bizer & Cyganiak, 2006], with the appropriate linking to the Audio Features[19] and SoundBite ontologies (see figure 3.4). They are then made available via a Query Language for RDF (SPARQL) end-point on our server.

### 3.4.2 A Knowledge-based approach for MIR applications

Knowledge management is an important issue in the field of music informatics. Researchers usually use different configuration parameters and data formats. While music information retrieval community has developed a significant amount of tools and frameworks, it is very difficult to collaborate and share data meaningfully without having to interpret divergent data formats. This becomes even a greater problem with the utilisation of data which could be produced, for instance by audio analysis algorithms providing higher-level representations than the audio signal itself.

Many organisations defined their own standards to encode information for different aspects of the audio domain. As a result, many incompatible standards and methods were produced in

[19] http://purl.org/ontology/af/

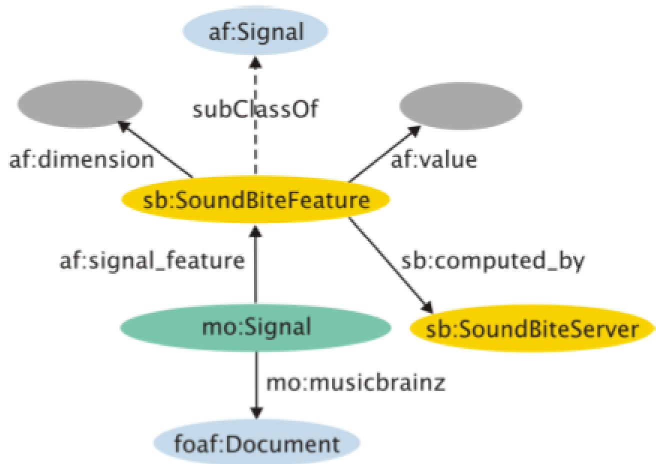



Figure 3.4: Accessing the SPARQL endpoint using the SoundBite ontology

different schema and syntax without common grounds and principles. There has been a great deal of effort to standardise description formats, for example the MPEG-7 International Standard for Multimedia Content description[20], as well as in providing vocabularies for content, tools, and methods.

The Semantic Web technologies and Linked Data concepts include widely accepted standards and formats as a knowledge-base approach to inefficiencies in data representation exists in MIR. A substantial quantity of music-related linked data is already published to date. Using a compatible framework for data will make it easy to augment results with links to these existing resources. Thus, researchers can reliably associate analytical results with other information about the source material, and also use existing implementations for free library code for data structure, storage, syntax, serialisation, and querying.

Several ontologies have been developed for describing musical metadata for the Semantic Web within the OMRAS2 project. For instance, the *Music Ontology*[21] provides terms for describing

[20] http://mpeg.chiariglione.org/standards/mpeg-7/
[21] http://musicontology.com/

musical metadata such as artist or performance. It relies on the *Event*[22] and *Timeline*[23] ontologies. The *Similarity*[24] ontology permits the expression of the similarity between things as a class with its own properties, such as to describe the quality of that similarity. It is flexible enough to be able to encompass concepts as diverse as artistic influence and timbral similarity. At a signal and feature description level, the *Audio Features*[25] ontology provides terms to represent features of audio signals, such as chromagrams or onset detection function . The *Chord*[26] ontology provides vocabulary for describing chords and chord sequences.

The utilisation of different programming languages, software libraries and Application Programming Interfaces (API) can cause massive issues during data or result exchange. Hence, the Vamp audio analysis plugin format, which is an audio processing API enhanced with Semantic Web ontologies, was proposed as an alternative solution to this issue along with an ontology, called the *Vamp Plugin Ontology*[27]. Moreover, RDF representation of the produced data allows it to be linked to further musical metadata such as title and artist information via further terms from the Music Ontology.

Another important use case includes instrument identification in a knowledge based environment. While several instrument classificaiton systems have been proposed by musicologists and etno-musicologists, there's no universally accepted system to date. More details will be provided regarding the musical instrument ontologies in Chapter 4.

[22] http://purl.org/NET/c4dm/event.owl
[23] http://purl.org/NET/c4dm/timeline.owl
[24] http://purl.org/ontology/similarity/
[25] http://purl.org/ontology/af/
[26] http://purl.org/ontology/chord/
[27] http://omras2.org/VampOntology/

## 3.5 Summary

In this chapter we have discussed ontology representation models and Semantic Web technologies and how they have been applied to music-related information and other research fields. Despite some of the difficulties mentioned in Section 3.4, the utility of Semantic Web technologies for modelling the complexity of the musical instrument domain remains untouched. In Chapter 4, we will discuss the knowledge representation issues in musical instrument ontology design and how the semantic web technologies we discussed in this chapter can be used to model a wide range of instrument characteristics.

# Chapter 4

# The Semantic Web and Musical Instrument Ontologies

This chapter presents our work on musical instruments ontology design, and investigates heterogeneity and limitations in existing instrument classification schemes. First, we will provide information in knowledge representation issues in music classification schemes based on the investigations of organologists and museologists in section 4.1. Subsequently, we will highlight the music related ontologies, i.e. Music Ontology and several musical instrument ontologies, proposed to date in section 4.2.

Numerous research have focused on representation of musical instrument information. The works we examined are based on the well known Hornbostel and Sachs classification scheme. In section 4.3, the knowledge representation issues in musical instrument ontology design discussed in two categories: taxonomic issues and heterogeneity issues.

We also developed representations of these instrument classification schemes using the Ontology Web Language (OWL), and compared terminological and conceptual heterogeneity using SPARQL queries in section 4.4.

## 4.1 Knowledge Representation Issues in Musical Instrument Ontology Design

Knowledge representation in the domain of musical instruments is a complex issue, involving a wide range of instrument characteristics, for instance, physical aspects of instruments such as different types of sound initiation, resonators, as well as the player-instrument relationship. Since the 19th century, numerous studies developed systems for representing information about musical instruments, for instance, (ethno)musicologists have been working on creating a common vocabulary, which represents all instruments with relevant characteristics in a systematic way. The classification of instruments has also been investigated by organologists and museologists [Kartomi, 2001]. Hornobostel and Sachs [von Hornbostel & Sachs, 1914] proposed a musical instrument classification scheme as an extension of Mahillon's scheme [Lysloff & Matson, 1985], originally designed to catalogue the worldwide collection of musical instruments housed in the Brussels Conservatory Instrumental museum.

The Hornobostel and Sachs classification scheme (H-S system) relies on a downward taxonomy by logical division. The method later was formed by Draĕger [1948]. Although many attempts have since been made by scholars to improve the Hornobostel and Sachs' Systematik, it is still predominant in museums around the world. Kartomi [2001] attributes the success of the classification system to the fact that it is essentially numerical rather than lexical, making it an international system (e.g. 211.11-922 refers to the timpani or kettledrum in the H-S system). Elschek [1969], was the first to propose an upward method of classification based on instrument attributes complementing downward classifications schemes such as the Systematik.

## 4.2 Core and Domain Ontologies related to Musical Instruments

In this section, our primary aim is to investigate the instrument classification schemes and semantic richness of instrument domain ontologies proposed to date, which may be used in conjunction with a core ontology in music domain, called the Music Ontology[1]. Therefore, we outline the music ontology and previously published Semantic Web ontologies of musical instruments.

### 4.2.1 The Music Ontology

The Music Ontology[Raimond, 2008] provides a unified framework for describing music-related information (i.e. editorial data including artists, albums and tracks) on the Web. It is built on several ontologies such as the Timeline Ontology, the Event Ontology[2], the Functional Requirements for Bibliographic Records (FRBR) Ontology[3], and the Friend Of A Friend (FOAF) Ontology[4]. It subsumes specific terms from these ontologies, useful to describe music related data. The Timeline and Event ontologies, can be used to localise events in space and time. The FRBR model links books and other intellectual works with their creators, publishers or subjects, and provides a model to describe the life cycle of these works. This is reused by the Music Ontology to describe the music production workflow from composition to delivery. Finally, FOAF defines people, groups and organisations. The Music Ontology does not cover every music related concept, rather, it provides extension points where a domain specific ontology, such as a musical instrument or a genre ontology may be integrated.

### 4.2.2 Musical Instrument Ontologies

Based on the Musicbrainz instrument tree, Herman[5] published a musical instrument taxonomy expressed in SKOS. This serves as an extension to the Music Ontology. While SKOS is well suited for hierarchical classification schemes, it provides limited support for other types of relationships; `skos:related` for example, may be used to describe associative relations, but only in a semi-formal way, without a more explicit definition. Moreover, the transitivity of broader and narrower

[1] http://musicontology.com/
[2] http://purl.org/NET/c4dm/event.owl/
[3] http://vocab.org/frbr/core/
[4] http://xmlns.com/foaf/spec/
[5] http://purl.org/ontology/mo/mit#

relations are not guaranteed in SKOS, therefore it is difficult to infer for instance the instrument family of a given instrument, without additional knowledge not expressed in the model. While this taxonomy is suitable for applications that require only a semantic label to represent instruments associated with audio items, it is insufficient if the heterogeneity of instrument relations has to be explicitly represented.

The Kanzaki Music Ontology[6] also contains a small instrument taxonomy. However, there are only 5 instrument families defined (e.g. string instruments, woodwind instruments, brass instruments, percussion, and keyboard instruments), with 26 corresponding instrument classes. Although these works provide instrument taxonomies that can be used on the Semantic Web, there remains a need for a semantically rich ontology, which represents the heterogeneity as well as different components and aspects of musical instruments on the Web.

Finally, a recently published XML-based taxonomy serves as an extension to Music XML[7]. This system departs form Hornobostel and Sachs, and proposes a classification scheme based on materials and performance mechanism, instead of the sound production mechanism. However, it remains at a hierarchical design. Furthermore, XML in itself is insufficient for rich knowledge representations, therefore it is hard to see how this model may be extended to account for the heterogeneity and the diverse set of properties of musical instruments, and enable logical reasoning or answering complex queries.

## 4.3 Issues in Musical Instrument Ontology Design

Conceptualising a domain is inherent in developing knowledge based systems. In the fields of ethno-musicology and Music Information Retrieval (MIR), most conceptualisations of the domain of musical instruments are based on the taxonomical H-S system, and very few studies departed from this system. Taxonomies allow us to organise data in a hierarchical structure very efficiently.

[6] http://www.kanzaki.com/ns/music
[7] http://www.recordare.com/musicxml/

However, taxonomies encode a strict relationship between a parent node and a child node by using *sub-class* or *part-of* axioms, without defining the detailed relationships among instrument objects, therefore they are semantically weak structures for expressing knowledge [Daconta *et al.*, 2003; Hepp, 2005; Hepp & de Bruijn, 2007]. Musical instruments however have a multi-relational model, thereby instruments can belong to more than one instrumental family or sub-family. In order to illustrate the heterogeneity and taxonomic design problems occurring in current knowledge representations of instruments, two different instrument classification systems were taken into account: *i*) one proposed by Doktorski[8] which will be denoted taxonomy 'A', and *ii*) one proposed by Montagu & Burton [Montagu & Burton, 1971] which will be denoted taxonomy 'B'. We implemented both of the taxonomies in OWL, and they can be found at corresponding URL[9].

### 4.3.1 Taxonomic Issues

To overcome the knowledge representation issues in musical instrument ontologies, having a multi-relational design and providing information about the relationships among concepts is a crucial necessity. For instance, Figure 4.1 illustrates an example from the ontology design of the chordophones/string instrument family based on taxonomy A.

As shown in Figure 4.1, the violin and cello are classified as bowed instruments, the guitar and banjo are classified as plucked instruments, and the piano is classified as a struck instrument. However, violinists can vary their playing technique depending on the expressive intentions: the strings can be excited by drawing the hair of the bow across them (arco), or by plucking them (pizzicato). For these reasons, the violin should be classified as either a bowed or plucked instrument. In Figures 4.1 and 4.2, the concepts that occurred multiple times in various instrument families, are shown using dashed lined shapes ( e.g. struck, plucked and rubbed). We can demonstrate similar examples in the family of percussion instruments. For instance, in Figure 4.2, the tambourine is

[8] http://free-reed.net/description/taxonomy
[9] http://isophonics.net/content/musical-instrument-taxonomies

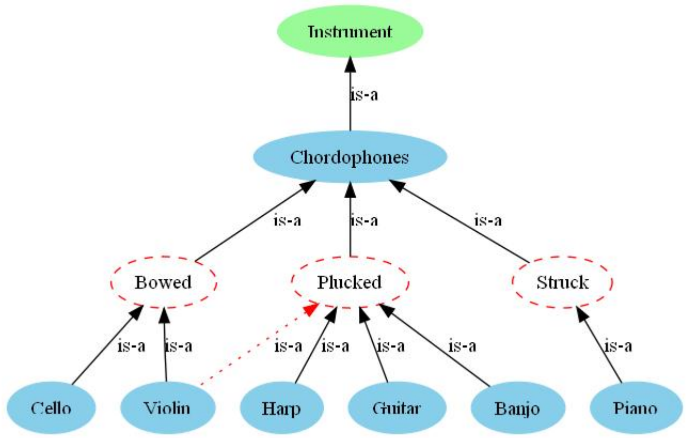



Figure 4.1: An example from musical instrument ontology design of chordophone/string instruments based on taxonomy A

classified as a membranophone, whereas if it is only shaken, it jingles, and therefore it could be classified as an idiophone as well. Many examples may be observer related to taxonomic classification problems, not only in the ethno-musicology, but also in other applications that rely on musical instrument knowledge representation or information management.

In taxonomy B, the use of classifications such as, *species*, *genus*, *family*, *sub-order*, *order*, based on the taxonomical system of Carl Linnaeus. However, this study only provides a terminological departure from the H-S system, since it is still based on the same taxonomy structure. A partial instrument ontology design of this classification scheme is depicted in Figure 4.3.

### 4.3.2 Heterogeneity Issues

The use of different words to refer to similar concepts, or different conceptualisations, induce terminological or conceptual heterogeneities among ontologies, that can be observed from the given graphical illustrations so far. For instance, in Figure 4.3, the idiophones and the membranophones are defined as a major instrument family according to taxonomy B, whereas both of these classes can be seen as sub-classes of the percussion instruments in taxonomy A (Figure 4.2).

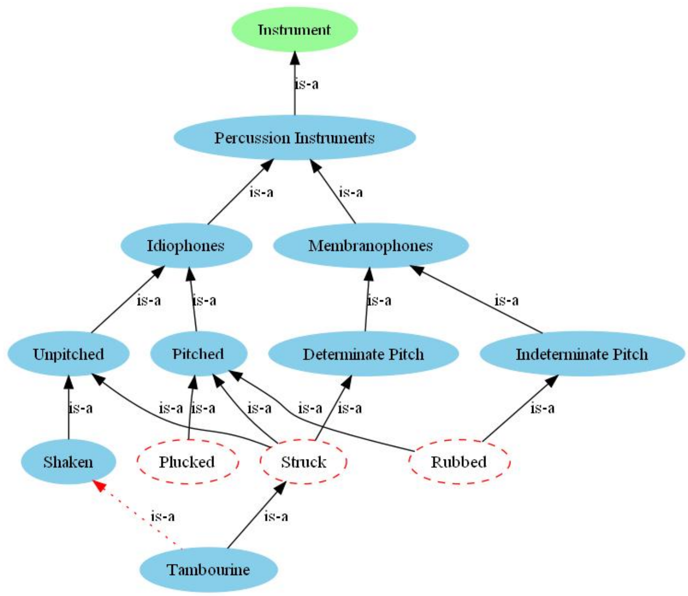



Figure 4.2: An example from musical instrument ontology design of percussion instruments based on taxonomy A

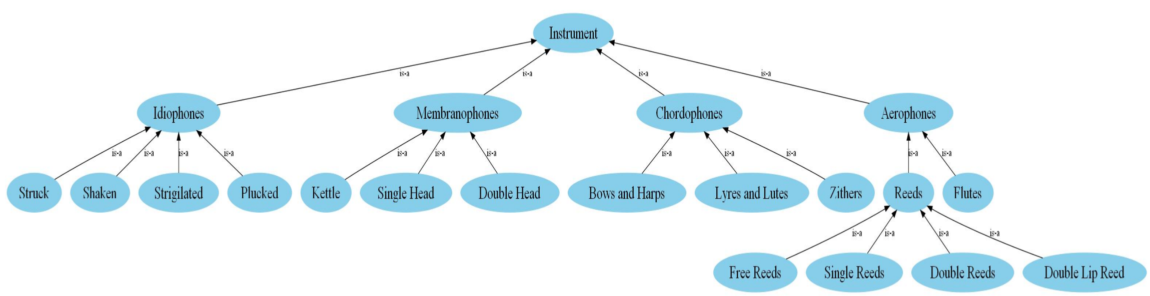



Figure 4.3: An example from musical instrument ontology design based on taxonomy B

The heterogeneity among these classes continues downward towards to the sub-class nodes: For instance, *idiophones* are divided into unpitched and pitched sub-categories, while *membranophones* are divided into determinate pitch and indeterminate pitch sub-categories (Figure 4.2). On the other hand, the *idiophones* have sub-classes such as struck, shaken, strigilated and plucked sub-classes, while *membranophones* have kettle, single head and double head sub-classes (Figure 4.3). Some concepts are present in the same taxonomic level without defining the relationship among concepts, and the concepts are classified according to sound initiation type (e.g. struck, plucked, or shaken), whereas others are classified according to the instrument construction type (e.g. single head, double head, harps, lyres and lutes). Therefore, the taxonomic classifications applied traditionally are not only heterogeneous in structure, but also provide an arbitrarily problematic solution to instrument classification, because of the inadequately defined knowledge representation.

## 4.4 Musical Instrument Taxonomies — Query driven evaluation

Both taxonomies described in the previous section were implemented in OWL and tested using SPARQL queries involving instruments present in both systems. In the following examples, we query the ontology structure, as well as RDF data corresponding to specific statements about instruments. Since in most knowledge-based environments, data and ontology can be represented in the same graph, these queries also demonstrate real-world use cases for instrument knowledge representation.

### 4.4.1 Query Example-1

The first example is based on the *tuba*, which is available in both taxonomies. The following paragraph provides a description of the tuba by Olson [1967]:

*The tuba is the lowest pitched Aerophone. Sound is produced by vibrating or buzzing the lips into a large cupped mouthpiece, which is coupled to a coiled tube about 18 feet in length with a slow rate of conical flare terminating in a large bell-shaped mouth. The tuba is usually equipped with three valves, each of which adds a different length of tubing. With piston valves it is possible to change the length of the air column.*

Identifying an instrument by its sound can be a difficult task, even for someone with a decent musical background. For this reason, visual cues can be just as important as hearing in instrument identification. For example, recognising the characteristic shape of an instrument is important, since it has a profound effect on the generated sound. Based on these considerations, we prepared the following four queries to retrieve the information underlined in the definition of the tuba above: What is the instrument family, the characteristic shape, the sound initiation type and the number of valves of the tuba?

```
PREFIX io: <http://example.org/io/taxonomyN#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?x WHERE { io:Tuba rdfs:subClassOf ?x }
```

Listing 4.1: Retrieving the immediate super class of the tuba.

In the first query the non-determined variable `?x` is assigned when the query engine finds the super class of the entity named *Tuba*. The query result for taxonomy *'A'* is `io:WithValves`, and for taxonomy *'B'* is `io:ValvesBugles`. This demonstrates terminological heterogeneity immediately on the first upper level. Note that name space prefixes such as `io:` and `rdfs:` are expanded to full URIs by the query engine. In the following queries, they will be omitted for brevity.

In order to retrieve the instrument family, we can either expand the query until we reach the corresponding node as shown in listing 4.2, or use a program to do so appropriately. This assumes knowledge about the depth and organisation of the taxonomy tree, that is, what information is

```
SELECT ?sc1 ... ?sc(N)
 WHERE { io:Tuba rdfs:subClassOf ?sc1 .
  OPTIONAL { ?sc1 rdfs:subClassOf ?sc2 } .
          .
          .
  OPTIONAL { ?sc(N-1) rdfs:subClassOf ?sc(N) } .
}
```

Listing 4.2: Hypothetical query for finding the instrument family of the tuba.

described on each level given a specific branch. Given this information, a reasoning engine could infer the instrument family relation, so that a direct query could be written. However, taxonomy based knowledge organisation systems do not contain this type of information, which is their main drawback in answering complex queries.

Intuitively, this query graph means that there exists an entity *Tuba* that is a subclass of *?sc1* having a relation with another entity whose name is non-determined. We may recursively go on until finding the entity *Aerophones*, the super-class of the last non-determined class. The query would succeed at the 4th super-class node for the taxonomy 'A' (e.g.*WithValves, BrassInstrument, PipeAerophones, Aerophones*), whereas the corresponding result would be obtained at the 10th node for the taxonomy 'B' (e.g.*ValvedBugles, SingleBell, Valves, EndBlown, Metal, Conical, DoubleLipReed, Reeds, Aerophones*).

The main problem with taxonomical representations is that it's difficult to answer certain queries without a more explicit knowledge representation. Taxonomic systems propagate meaning via the parent child relationship. We could infer that the tuba is an (*is-a*, or `rdf:type`) instrument with *Valves*, a *Brass instrument* and an *Aerophone*, according to taxonomy 'A'. The instrument family could be directly encoded using a semantically rich ontology. Although both taxonomies are based on the H-S system, it is easy to observe the diversity among different instrument taxonomies from these query results. The problem is not only the conceptual heterogeneity of the instruments themselves, but also the terminological heterogeneity among different knowledge representation schemes.

### 4.4.2 Query Example-2

The second query is *'What is the characteristic shape of the tuba?'*. To find this information, an upward recursive query, such as the one in Listing 4.2, or downward recursive query, which starts from the *Conical* concept, can be used to verify that the tuba is a conical instrument. However, both types of queries rely on external knowledge that can not be inferred from the pure taxonomical relationships directly. While taxonomy *'B'* at least contains the information about the characteristic shape of the tuba, being *Conical*, taxonomy *'A'* does not contain this information. In the third and fourth questions, we ask *'What is the sound initiation type of the tuba ?'* and *'How many valves has the tuba?'*. Unfortunately none of the implemented systems encode these relationships, therefore it is not possible to write queries to answer these questions that would produce any results.

In our second example shown in listing 4.3, we use the Music Ontology to represent the *Composition* and *Performance* events from the sentence below, assuming the composer also performed the piece:

*The American accordionist and composer Guy Klucevsek has written a piece for solo accordion, 'Eleven Large Lobsters Loose In The Lobby', which does not use the reeds of the accordion. The performer produces sounds by clicking the register switches, tapping the keys, and other percussive means. In this piece the accordion is used as an idiophone and not as a free-reed.*[10]

This example presents a case for knowledge discovery using instrument taxonomies. As shown in the example, lacking a more detailed ontological representation, we could not describe the accordion further to take into account the specific playing style. Since none of the taxonomies may be used to encode information about possible alternative sound initiation types, we may only obtain the instrument's default characteristics given a taxonomy, using recursive queries such as query

[10] http://www.ksanti.net/free-reed/description/taxonomy.html

```
@prefix io: <http://example.org/io/taxonomyN#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix mo: <http://purl.org/ontology/owl/> .
@prefix ex: <http://example.com/> .

ex:guy_klucevsek
  a mo:MusicArtist ;
  foaf:name "Guy Klucevsek" ;
  owl:sameAs <http://dbpedia.org/page/Guy_Klucevsek> ;

ex:guy_klucevseks_accordion
  a io:Accordion .

ex:ellltl
  a mo:Composition ;
  dc:title "Eleven Large Lobsters Loose in the Lobby"^^xsd:string ;
  mo:composer ex:guy_klucevsek ;
  mo:produced_work ex:w_ellltl;
  owl:sameAs
<http://dbtune.org/musicbrainz/page/track/8093f69e-194f-4cb1-8943-2
    d11fac6dcc6> .

ex:p_ellltl
  a mo:Performance ;
  rdfs:label "A performance of the composition."^^xsd:string ;
  mo:performer ex:guy_klucevsek ;
  mo:performance_of ex:w_ellltl ;
  mo:instrument ex:guy_klucevseks_accordion .
```

Listing 4.3: RDF Data based on Music Ontology and Music Instrument Taxonomy (taxonomy A).

4.2. Given this representation a reasoner can only infer that the Accordion is a *Hand blown, Free-reed, Aerophone* instrument. However, in this particular example, the instrument was played using different techniques, such as clicking the register switches and tapping the keys, which implies its use as an idiophone. The inductive challenge is to infer statements about the relations and objects that are true but unobserved. Due to the drawbacks of traditional taxonomies, the reasoner would not be able to discover new knowledge about the particular individual played as an idiophone in this specific example.

## 4.5 Summary

In this chapter, we have provided our investigation on some issues arising in the representation of knowledge about musical instruments. In order to demonstrate their drawbacks in complex query answering, we implemented two instrument taxonomies based on the well-known H-S system in OWL. We found that many instrument classification schemes exhibit insufficient or ill-defined semantics for our purposes, thus a more flexible representation is required. We demonstrated using different SPARQL queries that depending on the terminology and conceptualisation used by (ethno)musicologists, we obtain different results for the same instrument object. It also became evident, that ontologies that define relationships between entities are better than traditional taxonomies at providing meaningful answers to queries.

# Chapter 5

# Fundamentals of Automatic Ontology Generation

---

This chapter is organised in three parts. In Section 5.1, we will discuss the automatic ontology generation frameworks based on the type of input data and the proposed ontology generation tools to date.

In Section 5.2, we present the musical audio texture and timbre features, such as timbral texture features, rhythmic content features, and pitch content features. We also highlight machine learning techniques used in musical applications, in particular the Multilayer Perceptron (MLP), Support Vector Machine (SVM), and K-means algorithm. Subsequently, we will review music classification tasks (i.e. tag, mood, instrument identification).

Finally, we will overview our work on conceptualisation, and present Formal Concept Analysis along with examples, in Section 5.3.

## 5.1 Automatic Ontology Generation Frameworks

Ontology development is an expensive, time consuming and laborious task. There are two kind-of methods in order to construct an ontology. $i$) providing tools (e.g. editors, consistency checkers, and ontology import tools) which are employed by knowledge engineers or domain experts to construct the ontology, $ii$) providing semi-automatic or automatic systems depending on machine learning and natural language processing techniques which are designed to extract concepts and relations from structured and unstructured data for instance databases and text. In this section, we will review automatic ontology generation systems.

### 5.1.1 Types of Input Data

Automatic Ontology Generation Systems use a diverse spectrum of technologies to develop ontologies completely from scratch, or enrich or adapt a pre-existing ontology in a semi-automatic manner working with various resources. Within the last few years, a number of ontology generation systems have been proposed. These systems can be classified based on the type of data that they are using in the training phase: structured, semi-structured, and unstructured types.

*Unstructured Data:* is natural text, such as books and journals. There exists a requirement of more processing than in the case of semi-structured and structured data. The methods proposed for learning from unstructured data usually rely on natural language processing. The shallow text processing with statistical analysis is among the natural language processing techniques. It employs a shallow parser to acquire noun phrases based on the frequency count of noun and noun phrases in documents which are retrieved from the web to obtain concepts and taxonomical relations. For instance, Sanchez & Moreno [2004] proposed a method utilising a search engine to retrieve the related pages based on keywords, and then analyse these web sites to discover essential candidate concepts for a domain. Thus, it retrieves the ontology concepts that are related or subsumed by a keyword.

There are also systems completely based on NLP. Sabou *et al.* [2005a] proposed a system which employs a typed dependency grammar and parser to obtain the relation between syntactic entities. Additionally, a set of syntactic patterns are used to discover relations among words in order to create an ontology. Their ontology extraction steps are: dependency parsing, syntactic patterns, ontology building, and ontology pruning.

Cimiano *et al.* [2005] developed an automatic approach for acquiring concept hierarchies from a text corpus. They parse the corpus to tag words with part-of-speech and generate parse trees for each sentence. The verb/subject, verb/object and verb/prepositional phrase dependencies are extracted from these parse trees, and then finally, Formal Concept Analysis is used to obtain the hierarchical structure of ontologies. More information on Formal Concept Analysis will be given in Section 5.3.

*Semi-structured data:* is text in HTML, XML files. Building ontology from semi-structure data uses both traditional data mining and web content mining techniques. Davulcu developed a system to detect HTML regularities in the Web documents in order to obtain hierarchical semantic structures as XML Davulcu *et al.* [2004]. As complementary they use tree mining algorithms to identify key domain concepts and their taxonomical relationships. In study [Karoui *et al.*, 2004; Nacéra Bennacer & Karoui, 2005], HTML documents have been used to identify the candidate concepts and build a database table with utilisation of clustering method. Considering the fact that semi-structured data provides more semantics than unstructured data, there isn't any doubt that more structured input data yields richer results. However, the vast majority of available knowledge is available as unstructured data.

*Structured data:* are the databases and dictionaries. Therefore, it is possible to extract parts of the ontology using the available structural information. Examples of structured information sources are database schemas, existing ontologies and knowledge bases. The central problem in learning from structured data is to determine which pieces of structural information can provide relevant knowledge.

### 5.1.2 Ontology Generation Tools: State of the Art

Ontology generation tools can be divided into two different categories. First, those which mainly focus on ontology generation from plain text and second, those that mainly focus on ontology generation from semi-structured data. There are a number of applications of ontology generation tools which are using unstructured text, taking advantage of automatic acquisition methods to automatically obtain an ontology.

For instance, Wong *et al.* [2009] proposed a hybrid system that consist of lexical simplification, word disambiguation and association inference for obtaining coarse-grained relations among potentially ambiguous and composite terms based on Web resources. However, the output of the proposed system is a lightweight ontology which can be seen as a taxonomy with mainly is-a relations. ORE (Ontology Repair and Enrichment) is a semi-automatic ontology enrichment tool which offers an assistance to knowledge engineers to identify issues in their knowledge base and fix them [Lehmann & Bühmann, 2010]. ORE combines state of the art techniques from ontology debugging and supervised learning in OWL, and provides recommendations for repairing and enriching a knowledge base through the use of supervised machine learning on the corresponding dataset.

LexOnt is yet another ontology generation tool — that is a plugin for the Protege ontology editor — which interacts with the user to facilitate the ontology development process [Arabshian *et al.*, 2012]. It is based on a semi-automatic approach which builds the ontology iteratively by utilising external knowledge such as Wikipedia and WordNet. The process involves three steps: *i*) match-terms to the Wikipedia page description of the category, *ii*) find synonymous words from the Wordnet, and *iii*) add terms to the ontology and rank in accordance with the external knowledge base. Subsequently, the user decides on concept names and object properties, that are more suitable for the ontology. As a result, it allows user to obtain an ontology including concepts and properties based on common terms and phrases within a corpus and domain.

Contrary to the ontology generation tools that we reviewed above, RelExt [Schutz & Buitelaar, 2005] and OntoGen [Fortuna *et al.*, 2006] are semi-structured text based ontology generation tools. RelExt utilises a pre-existing ontology and enhances it for the application domain. It identifies relevant triples over concepts from an existing ontology. Instead of is-a relations it extracts relevant terms and verbs in an effort to discover concepts and properties from a domain-specific text collection. On the other hand, OntoGen targets providing concept suggestion and concept naming, as well as ontology and concept visualisation to an ontology engineer via a fast semi-automatic ontology construction approach.

## 5.2 Semantic Analysis of Musical Audio

In this section we introduce the audio features and research areas which are related to our aim. We start by discussing various acoustic features and their specific characteristics. We then introduce machine learning techniques which will be used in our experiments, and conclude the section reviewing the main research fields in music classification.

### 5.2.1 Musical audio texture and timbre

Extracting information from audio recordings is an essential building block of content-based audio analysis. For that reason, an outline of the acoustical features is necessary for describing feature extraction techniques we discuss in this work. The following brief review highlights acoustic features in three categories: timbral texture features, rhythmic content features, and pitch content features.

*Timbral textural feature*

The musical term timbre is used broadly to refer to the variability in sonic characteristics that different instruments produce. Although its definition has been a matter of some debate, in the context of music analysis, timbre has been generally interpreted as the time-varying spectral envelope. Two most commons ways to obtain spectral envelopes are Linear Prediction and the Cepstrum; both are well established in speech communication research. *Linear Prediction* analysis used to derive formant information (e.g. frequency and bandwidth) using the source-filter model of the vocal tract. In Linear Predictive Coding the production of sound in resonant systems is approximated by a source-filter model. In this model, an excitation is passed through a resonant all-pole synthesis filter (inverse of the analysis filter). The analysis filter coefficients are estimated on a short-time basis from the auto-correlation of the signal. However, much research has focused on deriving more stable transformations of the parameters, since the LPC coefficients by themselves cannot be reliably quantised without causing instability. A solution to this problem is provided by using *Line Spectral Frequencies* [Kabal & Ramachandran, 1986]. LSFs are popular due to their excellent quantisation characteristics and consequent efficiency of representation. They are derived from the polynomial coefficient of the inverse filter associated with Linear Predictive Coding (LPC) analysis. The most important advantages of LSF features compared to direct form LPC coefficients are their simple frequency domain interpretation and their robustness to quantisation [Fazekas & Sandler, 2007]. Both LPC and LSF coefficients can be used to characterise the same aspect of timbre: the formant structure of a sound.

An alternative to the physical modelling used in linear prediction is perceptual modelling. As a way of replicating many of the behaviours of the components in the human auditory system, *mel-frequency cepstrum* is widely used which parameterises the shape of an audio spectrum after warping the frequency axis to roughly represent the salience of different frequency bands in the auditory system. In other words, MFCCs are derived from the cepstrum which is the inverse of the Foruier Transform of the log spectrum. When the log-spectrum is provided in the perceptually defined Mel Scale, the cepstrum coefficients are called *Mel Frequency Cepstral Coefficients*. The Mel-frequency cepstral coefficients are therefore designed to represent perceptually salient aspects of spectral shape in a few coefficients. They are usually employed compressing and modelling frequency distributions [Brown & Deffenbacher, 1979]. Another important feature of MFCCs is the reduction of data through the elimination of redundancies using the Discrete Cosine Transform (DCT) [Merhav & Lee, 1990]. It is worth to note that the DCT plays a significant role in audio analysis and was initially used in speech analysis [Young *et al.*, 1993]. To capture some measure of temporal variability, the MFCCs are often augmented with their variance, deltas and sometimes also the double-deltas.

*Rhythmic content features*

Rhythmic structure and organisation is another essential feature of music. However, computing the rhythm, tempo and beat of music is a challenging task for automated systems. A beat is a fundamental rhythmic element of music. The rhythm describes the timing relationships between musical events within a piece. The feature set for representing rhythmic structure is based on detecting the most salient periodicities of the signal. In some studies, for instance, it is extracted from a beat histogram. As introduced in [Tzanetakis, 2002] and [Li *et al.*, 2003], the amplitude envelop, in time domain, of each band is initially extracted by decomposing the music signal into a number of octave frequency bands. Subsequently, the autocorrelation is obtained based on the envelopes and the dominant peaks of the autocorrelation function — corresponding to the various periodicities of the signal's envelope — are accumulated into a beat histogram where each bin corresponds to a time lag.

*Pitch content features*

Pitch analysis measures the pitch content of music related to fundamental frequency. In some studies, for instance, the pitch content is extracted from the pitch histograms where the dominant peaks are accumulated through autocorrelation function, as described in [Martin *et al.*, 1998] and [Luc *et al.*, 2006]. Each peak can be assigned to a musical note. For example, there are more chord changes in Jazz music. On the other hand, there is a high degree harmonic variability in classical music, as described in [Heittola, 2003].

### 5.2.2 Machine learning and musical applications

Machine learning is a field of research with the aim of creating computer algorithms that simulate or approximate human decisions. Making computers learn through the utilisation of machine learning algorithms facilitates numerous new applications for computers. It also contributes to a greater comprehension of human learning abilities. There are many algorithms that proved success in many problems with clearly defined objectives. For comprehensive introductions the reader is referred to Mitchell [1997], MacKay [2003], Marsland [2009]. Here we introduce some general concepts in machine learning as well as the application of such techniques to musical data which we will be used in this thesis.

An important issue in most applications is the amount of data set available for training a classification algorithm, which means that the algorithm's ability to generalise correctly will likely to deteriorate as the dimensionality of the input data becomes large. Machine learning algorithms, therefore, must be supplied with relatively small number of informative input features and eliminate irrelevant input dimensions in order to achieve a greater generalisation from training data. Thus, it is important to have an *feature extraction* process which involves simplifying the amount of resources required to describe a large set of data accurately projecting higher dimensionality onto a smaller number of dimensions. In previous section we discussed compact feature representations such as MFCCs and linear prediction, that allows to compress information from on the order of 1000 dimensions down to perhaps 8 or 32 dimension intended to capture the important aspects of the signal.

Other more general dimension reduction strategies attempt to automatically compress high-dimensional data into a smaller number of dimension. Principal Component Analysis (PCA) is one of the most common techniques which identifies the largest variance of data. It produces a new orthogonal basis in which most of the variance is captured in the first few dimensions. As a result, it achieves a data reduction by keeping only some of the principal components.

An alternative to dimension reduction is *K-means algorithm* which does not transform the input features but summarises them. The main idea is to specify in advance how many clusters are being sought which defines the parameter *k*. There are three main phases after choosing k points as cluster centres: *i*) all instances are assigned to the closest centre point (i.e. centroid); *ii*) centroids are calculated by taking average of the samples; *iii*) these centroids are taken to be new centre values for their respective clusters. While K-means clustering is a commonly used data clustering technique for unsupervised learning tasks, it can also be deployed like a data reduction technique that summarises a signal by partitioning observations into *k* clusters. Each observation belongs to the closest cluster which means that they can be represented by the closest cluster centre points. The obtained centre points can be used as a summary of a signal as well as a codebook vector containing *k* codevectors. It is extensively used for the vector quantisation of the LSF in speech coders as a process of redundancy removal that makes the effective use of nonlinear dependency and dimensionality by compression of spectral parameters [Paliwal & Atal, 1993]. Vector Quantization is one of the preferred methods to map vast amount of vectors from a space to a predefined number of clusters. As a results, a large set of feature vectors are taken and a smaller set of measure vectors is produced which represents the centroids of the distribution. Similar techniques have been experimented in speaker identification frameworks, for example in [Rosenberg & Soong., 1986; Gill *et al.*, 2010; Soong *et al.*, 1985; Pelecanos *et al.*, 2000].

Machine learning algorithms can be divided into two categories: *i*) supervised algorithms and *ii*) unsupervised algorithms. In some pattern recognition problems, the training data includes a set of input vectors without any corresponding target values. The objective of unsupervised learning problems is to learn categories of similar samples within the dataset, in this case it is called clustering, or to discover the distribution of data within the input space, generally known as density estimation, or to transform the data from a high-dimensional space into two or three dimensions with regards to visualisation. As an unsupervised algorithm, K-means is utilised thoroughly in music information retrieval, see [Kaminskyj & Czaszejko, 2005; Agostini *et al.*, 2003; Fujinaga *et al.*, 1998; Eronen, 2001].

On the other hand, the success of supervised classification is usually assessed by testing the trained classification system on an independent set of data, so called test data, in which the true classifications are known, yet not provided to the machine. The rate of success on test data allows having an objective way of measurement on how well the classification task has been performed. *Multi-Layer Perceptron (MLP)* is among the most common types of neural networks in supervised pattern recognition algorithms. The algorithm is based on an error-correction learning rule. Its effectiveness results from parallel and distributed structure, and the ability to learn. The error back-propagation learning involves two main stages during the learning phase: a forward pass and a backward pass. In the forward stage, an input vector is applied to the sensory nodes of the network by propagating its effect through the network layers. This process allows us to obtain a set of outputs as the actual response of the network. The error signal is then propagated backward through the network against the direction of syntactic connections. This mechanism in neural network training is called back-propagation. The main purpose is to modify the synaptic weights in order to minimise the error between a desired output and an actual response of the network [Haykin, 1998] The most exhaustive studies on automatic instrument classification using mural networks can be found in [Kostek, 1999] and [Park, 2004].

Alternatively, *Support Vector Machine (SVM)* is a very efficient classifier in pattern recognition which has been widely used in various machine learning tasks, since it has been popularised by Vapnik, in [Cortes & Vapnik, 1995]. SVM aims to discover the best separating hyperplane leading to the largest distance separation between classes [Bishop, 2006]. It determines the data points in each class that lie closest to the margin (decision boundary), which are called *support vectors*, and minimises the probability of error relative to the learnt density model. In some cases, the dataset may not be linearly separable. The original input space, therefore, needs to be mapped into a high-dimensional dot-product space known as *the feature space*, where each coordinate corresponds to one feature of the data items, in order to enhance linear separability. This method is called *kernel trick*. It is computationally less expensive than the explicit computation of the coordinates. There are a number of different basis functions for kernels that can be used in Support Vector Machines models. Some of the important kernels are RBF kernel, Polynomial kernel, Linear kernel. Polynomial kernels are less widely used than the RBF kernel, which maps data to a potentially infinite dimensional space. In [Chang *et al.*, 2010], it has been suggested that one reason might be that a polynomial kernel is not as nonlinear as RBF. Moreover, according to their detailed comparison between polynomial kernel with degree 2 and RBF kernel, the results have shown that the polynomial kernel gives a better accuracy than the RBF kernel which they found consistent with previous observations.

SVM successfully used in a number of classifications and percussion transcription tasks such as detection of segment that contains a percussive sound event or an individual drum [Helen & Virtanen, 2005; Steelant *et al.*, 2004; Gillet & Richard, 2005]. It has been shown that the performance of SVM method is much higher than other methods such as KNN, GMM [Lu *et al.*, 2003]. Additonally, it has also been pointed out that SVM is computationally more efficient than the KNN method. In another study, there has also been a comparison in which SVM classifier without any sequence modelling performed better than a HMM-based approach [Gillet & Richard, 2004].

Another consideration in machine learning is automated vs autonomous systems. In *automated systems*, the parameter setting and the selection of a preprocessing method are conducted by an external supervisor on a trial and error basis when a learning algorithm is applied to a particular problem. The automated systems are usually based on batch learning in which training data is processed together. On the other hand, *autonomous systems* involve online learning process in which new data arrives in a continuous stream and every training instance is processed just once. It is sometimes desirable to have an online algorithm which learns at the same time as it outputs decisions. However, it is worth to point out that batch algorithms may be adapted for online application or algorithms may intrinsically be amenable to online use, as in [Davy *et al.*, 2006; Artac *et al.*, 2002].

### 5.2.3 Music classification based on audio analysis

Music Information Retrieval applies machine learning and other techniques to topics related to musical information. A few of the popular subjects providing a content based audio analysis background are musical instrument identification, music tag prediction and music mood classification. In the rest of this section, the above mentioned music information retrieval tasks will be outlined based on the state-of-the art in each area.

*Music Tag Prediction*

Tags are useful text-based labels that encode semantic information about music content (Genres, instrumentation, geographic origins, emotions). Music tag prediction aims to describe the best terms for a given song [Hoffman *et al.*, 2009]. However, there are numerous ways of collecting experimental data and perhaps only a few songs and artists have been annotated accurately.

That is, it is often difficult to know what makes a tag accurate and what kind of inaccuracies are tolerable [Kim *et al.*, 2009]. The popularity bias, thus, results in significant amount of disproportion for tags. Considering the fact that tags are not annotated by experts, the obtained meta-data is likely to include misleading information. Moreover, social network users usually prefer using the most frequent tags rather than contributing new tags to the system. For instance, roughly a third of 5,265 artists received no tags for any of their tracks, while even amongst the artists with tagged tracks, roughly a third have no more than five distinct tags per track on average in [Levy & Sandler, 2009]. Evidently, music recommendation or search systems based on tags will obtain bias results utilising tracks by well-known and well-liked artists [Lamere, 2008]. Thereby, it provides a practical motivation to support MIR models with information extracted from audio signals which may improve the quality and variety of results returned to set of search queries.

In the context of music tag identification, The mostly used algorithms are standard binary classifiers, such as Support Vector Machines or AdaBoost [Trohidis *et al.*, 2008b]. These classification approaches use standard training and testing phases. Thereby, the classifier predicts the musical tags of a testing dataset. Gaussian Mixture Model (GMM) is another well known technique that has been widely used in music tag prediction [Turnbull *et al.*, 2008]. The approach has shown reasonable performance on diverse set of songs and retrieved relevant songs given a text-based query.

*Music Mood Classification*

Music is an art form that can influence our mood and even more life style. Social and psychological outcomes of music have been studied extensively for decades [Li & Ogihara, 2004]. Note that the classification of conceptual structure of psychological moods of human is an intensely challenging problem on its own.

Traditionally, MIR approaches have been based on standard meta-data such as the name of the composer, the album title, the style of music, and so on. However, there is also a need to use higher level features (e.g., beat, tempo, and mood.) in order to obtain better information. To extract emotional features in music, numerous studies established to understand what people might feel when they listen to music. For example, Trohidis *et al.* [2008a] proposed a multi-label classification of music emotions utilising four algorithms, including binary relevance (BR), label powerset (LP), random k-labelsets (RAKEL), and multilabel k-nearest neighbor (MLkNN). These algorithms were evaluated and compared using rhythm and timbre features. The experiments are conducted on a set of 593 songs with 6 clusters of music emotions based on the Tellegon Watson-Clark model. The employed emotions are 'amazed-surprised',' happy-pleased', 'relaxing-calm', 'quiet-still', 'sad-lonely' and 'angry-fearful'. The RAKEL algorithm performed relatively better than others. In another automatic mood detection approach employed by Luc *et al.* [2006], where a Gaussian Mixture Model is used to model each feature set regarding to each mood cluster such as *contentment*, *depression*, *exuberance* and *anxious/frantic*. The dataset has been annotated by experts. The results indicate that the timbre features are much more important than the rhythm features in classifying *Contentment* and *Depression*, while the rhythm features are slightly more important to discriminate *Exuberance* from *Anxious/Frantic*.

*Musical Instrument Classification*

To automate musical instrument identification, various approaches have been developed based on isolated notes, solo performances, or complex mixtures (see [Klapuri & Davy, 2006] for a thorough review). The results obtained depend on three main factors: the databases used during the learning and testing stages, the features selected to characterize the timbre of the instruments, and the classification methods.

The isolated note or solo performances present an advantage of simplicity and tractability, since there is no need to separate the sounds from different instruments. For example, Chétry *et al.* [2005] proposed a system based on Line Spectral Frequencies (LSF), which are derived from a linear predictive analysis of the signal and represent well the formant structure of the spectral envelope. The instrument identification unit of our system is based on this model. K-means clustering is used to construct a collection of LSF feature vectors, called codebook. The collection of K codevectors (LSF vectors) constitutes a codebook, whose function is to capture the most relevant features to characterise the timbre of an audio segment. Hence, to a certain extent, K-means clustering can be viewed here both as a classification and a feature selection technique, as described in [Barthet & Sandler, 2010b]. The classification decisions are made by codebook-to-codebook distance measurement based on the minimisation of an error. The system achieved 95% performance on a dataset comprising 4415 instrumental sound instances.

In another study, Vincent & Rodet [2004] proposed a system based on Gaussian Mixture Models (GMM) which were trained and tested on isolated notes and solo recordings. The dataset was gathered by extracting 2 excerpts of 5 seconds from each of the 10 solo recordings used in the experiment. This approach yielded 90% accuracy. Essid *et al.* [2006] proposed a system tested on a relatively large dataset. The same classification technique, GMM, was compared to Support Vector Machines (SVM) with different audio features. Their system obtained a 12% performance improvement compared to a system based on the SVM classifier, leading up to 87% of accuracy for 0.5s-long audio segments. Furthermore, the performance of their system increased from 6% points up to 93% of accuracy, using SVM on “5 s-long audio segments”.

There are, however, only a few studies where instrument recognition produces a hierarchical instrument structure. For example, Martin [Martin *et al.*, 1998] proposed a system which was based on three different hierarchical levels: 1) pizzicato (plucked) and sustained sounds, 2) instrument families such as strings, woodwinds, and brass 3) individual instruments for the corresponding instrument families. While the dataset consisted of 1023 solo tones samples from 15 instruments, the recognition rate obtained with this system was 90% for instrument family and 70% for individual instruments. Other hierarchical systems have been developed by Eronen & Klapuri [2000], Kitahara *et al.* [2003] and Peeters & Xavier [2003]. The overall correct identification rate of these systems are in the range of 35% to 80% for individual instruments, and 77% to 91% for instrument family recognition. In general, the problem with hierarchical classification systems is that the errors at each level propagate increasingly to the other levels of the hierarchy.

## 5.3 Conceptual Analysis

In this section, we discuss the studies on conceptual analysis and automatic ontology generation to date. Additionally, we will present the Formal Concept Analysis technique that constitutes the core of our system.

### 5.3.1 Concepts and Conceptual Analysis in Ontology Generation Systems

Creating a class hierarchy is an important aspect of ontology design. Establishing such a hierarchy is a difficult task that is often accomplished without any clear guidance and tool support. Yet, the most commonly used hierarchical data mining techniques such as Hierarchical Agglomerative Clustering [Cimiano *et al.*, 2005] and Decision Trees [Elsayed *et al.*, 2007] do not take into account the relationships between objects. For instance, decision trees employed for the discrimination of instrument sounds have usually produced worse results than other classification techniques [Peeters, 2003], [Herrera *et al.*, 2001]. Alternatively, they've provided information on the characteristics of the features and values that discriminate the pitched instrument classes [Jensen & Arnspang, 1999], [Wieczorkowska, 1999]. Latest improvements to basic decision tree, for example Ada Boost [Freund & Schapire, 1996], may deliver success which can be as good as other classification algorithms. Nevertheless, they do not provide an applicable solution to knowledge representation issues of ontology systems. This problem becomes even more apparent considering the multi-relational nature of musical data.

On the other hand, Formal Concept Analysis (FCA) allows to generate and visualise the hierarchies relying on the relationships of objects and attributes. FCA, also known as concept lattice, was first proposed by German mathematician Wille in 1982 [Wille, 1982]. It has been used in many software engineering topics such as the identification of objects in legacy code, or the identification and restructuring of schema in object-oriented databases [Snelting, 2003]. In the broad sense, these works are important since ontologies provide the basis for information and database systems [Yahia *et al.*, 1996]. Various specification techniques for hierarchical design in object-oriented software development have been proposed in [Godin & Valtchev, 2005]. This study suggested alternative designs for FCA by not only utilising attribute-based categorisations but also using different levels of specification details (e.g., objects, attributes, methods) in order to obtain the class diagram of the software system. Furthermore, FCA has been used in conceptual knowledge discovery in collaborative tagging systems [Kang *et al.*, 2009], and web mining studies in order to create adaptive web sites utilising user access patterns extracted from Web logs [Vasumathi & Govardhan, 2009].

By offering a solution to bridge the gap between data and knowledge automatically, FCA has generated considerable research interest. Recently one of the influential ideas of automatic ontology generation has been proposed by Maedche & Staab [2001] and can be described as the acquisition of a domain model from data. FCA has also been used in other systems [Cimiano, 2006; Stumme *et al.*, 1998].

### 5.3.2 Formal Concept Analysis

Formal Concept Analysis is a mathematical theory of concept hierarchies which is based on Lattice Theory. The keystone of the FCA is the notion of formal context. A *formal context* is defined as a binary relation between a set of objects and a set of attributes. In a formal context, a pair, formed by a set of objects and a set of attributes that uniquely associate with each other, is called a *formal concept*. The set of objects are referred to as *extent closure*, and the set of attributes are referred to as *intent closure*. In the reminder of this Section, the notions underlying Formal Concept Analysis are defined following Ganter et al.'s formalism [Ganter *et al.*, 2005] and illustrative examples are given.

**Definition 7** (Formal Context)**.** A formal context $\mathbb{K} := (G,M,I)$ is composed of a set of objects G, a set of attributes M, and a binary relation $I \subseteq G \times M$. We call I the incidence relation and read $(g,m) \in I$ as the object g has the attribute m. The relation of an object to an attribute is denoted as $gIm$.

A formal context can be represented by a cross table where the rows are defined by the object names and the columns are defined by the attribute names. In Table 5.1, the formal context is composed of three objects representing three instruments (*cello, piano, violin*), and three attributes representing three instrument properties (*vibrating string, sound initiation process:Bowed*, and

| | vibrating string | sound initiation process:Bowed | sound initiation process:Struck |
|---|---|---|---|
| Cello | ✓ | ✓ | - |
| Piano | ✓ | - | ✓ |
| Violin | ✓ | ✓ | - |

Table 5.1: Cross table representing a formal context between a set of instruments (cello, piano, violin) and a set of attributes (vibrating string, bowed, struck).

*sound initiation process:Struck*). A symbol "✓" in row $g$ and column $m$ means that the object $g$ has the attribute $m$ — that is, the object has the indicated attributes (e.g. the *cello* instrument has the attributes "*vibrating string*' 'and "*sound initiation process:Bowed*").

**Definition 8** (Derivation Operators)**.** For a subset $A \subseteq G$ of objects, we define a set of attributes common to the objects in A as:

$$A' := m \in M \mid gIm \,\forall\, g \in A \tag{5.1}$$

and reciprocally, for a subset $B \subseteq M$ of attributes we define a set of objects which have all attributes in B as:

$$B' := g \in G \mid gIm \,\forall\, m \in B \tag{5.2}$$

The following statements are the derivation operators for a given context $(G,M,I)$, its subsets $A, A1, A2 \subseteq G$ of objects as well as its subsets $B, B1, B2 \subseteq M$ of attributes:

$$A_1 \subseteq A_2 \Rightarrow A_2' \subseteq A_1' \text{ and } B_1 \subseteq B_2 \Rightarrow B_2' \subseteq B_1' \tag{5.3}$$

$$A \subseteq A'' \text{ and } B \subseteq B'' \tag{5.4}$$

$$A' = A''' \text{ and } B' = B''' \tag{5.5}$$

$$A \subseteq B' \Leftrightarrow B \subseteq A' \Leftrightarrow A \times B \subseteq I \tag{5.6}$$

The first derivation of the set of objects ($A$) is the attributes ($A'$) which are possessed by those objects, and we can apply the second derivative operator to obtain the objects ($A''$) possessed by these attributes ($A'$). In addition, if a selected object set is enlarged, then the common attributes of the larger object set is among the common attributes of the smaller object set. The same principle applies for the enlarged attribute set.

**Definition 9** (Formal Concept)**.** A pair (A, B) is a formal concept of $(G, M, I)$ if and only if

$$A \subseteq G,\ B \subseteq M,\ A' = B,\ \text{and}\ B' = A. \tag{5.7}$$

The set A is called the **extent**, and B is called the **intent** of the formal concept (A, B).

**Example 1.** Table 5.1 gives an example of formal context based on M=$\{$*vibrating string, sound initiation process:Bowed, sound initiation process:Struck*$\}$, G=$\{$*Cello, Piano, Violin*$\}$ and the binary relation "*I*" represented by the "✓" (has/has not) in the cross table. As intent(*Cello*)= $\{$*vibrating string, sound initiation process:Bowed*$\}$, and extent(*vibrating string, sound initiation process:Bowed*)=$\{$*Cello, Violin*$\}$, ($\{$*Cello*$\}$, $\{$*vibrating string, sound initiation process:Bowed*$\}$) is not a formal concept of (G, M, I). However, intent(*Piano*)=$\{$*vibrating string, sound initiation process:Struck*$\}$, extent(*vibrating string, sound initiation process:Struck*)=$\{$*Piano*$\}$, therefore the pair ($\{$*Piano*$\}$,$\{$*vibrating string, sound initiation process:Struck*$\}$) is a formal concept.

**Definition 10.** Let (A1, B1) and (A2, B2) be two formal concepts of a formal context (G, M, I), (A1, B1) is called the **subconcept** of (A2, B2) and denoted as $(A1, B1) \leq (A2, B2)$, if and only if $A1 \subseteq A2$ ($\Leftrightarrow B2 \subseteq B1$). Equivalently, $(A2, B2)$ is called the **superconcept** of $(A1, B1)$. The relation $\leq$ is called the hierarchical order (or simply order) of the formal concepts.

**Example 2.** Let C1=($\{$*Cello, Violin*$\}$,$\{$*vibrating string, sound initiation process:Bowed*$\}$) and C0=($\{$*Cello, Violin, Piano*$\}$, $\{$*vibrating string*$\}$) be two formal concepts by considering Table 5.1. As $\{$*Cello, Violin*$\}$ $\subseteq$ $\{$*Cello, Violin, Piano*$\}$ and $\{$*vibrating string*$\}$ $\subseteq$ $\{$*vibrating string, sound initiation process:Bowed*$\}$, C0 is a superconcept of C1. Equivalently, C1 is called a subconcept of C0.

**Definition 11.** The family of concepts which obeys the above mathematical axioms is called a concept lattice. The lower bound of the concept lattice is called **infimum**, and its upper bound is called **supremum**.

### 5.3.3 Many valued contexts

In the real world the attribute is not only a property that an object may have nor not have. Attributes can have values. For instance, the "shape" attribute may have many values, such as "conical", "circular" or "rectangular". As a result the context is also many-valued compared the context we mentioned before which is one-valued.

**Definition 12.** A many-valued context is a tuple $\mathbb{K} := (G, M, (W_m)_{m \in M}, I)$ where $G$ is a set of objects, $M$ is a set of attributes, $W_m$ the set of possible values for the attribute $m \in M$, and $I$ is a relation. $I \subseteq G \times \{(m,w) | m \in M, w \in W_m\}$ with the constraint $(g,m,w_1) \in I$ indicates that object $g \in G$ has value $w \in W_m$ for attribute $m \in M$.

From a many-valued context, a concept lattice cannot be computed directly. One has to transform it first into a one-valued context. This transformation is called *conceptual scaling*.

**Definition 13.** A conceptual scale for a subset $B \subseteq M$ of attributes is a (one-valued) formal context $\mathbb{S}_B := (G_B, M_B, I_B)$ with $G_B \subseteq \times_{m \in B} W_m$.

After interpretation of each attribute by means of a context which is called *conceptual scale*, there is a need to perform conceptual scaling. Thus, for any subset $S$ of scales, we can now transform the many-valued context into a one-valued one:

**Definition 14.** The derived context $\mathbb{K}_S$ is defined by $\mathbb{K}_S := (G, \bigcup_{\mathbb{S}_B \in S} M_B, I_S)$ with $(g,n) \in I_S$ if there exists a scale $\mathbb{S}_B \in S$ with $m \in M_B$ and $w \in W_m$ where $(g,m,w) \in I$ and $(g,n) \in I_B$.

To obtain the scaled context, each many-valued attribute has been replaced by the corresponding row of the scale. We applied this method on instrument many-valued attributes, such as *sound initiation process, reeds no* and *valves*, in order to understand if the property is an 'object property' or a 'data property'. A many-valued context example is given in Table 5.2 illustrating the transformation of the many-valued context into a one-valued context.

**Example 3.** In Table 5.2, the two attributes *Struck* and *Bowed* have been derived from a many-valued attribute, called *sound initiation process*, which has a set of possible values $W_{sound\ initiation\ process} := \{Struck, Bowed\}$. Thus, these attributes have been replaced by scale attributes: e.g., *sound initiation process:Bowed*, and *sound initiation process:Struck*.

| | vibrating string | sound initiation process |
|---|---|---|
| Chordophone | ✓ | Struck, Bowed |
| Cello | ✓ | Bowed |
| Violin | ✓ | Bowed |
| Piano | ✓ | Struck |

(a) Many-valued context

| | vibrating string | sound initiation process:Bowed | sound initiation process:Struck |
|---|---|---|---|
| Chordophone | ✓ | ✓ | ✓ |
| Cello | ✓ | ✓ | - |
| Violin | ✓ | ✓ | - |
| Piano | ✓ | - | ✓ |

(b) Derived one-valued context

Table 5.2: A naive scaling and cross table of a formal context

### 5.3.4 Lattice Pruning

Lattice Pruning is the process of removal of "empty or unnecessary repetitions" of concepts, objects or attributes based on any of the necessity and stability notions that are defined by knowledge engineers. The concept lattice of $(G,M,I)$ is the set of all formal concepts, ordered as subconcepts-superconcepts, that depicts particularities and relationships of our data. Each node represents a formal concept. However, each of these nodes involve object and attribute repetitions in order to illustrate the relationship among the nodes. Therefore, in order to formally define the transformation of the lattice into the partial order or a concept hierarchy, to subsequently make it simpler and more readable, we used a pruning technique called *reduced labelling*. Figure 5.1 depicts the representation of a concept lattice for the formal context between a set of instruments and attributes which was given in table 5.1.

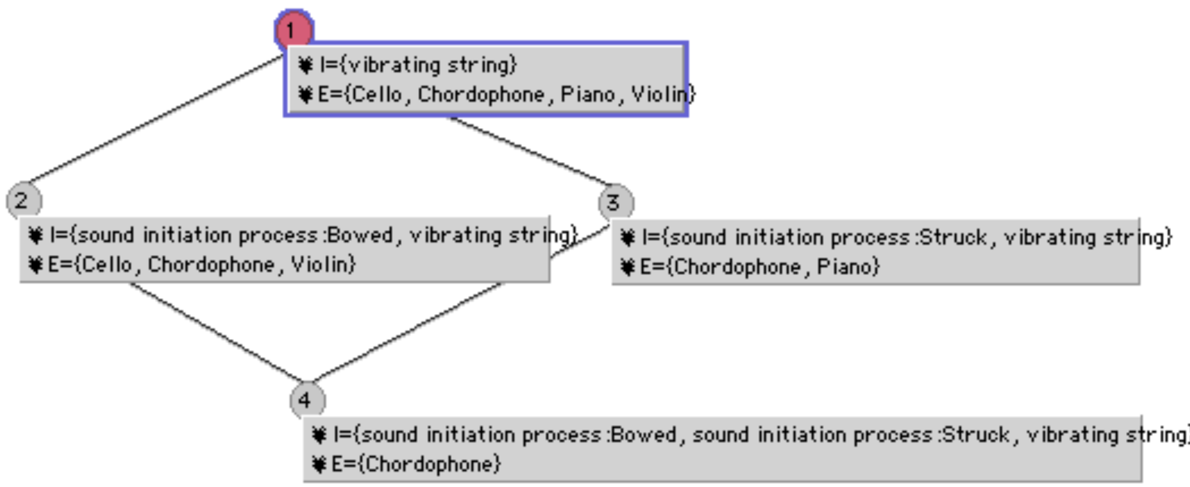



Figure 5.1: A concept lattice for objects consisting of the instruments (i.e. Chordophones, Cello,Violin, Piano) and attributes (i.e. vibrating string, bowed, struck). Numbers refer to its conceptual order. This graph is produced by using a software called FCA Stone[1].

The principle is to have each object entered only once in the hierarchical form [Krohn *et al.*, 1999]. In other words, we remove any terms from the inner node which are the same as their children [Cimiano, 2006]. The objective of lattice pruning isn't just to acquire a hierarchy to display the concepts in a correct way, but an ideal output that can be transformed into an OWL represen-

[1] http://www.fcahome.org.uk/fcasoftware.html

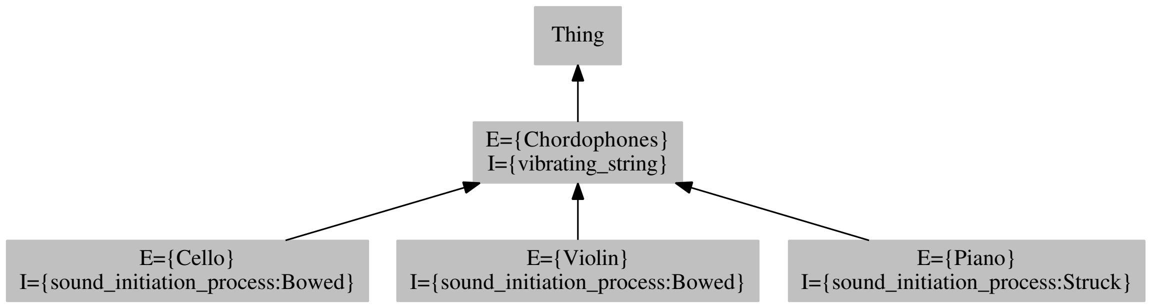



Figure 5.2: An illustration of the pruned concept lattice which is depicted in Figure 5.1.

tation. The visible difference, compared to prior Formal Concept Analysis studies, is we remove the infimum edges of the empty and the non empty sets on the lattice form. Thus, the symmetric reflection of each superconcept is removed. A simple illustration can be given for the lattice diagram which is depicted in Figure 5.1, We refer to each concept with its numbers, such as $C1$ for concept 1. The pruning process consists of the following this steps: $i$) remove any concept with no object or attribute; $ii$) retrieve the union of all objects and attributes from the concepts which possess the corresponding attribute (e.g. union of objects: *Chordophone, Cello, Piano, Violin*; attributes: *vibrating string, bowed, struck; corresponding attribute: vibrating string.*); $iii$) retrieve the intersection of all objects from the concepts the which possess the corresponding attribute (e.g. *Chordophone*). Therefore, we remove the concept that possess the union of all the corresponding attributes and the intersection of all objects (e.g. $C4$). $iv$) We have also deleted all the intersection of objects from the concepts which does not subsume each other, such as Chordophone object of $C2$ and $C3$; $v$) Then, the reduced labelling technique is applied to the lattice diagram in parallel with the OWL representation. The purpose of this task is to remove any attribute of a concept that occurs in its superconcept (e.g. *vibrating string* in $C2$ and $C3$). However, we use this information for OWL representation before we remove it from the lattice, since this is the last remaining feature providing information about the concept order of lattice.

The final form of the lattice can be summarised as $C1$ (i.e. objects: *Chordophone*; attributes:

*vibrating string*), $C2$ (i.e. objects: *Cello and Violin*; attributes: *sound initiation process:Bowed*) and $C3$ (i.e. objects: *Piano*; attributes: *sound initiation process:Struck*). Figure 5.2 depicts the pruned concept lattice.

## 5.4 Summary

The algorithms and the research fields which constitute the foundations of our research have been outlined in this chapter. Firstly, we presented proposed automatic ontology generation frameworks discussing the type of inputs in conjunction with the utilised ontology generation tools to date.

In Section 5.2, we presented the music audio features (i.e. timbral texture features, rhythmic content features, and pitch content features). From these audio features, we will concentrate on the utilisation of the timbral texture features (i.e. MFCCs and LSFs) in our system. Subsequently, the general machine learning algorithms which have been used in most of the music classification systems are highlighted. In these algorithms SVM and MLP constitutes the basis for our musical instrument identification experiments, whereas the K-Means clustering method is used as a data reduction method in our system, which will be explained in Chapter 6. Furthermore, even though our experimental setup only involves instrument recognition, due to the fact that the focus of thesis is based on semantic audio analysis, we have provided a comprehensive overview on music classification studies (i.e. tag, mood, and instrument identification tasks).

Finally, previous works on conceptualisation techniques have been outlined. Since the conceptualisation constitutes the core of the automatic ontology generation, we described the small amount of literature in this section, which indicates that it is a new challenge. As we regard the Philosophical origin of ontologies, which dates back to Aristotle the Greek, this challenge can be seen as a new attempt to solve an old problem. Consequently, we have taken FCA as being a core of the proposed system, therefore we have also provided information on FCA in depth through examples. In the next chapter, we will outline evaluation techniques for ontologies.

# Chapter 6

# Automatic Generation of a Semantic Web Ontology for Musical Instruments

---

In this chapter, we introduce the general architecture of the automatic ontology generation system. The system is based on the foundations and definitions of the ontology and knowledge base structure presented in chapter 2.

This chapter is divided into four sections. First, we will briefly present the general architecture of the proposed system in section 6.1, and identify the data sets that have been used to examine the ontology generation system–namely isolated notes and solo music. In section 6.2, we will then describe the content based analysis part of the system. The parameters and methods that have been used for audio feature extraction and classification will be detailed in section 6.3. This system was tested on various musical instruments, wind and string families, using timbre features extracted from audio: namely Line Spectral Frequencies (LSFs), or Mel-frequency Cepstral Coefficients. Additionally, two classification techniques based on Multi-Layer Perceptron (MLP) neural network and Support Vector Machines (SVM) were tested to classify the analysed instrument recordings.

Finally, we will describe the conceptual analysis part of the system in section 6.4. With an emphasis on the formal concept analysis and lattice pruning components of the system, in order to describe the construction of the hierarchical structure of the obtained ontologies, we describe the method for converting conceptual hierarchies into a domain ontology.

## 6.1 The Architecture of the Ontology Generation System

The process of ontology engineering currently involves human supervision at every level. To create automatic ontology generation systems, we must cope with the rapidly increasing and large datasets through knowledge acquisition and automatic ontology construction. Thus, we present a hybrid ontology generation system for musical instruments, which the architecture is shown in Fig 6.1.

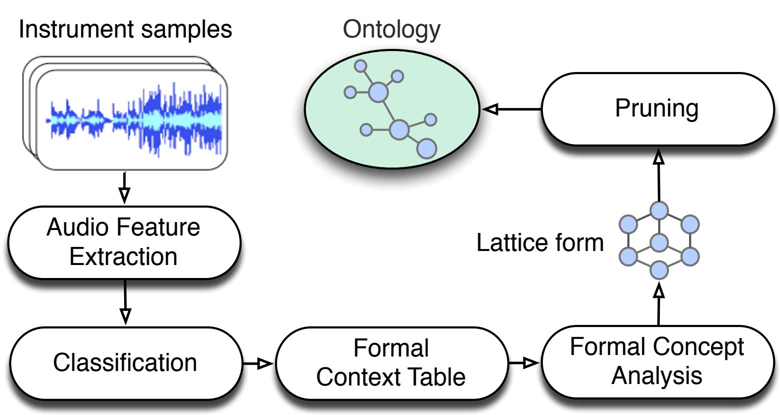



Figure 6.1: Automatic ontology generation system based on audio features.

The system aims to automatically obtain ontology designs in an OWL representation from pre-labelled (tagged) music audio collections. As discussed in section **??**, the proposed system delivers a few major contributions. Firstly, it permits the automatic construction of concept hierarchies of ontologies, and helps domain experts to provide an initial ontology draft to avoid the bulk of ontology engineering work. Secondly it establishes a formal bridge between concept hierarchies and OWL ontologies which fits with established knowledge representation techniques, in order to deliver a meaningful and beneficial approach. Thirdly, it is based on a domain independent methodology, so it can be applied to other domains as well.

The input of the system is a pre-labelled music audio collection and the output is an OWL document that represents the corresponding conceptualised structure of the data collection. The taxonomy of musical instruments given by Hornbostel and Sachs[1] was considered as the basis for instrument terminology and the initial hierarchical structure. In the architecture of the system (shown in Figure 6.1), there are five components: feature extraction, classification, formal concept analysis, lattice pruning, and OWL representation of the concept hierarchies. These components are tested by two main experimental design: *i*) content-based analysis and *ii*) conceptual analysis. We will outline the components of the automatic ontology generation system within these two experimental designs.

Content-based audio analysis involves audio feature extraction and classification components, which we have examined for various musical instruments, wind and string families, extracting timbre features (i.e. LSF and MFCCs) from audio samples in order to identify the instruments based on classification techniques (i.e. MLP and SVM). Conceptual analysis also involves FCA, lattice pruning and OWL representation components. This experimental design is dependent on the obtained output of the content-based analysis. The system conceptualises the results through FCA and prunes the obtained lattice form into a conceptual hierarchy. Finally, it converts concept hierarchies into a domain ontology by using Ontology Web Language. The reminder of this chapter describes each functional unit of our system in more detail.

[1] H. Doktorski, http://free-reed.net/description/taxonomy.html

## 6.2 Databases

Our experimental dataset consists of two sets of audio samples: one set of instruments' isolated notes and another set of solo performances, both collected from the following datasets: the Real World Computing (RWC) music collection[2], University of IOWA's Musical Instrument Samples (MIS)[3], McGill University Master Samples[4], and additional samples recorded at the Centre for Digital Music, Queen Mary University of London (QMUL).

The isolated note dataset contains recordings of 10 different musical instruments — 15 pre-defined classes/objects — *chordophones*[5], *aerophones*[6], *edge instruments, brass instruments, reed pipe instruments, bassoon, cello, clarinet, flute, oboe, piano, saxophone, trombone, tuba, violin* — and 12 musical attributes — *vibrating string, vibrating air, sound initiation process:Bowed, sound initiation process:Struck, reeds, edge, lip vibrated, reeds no:1, reeds no:2, valves:With valves, valves:Without valves, true flutes*.

The solo instruments dataset contains recordings of 8 musical instruments — 12 pre-defined classes/objects — *chordophones, aerophones, edge instruments, reed pipe instruments, bassoon, cello, clarinet, flute, oboe, piano, saxophone, violin* — and 9 musical instrument attributes — *vibrating string, vibrating air, sound initiation process:Bowed, sound initiation process:Struck, reeds, edge, reeds no:1, reeds no:2, true flutes*.

The data sets that have been used in the experiments summarised according to musical instruments in Table6.1. The dataset was randomly divided 4 times for cross-validation. In each experimental run 75% of the samples were used for training and 25% were used for testing. The overall results are obtained by averaging the results obtained in the 4 experimental runs.

[2] http://staff.aist.go.jp/m.goto/RWC-MDB/
[3] http://theremin.music.uiowa.edu/MIS.html.
[4] http://www.music.mcgill.ca/resources/mums/html/.
[5] Chordophone is a musical instrument category in which sounds are initiated by string vibrations.
[6] Aerophone is a musical instrument category in which sounds are initiated by a vibrating mass of air.

| Instrument Categories | Database | |
|---|---|---|
| | Isolated Notes # Data. | Solo Music # Data. |
| *Chordophone* | 2994 | 3162 |
| *Aerophone* | 2742 | 8550 |
| *Brass Instruments* | 354 | - |
| *Reed Pipe Instruments* | 2004 | 6564 |
| *Edge Instruments* | 384 | 1986 |
| *Cello* | 1446 | 624 |
| *Violin* | 888 | 1476 |
| *Piano* | 660 | 1062 |
| *Tuba* | 113 | - |
| *Trombone* | 241 | - |
| *Clarinet* | 648 | 1746 |
| *Saxophone* | 528 | 2328 |
| *Bassoon* | 564 | 1722 |
| *Oboe* | 264 | 768 |
| *Flute* | 384 | 1986 |
| Instrument Attributes | Isolated Notes # Data. | Solo Music # Data. |
| *vibrating string* | 2994 | 3162 |
| *sound initiation process:Bowed* | 2334 | 2100 |
| *sound initiation process:Struck* | 660 | 1062 |
| *vibrating air* | 2742 | 8550 |
| *lip vibrated* | 354 | - |
| *reeds* | 2004 | 6564 |
| *edge* | 384 | 1986 |
| *reeds no:1* | 1176 | 4074 |
| *reeds no:2* | 828 | 2490 |
| *valves:With Valves* | 113 | - |
| *valves:Without Valves* | 241 | - |
| *true flutes* | 384 | 1986 |

Table 6.1: The isolated and solo music datasets that have been used in the experiments according to instrument categories and instrument attributes.

## 6.3 Content-based Audio Analysis

Content-based audio analysis aims to process information from audio files according to a classification system and identify musical instruments and their properties accurately. The system identifies the pre-designated musical terms and the hierarchy is deliberately omitted from the input. For the content-based analysis, however, there are three important issues to consider:

- the label inaccuracies in the social-data-based data sets may affect the results for both content-based and conceptual analysis techniques.
- the lack of actual audio files may reduce the flexibility to extract different type of audio feature sets for the system.
- the adequate representation of concepts and properties may affect the results of domain coverage for the generated ontology.

In previous experiments [Kolozali *et al.*, 2012] we have encountered some of these difficulties. Therefore, the quality of data set is crucial to our system since the utilisation of content-based analysis to obtain the contextual structure of a dataset is a very challenging task on its own. To deal with these issues, we have grounded our instrument identification experiments in line with Chétry [2006], due to the fact that his study indicated that using corresponding dataset and audio features leads to reasonably good results in the context of instrument identification. Content-based audio analysis involves two stages: feature extraction and classification.

### 6.3.1 Feature extraction and clustering

The feature extraction component is based on the Short Time Fourier Transform (STFT) time frequency representation of audio signals. The recordings were sampled at 44100 Hz and short-term audio features were considered on successive frames of 1024 samples, weighted by a Hamming window [Chétry *et al.*, 2005]. On account of the fact that the spectral envelope provides a good representation of the spectrum, and the timbre models used in this study rely on features modelling the spectral envelope which are obtained either from linear prediction (LP) or from Mel-Frequency Cepstral Coefficients (MFCCs). An illustration of the feature extraction steps is depicted in Figure 6.2.

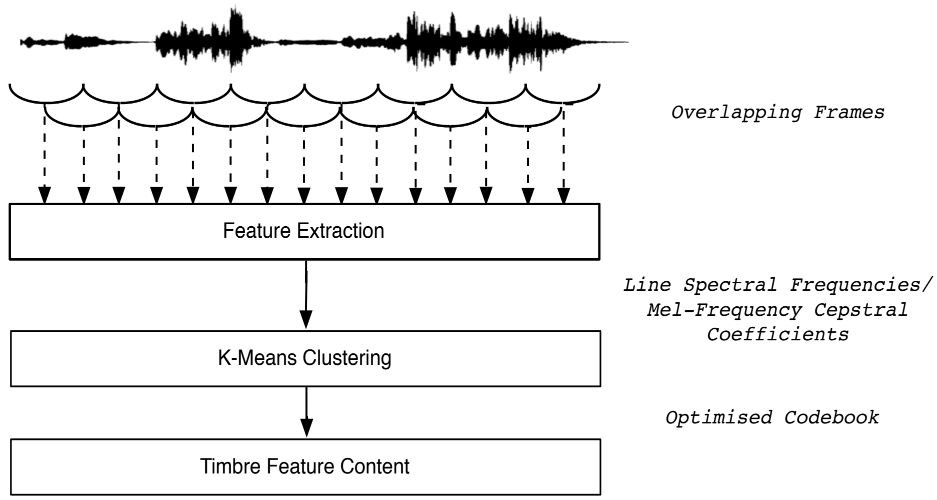



Figure 6.2: Illustration of an audio waveform being represented using overlapping frames, where feature extraction indicates the process of obtaining Line Spectral Frequencies or Mel-Frequency Cepstral Coefficients, and K-Means Clustering indicates the process of obtaining optimised codebooks as a timbre representation of the audio waveform based on the LSFs/MFCCs.

In order to classify various music performances or isolated notes, we extract Mel-Frequency Cepstral Coefficients and Line Spectral Frequencies (LSFs) over overlapping audio frames using the method proposed in [Chétry *et al.*, 2005] and [Barthet & Sandler, 2010b]. The timbre of each instrument is then characterised by a collection of MFCC and LSF feature vectors. In order to determine the best feature vector (codebook) dimensions with regard to performance, a different number of feature coefficients (8, 16, 24 and 32) and number of clusters (K-means) were tested (8, 16, 32, and 64) [Linde *et al.*, 1980]. In total 16 different codebook dimensions were tested for each spectral feature set (LSFs and MFCCs). The details are given in the statistical analysis section.

### 6.3.2 Classification

The classification was performed using both a MLP neural network and SVM which are supervised learning algorithms. Our goal is to associate audio signals related to instrument and attributes. An illustration of the classification task is depicted in Figure 6.3.

The input features are the average and variance of the feature vectors obtained from K-means clustering of LSFs and MFCCs. In the training and testing stages, we use a multi-network system (one network for each instrument concept or property) consisting of 27 networks for the isolated notes and 21 networks for the solo music dataset in both the case of the MLP and SVM classifiers. For this experiment, we used the default Matlab Neural Network Toolbox and the SVM Toolbox which was published in [Canu *et al.*, 2005].

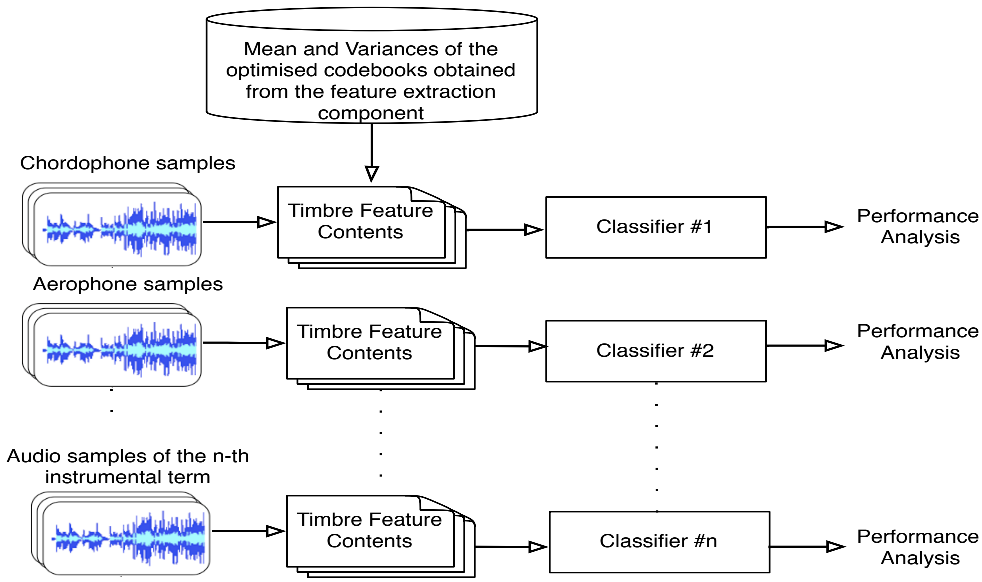



Figure 6.3: An illustration of the classification process in which there is only one classifier for each instrumental concept to predict the output. Timbre feature contents refers to the average (i.e. mean) and variance of the optimal codebooks obtained from the feature extraction component for each instrument concept. The classifiers, which are denoted as rectangles, refer to the process of supervised classification for SVM or MLP classifiers. Classification is based on the inputs obtained by taking the average (i.e. mean) and variance of the optimised codebooks, which represent the audio waveforms based on the LFSs/MFCCs.

*Multi-Layer Perceptron*

The Multi-Layer Perceptron is among the most common types of neural network. Its computing power results from its parallel and distributed structure, and its ability to learn. Our MLP networks contain two hidden layers, with 10 neurons in each hidden layer and an output layer with one neuron. The activation function for each neuron in the hidden layer is a tan-sigmoid function, and a linear transfer function was selected for the output (*purelin* function in Matlab). The MLP was trained using the Levenberg-Marquardt (LM) [Hagan & Menhaj, 1994] back-propagation algorithm. For this experiment, the number of iterations was set to 1000 and the parameters for the learning rate and momentum of the MLP were 0.3 and 0.6 respectively.

*Support Vector Machines*

Support Vector Machines (SVM) have been widely used as an alternative to Neural Networks in modern machine learning. Their basic principle is to discover the best hyperplane leading to the largest distance separation between classes. This formulation embodies the Structural Risk Minimization (SRM) principle used to maximize the margin of separation. SVM algorithms determine the data points in each class that lie closest to the margin (decision boundary), which are called support vectors. Intuitively, a good separation is achieved by a margin that has the largest distance from the support vectors [Bishop, 2006].

There are a number of different kernels that can be used in SVM models. These include linear, polynomial, radial basis function (RBF), and sigmoid. In our experiments, we focused on the polynomial kernel functions of various degrees. We only tested one type of kernel in the present study since the use of different SVM kernels has only resulted in small accuracy differences (2-3%) in previous musical instrument classification studies [Essid *et al.*, 2004, 2006]. For this experiment, the lambda kernel parameter ($\lambda$) was set to $1/I$ as described in [Chétry, 2006], where $I$ is the number of instruments in the database. The degree of the polynomial in kernel function was set to two different values, 2 and 3, to test which performed best. Due to the satisfactory results obtained with a polynomial kernel of degree 3, higher degrees were not tested.

## 6.4 Conceptual Analysis

The conceptual analysis comprises two stages: FCA and lattice pruning. FCA is performed using the Colibri-Java library [Gotzmann & Lindig] in order to generate a hierarchical structure using the outputs of the classifiers. To determine the binary associations between instruments and attributes two criteria need to be verified: $(i)$ a candidate relationship is determined as follows:

$$rel = max(Prec(att|inst)\&Prec(inst|att)) \tag{6.1}$$

where $att$ is the attribute, $inst$ is the instrument, and the $Prec$ is Precision. $(ii)$ the binary association criteria is given by:

$$max(Prec(att|inst)\&Prec(inst|att)) > 0.5 \tag{6.2}$$

The obtained relationships are used to create the formal context which represents the relationships between instrument concepts and properties, to generate a graphical representation of concepts in a lattice form. Finally, in the lattice pruning stage, empty concepts are eliminated and the hierarchical form is revised in order to generate the OWL output of the system.

### 6.4.1 Formal Concept Analysis

As mentioned previously, the aim of the instrument identification experiments was to find the associations between musical instruments and their properties, in order to automatically generate a Semantic Web ontology for musical instruments. Therefore, the outputs of the best musical instrument recognition systems for each dataset were used to obtain the associations between instrumental attributes. The overall performance of the musical instrument recognition system was evaluated by computing the average and standard deviation of the system's precision across instruments. A binarisation process was applied to the obtained results and each network experiment was run for 4 different training/testing sets (cross-validation technique), to prevent biased results. Table 6.2 shows the results for isolated notes and solo music where the mean values are reported along with the standard deviation in brackets. The precision's mean and standard deviation for the correct concept/property associations are shown in bold.

For example in our formal context, high instrument concept/property association Precisions were obtained for most attributes in isolated notes: e.g., the precision for the *Saxophone/single reed instruments* was 0.98, and the mean score for *Cello/bowed* was 0.96. Lower scores were obtained for the association *Trombone/brass instrument* (0.92, on average) and *Flute/edge instrument* (0.89, on average).

Slightly lower association scores were obtained for the solo music dataset than for the isolated notes: the mean score for *Violin/bowed* was 0.87, and the mean score for *Oboe/exciterIs:Air* was 0.96. Lower scores were obtained for the association *Flute/true flutes* (0.88, on average) and *Piano/struck* (0.88 on average). The lowest score was obtained for the edge instruments (flute). The highest recognition scores across all attributes were obtained for *valves:With Valves* (1.00, on average), and *exciterIs:air* (0.98, on average).

In order to generate a binary context for FCA, a threshold of 0.5 was used to determine whether an instrument possessed an attribute or not, as given in Eq. (6.1-6.2).

The results obtained with SVM (3rd degree polynomial kernel) are satisfactory (on both datasets, solo music and isolated notes) for the purpose of formal context generation, since all the associations have been correctly found between instrument attributes and concepts (no errors were made after binarisation). The formal context obtained after binarisation of the results of the isolated notes can be seen in Table 6.3. The identified formal context was used as an input to the FCA algorithm. The formal concepts are extracted by applying FCA to the context generated by the instrument recognition system. Figure 6.4 shows the extracted formal concepts together with the graphical representation of the corresponding concepts in a concept lattice form using a line diagram. Each

| Instrument Categories \ Instrument Attributes | vibrating string | sound initiation process:Bowed | sound initiation process:Struck | vibrating air | edge | lip vibrated | reeds | reeds no:1 | reeds no:2 | valves:With Valves | valves:Without Valves | true flutes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Chordophones | **0.94 (0.01)** | **0.94 (0.02)** | **0.98 (0.02)** | 0.06 (0.01) | 0.06 (0.04) | 0.03 (0.02) | 0.03 (0.01) | 0.02 (0.01) | 0.01 (0.01) | 0 (0) | 0.07 (0.03) | 0.05 (0.04) |
| Aerophone | 0.05 (0.01) | 0.06 (0.02) | 0.02 (0.02) | **0.96 (0.01)** | **0.94 (0.04)** | **0.97 (0.02)** | **0.97 (0.01)** | **0.98 (0.01)** | **0.99 (0.01)** | **1.00 (0.00)** | **0.93 (0.03)** | **0.95 (0.04)** |
| Reep Pipe Instruments | 0.03 (0.01) | 0.03 (0.01) | 0.01 (0) | **0.97 (0.01)** | 0.02 (0.01) | 0.02 (0.01) | **0.93 (0.01)** | **0.97 (0.01)** | **0.93 (0.03)** | 0.03 (0.03) | 0.10 (0.02) | 0.08 (0.04) |
| Brass Instruments | 0.03 (0.02) | 0.01 (0.01) | 0.02 (0.01) | **0.97 (0.02)** | 0.01 (0.01) | **0.91 (0.03)** | 0.05 (0.03) | 0.04 (0.03) | 0.02 (0.01) | **0.97 (0.03)** | **0.83 (0.02)** | 0 (0) |
| Edge Instruments | 0.05 (0.01) | 0.06 (0.02) | 0 (0) | **0.94 (0.04)** | **0.91 (0.05)** | 0 (0) | 0.03 (0.03) | 0.03 (0.01) | 0.01 (0.02) | 0 (0) | 0 (0) | **0.95 (0.04)** |
| Bassoon | 0.04 (0.03) | 0.04 (0.03) | 0.00 (0.00) | **0.96 (0.03)** | 0.01 (0.02) | 0.01 (0.01) | **0.94 (0.03)** | 0.02 (0.03) | **0.92 (0.04)** | 0 (0) | 0.01 (0.02) | 0.01 (0.01) |
| Cello | **0.97 (0.03)** | **0.96 (0.02)** | 0.01 (0.01) | 0.03 (0.03) | 0.02 (0.02) | 0.01 (0.01) | 0 (0) | 0 (0) | 0.01 (0.01) | 0 (0) | 0 (0) | 0.02 (0.02) |
| Clarinet | 0.01 (0.01) | 0.01 (0.01) | 0 (0) | **0.99 (0.01)** | 0.01 (0.01) | 0 (0) | **0.98 (0.01)** | **0.98 (0.01)** | 0.01 (0.01) | 0 (0) | 0 (0) | 0.01 (0.01) |
| Flute | 0.07 (0.06) | 0.07 (0.06) | 0 (0) | **0.93 (0.06)** | **0.89 (0.07)** | 0 (0) | 0.04 (0.02) | 0.04 (0.01) | 0.01 (0.01) | 0 (0) | 0 (0) | **0.89 (0.07)** |
| Oboe | 0.04 (0.03) | 0.04 (0.03) | 0 (0) | **0.96 (0.03)** | 0.01 (0.01) | 0.01 (0.02) | **0.94 (0.05)** | 0.04 (0.04) | **0.90 (0.02)** | 0 (0) | 0.01 (0.02) | 0.01 (0.01) |
| Piano | **0.98 (0.02)** | 0.01 (0.01) | **0.97 (0.02)** | 0.02 (0.02) | 0 (0.01) | 0 (0.01) | 0.01 (0.01) | 0.01 (0.01) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Saxophone | 0.01 (0.01) | 0.01 (0.01) | 0 (0) | **0.99 (0.01)** | 0 (0) | 0.01 (0.01) | **0.99 (0.02)** | **0.98 (0.03)** | 0.01 (0.01) | 0.01 (0.01) | 0 (0) | 0 (0) |
| Violin | **0.93 (0.03)** | **0.93 (0.02)** | 0 (0) | 0.06 (0.03) | 0.01 (0) | 0 (0.01) | 0.01 (0) | 0.04 (0.02) | 0.02 (0.02) | 0 (0) | 0 (0) | 0 (0) |
| Trombone | 0.04 (0.03) | 0 (0) | 0.03 (0.03) | **0.97 (0.03)** | 0 (0) | **0.92 (0.05)** | 0.05 (0.03) | 0 (0) | 0.05 (0.03) | 0 (0) | **0.92 (0.05)** | 0 (0) |
| Tuba | 0 (0) | 0 (0) | 0 (0) | **1.00 (0)** | 0 (0) | **0.98 (0.03)** | 0.02 (0.03) | 0 (0) | 0.02 (0.04) | **0.98 (0.03)** | 0.01 (0.02) | 0 (0) |

(a) SVM (3rd. degree polynomial kernel) – 32 LSFs – 64 codevectors – solo music

| Instrument Categories \ Instrument Attributes | vibrating string | sound initiation process:Bowed | sound initiation process:Struck | vibrating air | edge | reeds | reeds no:1 | reeds no:2 | true flutes |
|---|---|---|---|---|---|---|---|---|---|
| Chordophone | **0.81 (0.02)** | **0.88 (0.02)** | **0.90 (0.07)** | 0.05 (0.01) | 0.04 (0.05) | 0.06 (0.01) | 0.06 (0.02) | 0.03 (0.03) | 0.04 (0.01) |
| Aerophone | 0.19 (0.02) | 0.12 (0.02) | 0.11 (0.07) | **0.95 (0.01)** | **0.96 (0.01)** | **0.94 (0.01)** | **0.94 (0.02)** | **0.97 (0.03)** | **0.96 (0.01)** |
| Reed Pipe Instruments | 0.06 (0.01) | 0.04 (0.01) | 0.01 (0.02) | **0.94 (0.01)** | 0.01 (0) | **0.93 (0.01)** | **0.94 (0.02)** | **0.94 (0.04)** | 0.06 (0.04) |
| Edge Instruments | 0.04 (0.01) | 0.01 (0.02) | 0.02 (0.02) | **0.96 (0.01)** | **0.89 (0.06)** | 0.08 (0.06) | 0.05 (0.05) | 0.02 (0.01) | **0.89 (0.06)** |
| Bassoon | 0.01 (0.01) | 0 (0) | 0.01 (0.01) | **0.99 (0.01)** | 0 (0) | **0.99 (0.01)** | 0.02 (0.02) | **0.97 (0.02)** | 0 (0) |
| Cello | **0.83 (0.03)** | **0.81 (0.02)** | 0.02 (0.03) | 0.17 (0.03) | 0.03 (0.04) | 0.14 (0.06) | 0.06 (0.05) | 0.08 (0.04) | 0.03 (0.04) |
| Clarinet | 0.02 (0.03) | 0.02 (0.03) | 0 (0) | **0.98 (0.03)** | 0.01 (0.02) | **0.97 (0.03)** | **0.93 (0.07)** | 0.04 (0.04) | 0 .01 (0.01) |
| Flute | 0.05 (0.04) | 0.02 (0.01) | 0.03 (0.03) | **0.95 (0.04)** | **0.88 (0.07)** | 0.07 (0.04) | 0.06 (0.04) | 0.02 (0.01) | **0.88 (0.07)** |
| Oboe | 0.04 (0.04) | 0.02 (0.03) | 0.03 (0.02) | **0.96 (0.04)** | 0.04 (0.04) | **0.92 (0.04)** | 0.08 (0.02) | **0.84 (0.03)** | 0.04 (0.04) |
| Piano | **0.89 (0.06)** | 0.01 (0.02) | **0.88 (0.05)** | 0.11 (0.06) | 0.06 (0.07) | 0.05 (0.04) | 0.04 (0.03) | 0.01 (0.02) | 0.06 (0.07) |
| Saxophone | 0.06 (0.04) | 0.06 (0.04) | 0 (0) | **0.94 (0.04)** | 0.01 (0.01) | **0.93 (0.03)** | **0.92 (0.03)** | 0.01 (0.01) | 0.01 (0.01) |
| Violin | **0.85 (0.06)** | **0.87 (0.05)** | 0 (0) | 0.14 (0.05) | 0.02 (0.03) | 0.12 (0.06) | 0.10 (0.05) | 0.02 (0.03) | 0.02 (0.03) |

(b) SVM (3rd. degree polynomial kernel) – 32 LSFs – 64 codevectors – isolated notes

Table 6.2: Confusion matrices for musical instrument concept property associations in the case of isolated notes and solo music

| | vibrating string | sound initiation process:Bowed | sound initiation process:Struck | vibragin air | edge | lip vibrated | reeds | reeds no:1 | reeds no:2 | valves:With Valves | valves:Without Valves | true flutes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Chord. | ✓ | ✓ | ✓ | | | | | | | | | |
| Aero. | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Edge Inst | | | | ✓ | ✓ | | | | | | | ✓ |
| Brass Inst | | | | ✓ | | ✓ | | | | ✓ | ✓ | |
| Reed P Inst | | | | ✓ | | | ✓ | ✓ | ✓ | | | |
| Bassoon | | | | ✓ | | | ✓ | | ✓ | | | |
| Cello | ✓ | ✓ | | | | | | | | | | |
| Clarinet | | | | ✓ | | | ✓ | ✓ | | | | |
| Flute | | | | ✓ | ✓ | | | | | | | ✓ |
| Oboe | | | | ✓ | | | ✓ | | ✓ | | | |
| Piano | ✓ | | ✓ | | | | | | | | | |
| Saxophone | | | | ✓ | | | ✓ | ✓ | | | | |
| Violin | ✓ | ✓ | | | | | | | | | | |
| Trombone | | | | ✓ | | ✓ | | | | | ✓ | |
| Tuba | | | | ✓ | | ✓ | | | | ✓ | | |

Table 6.3: Formal context obtained after binarisation of the results for SVM with the 3rd degree polynomial kernel using 32 LSF features and 64 codevectors for isolated notes.

concept is a cluster of instrument concepts and properties shared by the concepts. The concept lattice is constructed to interpret the subconcept and superconcept relationships between concepts. It consists of 17 formal concepts which are represented by the 17 grey rectangles in the diagram. The labels of the rectangles represent the extent (E) and intent (I) of each formal concept node. By following the ascending paths and the rectangles connected by edges, the diagram shows the concepts and their subconcepts.

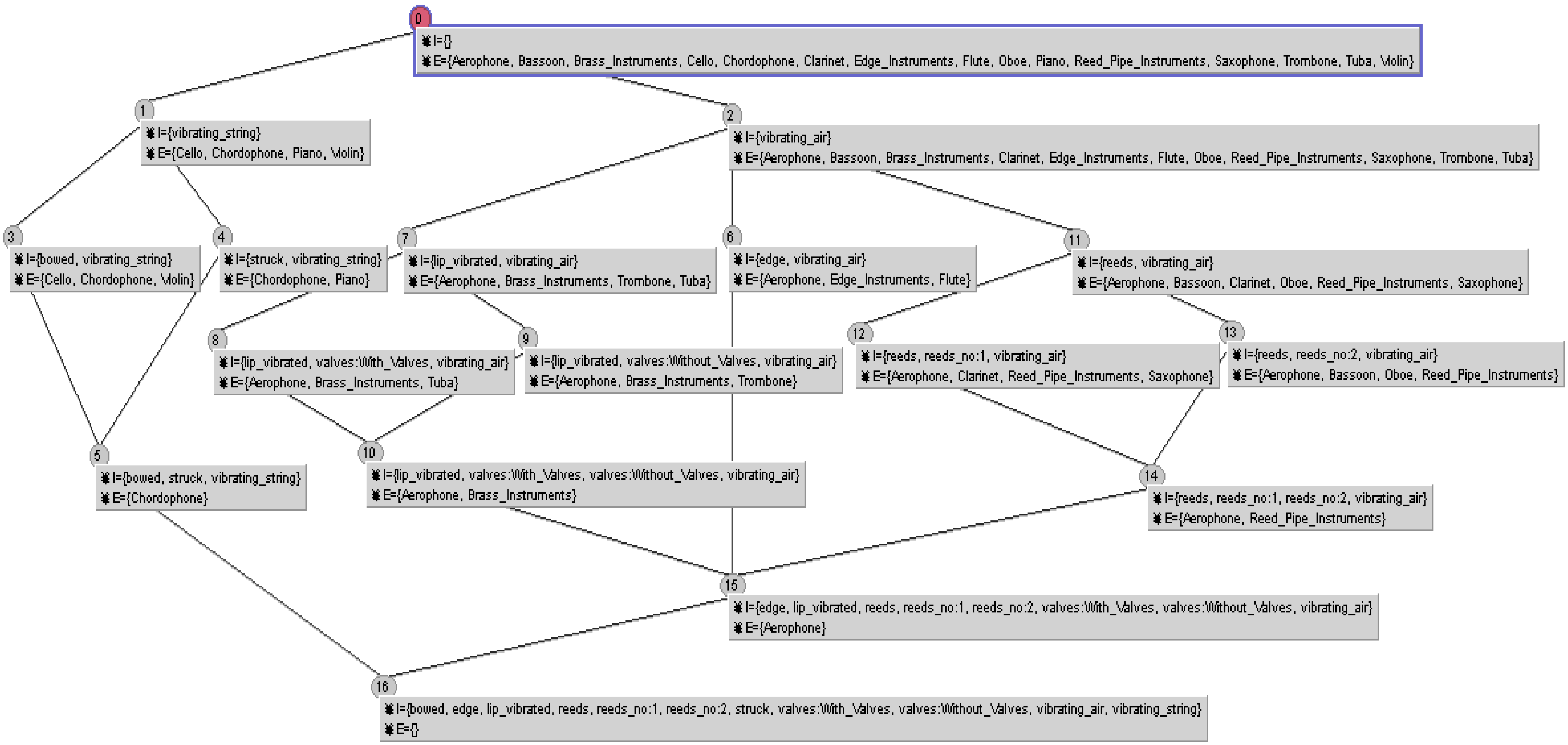



Figure 6.4: The concept lattice of the context in table 6.3. Numbers refer to its conceptual order. This graph is produced by using a software called FCA Stone.

### 6.4.2 Lattice Pruning

As we previously described in Section 5.3.4, in the lattice pruning component, we employ the pruning strategy to eliminate unnecessary repetitions, turning our lattice form into a clean structure that is human-readable and can be converted into an OWL document. This process aims the removal of "empty or unnecessary repetitions" of concepts, objects or attributes based on any of the necessity and stability notions that are defined by knowledge engineers. We have applied this process to the concept lattice of the extracted concepts — for the results of SVM with the 3rd degree polynomial kernel using 32 LSF features and 64 code vectors — given for isolated notes in Figure 6.4 . We again refer to each concept with its numbers, such as $C1$ for concept 1. The concept lattice labels are reduced successfully into 12 formal concepts after removing the empty and non empty infimum concepts, such as $C5$, $C10$, $C14$, $C15$ and $C16$.

Another critical factor is the interpretation of the partial order of the lattice in terms of the OWL representation of the concepts. To avoid information loss, the pruning process was run in parallel with the OWL representation task, so we have used the information to create OWL Class/Properties together with the hierarchical structure before removing the attributes from the concept lattice. We were thereby able to obtain a satisfactory interpretation of the concept lattice.

### 6.4.3 Many-Valued Context

FCA is a successful technique for analysing data that fits well into the structure of a many-valued context. It is an important component of the system that allows to represent the relational data in the formal context. This applies both to relational databases, and else to knowledge bases in the sense of Conceptual Graphs or RDF. The important point when using the many-valued context is identification of complex data: that is, the user has to be able to select the most appropriate context for the data entry to be displayed and understood easily. For example, Toscana[7] toolkit solves this

[7] http://tockit.sourceforge.net/toscanaj/

problem by predefining a number of context scales to be used.

In order to overcome this issue in the context of Formal Concept Analysis, before the experiment in the tagging stage we identify the type of attribute in a many-valued context, and during the experiment we use a conceptual scaling technique, the dichotomous scale, to transform a many-valued context into a one-valued context. The dichotomous scale which defines the negations among attributes has been used by Ganter [Ganter, 2006; Ganter & Wille., 1989].

We applied this method on instrumental many-valued attributes, such as *sound initiation process, reeds no* and *valves*, in order to understand if the property is an ‘object property’ or a ‘data property’. Thus, these attributes have been replaced by scale attributes: e.g., *sound initiation process:Bowed, sound initiation process:Struck, reeds no:1, reeds no:2, valves:With Valves* and *valves:Without Valves*. A many-valued context example is given in Table 6.4 illustrating the transformation of the many-valued context into a one-valued context.

| | sound initiation process | reeds no | valves |
|---|---|---|---|
| Cello | Bowed | - | - |
| Piano | Struck | - | - |
| Clarinet | - | 1 | - |
| Bassoon | - | 2 | - |
| Trombone | - | - | With Vales |
| Tuba | - | - | Without Valves |

(a) Many-valued context

| | sound initiation process:Bowed | sound initiation process:Struck | reeds no:1 | reeds no:2 | valves:With Valves | valves:Without Valves |
|---|---|---|---|---|---|---|
| Cello | ✓ | - | - | - | - | - |
| Piano | | ✓ | - | - | - | - |
| Clarinet | - | - | ✓ | - | - | - |
| Bassoon | - | - | - | ✓ | - | - |
| Trombone | - | - | - | - | ✓ | - |
| Tuba | - | - | - | - | - | ✓ |

(b) Derived one-valued context

Table 6.4: A naive scaling and cross table of the formal context given in Table 6.3

### 6.4.4 Converting Conceptual Hierarchies into a Domain Ontology

The representation of instrument classes and attributes in OWL is another important task in ontology generation. There are two core concepts used in the experiments: *Chordophone* and *Aerophone*. As mentioned above, there are 12 formal concepts containing 12 relations, of which 9 relations between *Aerophone* instruments (i.e. *vibrating air, edge, lip vibrated, reeds, reeds no:1, reeds no:2, valves:Without Valves, valves:Without Valves, true flues*) and 3 relations between *Chordophones* (i.e. *vibration string, sound initiation process: Bowed, sound initiation process: Struck*)

During the process of transforming the concept lattice into a domain ontology, at first step the formal concepts are transformed into ontology classes. A class is the most basic concept in an ontology and every ontology class is implicitly a subclass of *owl:Thing*. The fundamental taxonomic constructor for classes is `rdfs:subClassOf`, which relates a more specific class to a more generic class. For instance, if *Tuba* is a subclass of *Brass Instruments*, then every instance of *Tuba* is also an instance of *Brass Instruments*. The `rdfs:subClassOf` relation is transitive, therefore, if *Tuba* is a subclass of *Brass Instruments* and *Brass Instruments* is a subclass of *Aerophone*, then *Tuba* is a subclass of *Aerophone*.

Secondly, the system goes across the hierarchical level of concept lattice. Thus, root classes are parsed first, then their subclasses, and so on. The OWL class is created for every class in concept lattice with one-to-one relations between classes and their subclasses. It also makes sure that when a subclass is being created, its parent class in hierarchy has already been created. All the extent (E) labels are considered to be potential OWL Class after the lattice pruning stage. An example is given for generated OWL Class sample in Listing 6.1

```
:Aerophones rdf:type owl:Class .
:Chordophones rdf:type owl:Class .
```

Listing 6.1: OWL Class sample.

We need to explicitly define the type of attributes we use as an input. In terms of OWL representation of conceptual relationships, basically all intent (I) labels are considered as potential ontological relationships. However, in OWL, there are two kind of properties: object properties – which relates objects to objects – and data properties – which relates objects to data type values. Since the basic data type of Formal Concept Analysis is that of a binary formal context (single-valued), it is difficult to explicitly define whether an attribute is an object or a data property. Therefore, we relate this issue to the "single-valued and many-valued" representation of data in database design. In the real world, as we mentioned in 5.3.3, attributes are not only a property that an object either possess or not; it may have values such as "reeds no" attribute may have various values (i.e. 1, 2, 3). Therefore, we used conceptual scaling and assumed many-valued concepts as "data-properties" – property that can have data value as object, and single-valued concepts as "object property" – property that can have resource as object. Since we used a punctuation, "colon" to represent the single-valued context which are transformed from many-valued context, that is, our system detects the lexical terms that contains a "colon" to identify the "object" or "data properties". The system, therefore, automatically detects the data properties with their corresponding data, as these are easily identifiable (see right hand side of the table 6.4). Thus, once formal concepts are mapped to OWL classes, relationships used in concept lattice transformed into object-properties or data-properties. Extraction of the data properties (i.e. *reeds_no*) and object properties (i.e. *vibrating_air*) are based on the manual entries. It is worth noting that the extraction of the data properties and the object properties is a new idea which needs to be supported using natural language processing techniques based on a text-corpus data set. Listing C.2 depicts an OWL sample from the automatically generated ontologies for the *Aeropehones*, *Reed Pipe Instruments*, *Bassoon* concepts as well as their object and data properties.

---

```
:reeds rdf:type owl:ObjectProperty .
:vibrating_air rdf:type owl:ObjectProperty .
```

```
:reeds_no rdf:type owl:DatatypeProperty .
:Aerophones rdf:type owl:Class .
:Reed_Pipe_Instruments rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                   owl:onProperty :vibrating_air ;
                   owl:someValuesFrom :Aerophones
                  ] .
:Basoon rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds ;
                    owl:someValuesFrom :Reed_Pipe_Instruments
                  ] ,
                  [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds_no ;
                    owl:hasValue ''2''
                  ] .
```

Listing 6.2: Sample of generated musical instrument ontology.

When we define a property, there is a number of ways to restrict the relation: the domain and range can be specified; the property can be defined to be specialisation (i.e. sub-property) of an existing property, and so on. Finally, the system parses nodes having graphical relation on the concept lattice, and subsequently it parses their properties, and so on. Depending on the graphical relations of the class property, thus, one-to-one or one-to-many relations between classes are created. The class hierarchy of the instrument ontology can be transformed to the ontology web language (OWL) using the OWL API Java library [Horridge & Bechhofer, 2010]. More details regarding the generated OWL files can be found online [8].

## 6.5 Summary

In this chapter, we presented an automatic ontology generation system based on semantic audio analysis. We have given details for the proposed automatic ontology generation system and the database that has been used in the experiments. The system has been divided into two main experimental parts: *i*) content-based audio analysis, and *ii*) conceptual analysis.

The system was tested using MLP and SVM classifiers by modelling different codebook dimensions of the timbre features, LSFs and MFCCs on various instruments for wind and string families in order to find the best performance. We have provided the parameters of the feature extraction and audio classification components of the content-based analysis part of the system.

We have exemplified the construction of the conceptual hierarchy based on FCA for the isolated notes, since the isolated notes data set has a slightly larger terminology than the solo music data set. The OWL specifications, which have been used in converting conceptual hierarchies into a domain ontology, have also been explained. Most notably, to the author's knowledge, this is the first study to investigate automatic ontology generation in the context of audio and music analysis. The next chapter will be providing results and evaluations for both the content based analysis and the conceptual analysis experiments of the system.

[8] http://www.isophonics.net/content/automatic-ontology-generation-based-semantic-audio-analysis

# Chapter 7

# Evaluation of the Ontology Generation System

---

This chapter evaluates the work contained in the previous chapters including content-based and conceptual analysis experiments. The automatic ontology generation system is evaluated empirically by experiments on two different datasets. Our experimental evaluation focuses on suggested semantic audio based techniques for concept mapping.

Firstly, we review the evaluation techniques that have been used to date. In particular four different evaluation techniques: *i*) human assessment, *ii*) task-based assessment, *iii*) data-driven assessment, and *iv*) gold-standard based ontology evaluation techniques.

Secondly, we present the evaluation metrics employed in the assessment of our ontology generation system. Within this section, we consider the gold-standard evaluation approach on both a lexical level and a conceptual level. We describe the metrics utilised for calculations at each level.

Thirdly, we will examine and present the content-based analysis results using a four-way of Analysis of Variance (ANOVA) in section 7.3. To examine whether any significant differences occur for the different combinations of codebook dimensions, type of spectral features and classifiers, ANOVA tests were conducted with the classifiers (i.e. MLP, SVM with polynomial 2nd degree, SVM with polynomial 3rd degree), the audio spectral features (i.e. LSFs and MFCCs), and the codebook dimensions ranging from 8 to 32 and 8 to 64, as independent variables. The dependent variable was the F-Measure.

Finally, we will examine and present the conceptual analysis results using a three-way Multivariate Analysis of Variance (MANOVA) in Section 7.4. The independent variables for the MANOVA test were the classifiers (i.e. MLP, SVM with 2nd degree polynomial, SVM with 3rd degree polynomial), and the audio spectral features (i.e. LSFs and MFCCs). The dependent variables were the evaluation metrics that we present in section 7.2: the Lexical F-measure, the higher level F-measure (TF'), the taxonomic F-measure (TF) for common semantic cotopy, the higher level F-measure (TF') and the taxonomic F-measure (TF) for the semantic cotopy metric. The chapter finishes with a summary of the experimental results.

## 7.1 Ontology Evaluation Techniques

An essential element that makes a specific discipline or approach scientific is the capability to assess and compare outcomes in the area. Evaluations are firmly based on the requirements of the Semantic Web, when coping with abstractions available as ontologies. The Semantic Web possesses characteristics such as high interconnectivity, constant change and incompleteness, and it is possible to develop numerous ontologies conceptualising the same body of knowledge in this web-like structure. Thus, evaluations allow one to ensure that the resulting ontology meets some criteria.

The necessity of an ontology metric to assess ontologies and track their evolution has been revealed by the widespread interest in development and usage of ontologies. A significant number of evaluation methods have been widely studied, such as human assessment, task-based assessment, data-driven assessment, and gold-standard evaluation techniques. Several aspects of the current state of ontology evaluation will be discussed in this section.

### 7.1.1 Human Assessment

In this approach, ontologies are evaluated by considering the ratings of a group of experts according to a certain criteria. Human assessment depending on a set of criteria has been used in many studies. There have been many proposals regarding evaluations and quality measurements determined by schema and instance metrics, design of ontology metrics, and philosophical notions. OntoMetric [Lozano-Tello & Gomez-Perez, 2004], for example, is a tool that allows users to measure the suitability of the existing ontologies in terms of the requirements of their systems. For human assessment, however, probably the most widely utilised methodology is OntoClean methodology [Guarino & Welty, 2000, 2004]. OntoClean is a well-known methodology which formally analyses the concepts' intentional content and their subsumption relationships depending on four philosophical notions dating back to Aristotle. These four notions are: Rigidity (R) , Unity (U) , Dependence (D) and Identity (I).

*Rigidity* is based on the notion of essence. Therefore, a concept is called rigid (**+R**) if and only if it is essential to all of its instances. On the other hand, a non-rigid concept (**-R**) is one that is not essential to some of its instances, and an anti-rigid ($\sim$**R**) concept is one that is not essential to any of its instances.

*Unity* is based on the question "What is part of something and what is not?". It describes the parts and boundaries of objects, such that we know in general what is part of the object, what is not, and under what conditions the object is whole. Concepts carrying a unity are indicated with **+U**, no unity with **-U** , and anti-unity with $\sim$**U**.

*Dependence* describes the dependency of concepts. Thereby, a concept $C_1$ is dependent on another concept $C_2$ when every instance of $C_1$ is an instance of $C_2$. Dependency can be divided into two criteria, such as intrinsic and extrinsic concepts. Intrinsic concepts are independent, whereas extrinsic concepts need to be given to an instance by circumstances or definitions. Concepts carrying an externally dependent property are indicated with **+D**, otherwise with **-D**.

*Identity* refers to the problem of being able to recognise individual entities in the world as being the same or different. Therefore, a concept with identity is one whose instances can be identified as being the same at any time and in any world. When a concept is carrying an identity criterion, it is indicated with **+I**. Otherwise, it is indicated with **-I**.

Several tools and ontology editors have been integrated into OntoClean methodology in order to provide support for cleaning taxonomies and building ontologies, such as ODEClean for WebODE [Fernandez-Lopez & Gomez-Perez, 2002], OntoEdit [Sure *et al.*, 2003] and Protege [Noy *et al.*, 2000]. As long as a taxonomy of concepts are annotated according to philosophical notions, these tools are capable of automatically analysing an ontology and identifying cases of invalid generalisations. Nonetheless, it remains a difficult and time consuming approach due to the fact that these annotations ought to be done manually. In order to solve this issue, Volker *et al.* [2005] proposed an approach for automatically tagging concepts with respect to these criteria.

While this approach may evaluate the quality of the conceptual content of ontologies with a right set of experts, a shortcoming of this evaluation technique is the difficulty of finding the right set of users to perform the task (e.g. ontology engineers, end-users or musicologists). Besides, even though some ontology evaluation methodologies involve experts during the validation process, most of these expert involvements are described rather vaguely.

### 7.1.2 Task-based evaluation

Another evaluation approach is through performing a particular task to determine the performance of an ontology. The main principle is to constitute a comparative evaluation to find the optimum ontology: thus, the more effective ontology is the one that enables the application to acquire good results on the given task. This kind of evaluation paradigm is a lot like the paradigm employed in evaluation activities, such as TREC[1] or MIREX[2] in which diverse frameworks are compared

[1] http://trec.nist.gov/

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME

against each other according to their capability to fulfil a particular task.

Some ontologies are closely connected with their application, so this kind of ontologies cannot be simply exchanged. Such ontologies, so-called "application ontologies", obtain crucial aspects of the user interface: that is, internal data managements and parts of these ontologies may be hard-coded into the application. Normally, the best way to gain access to an application ontology, which is simply an additional component of the employed tool, is via the corresponding application. Hence, the application ought to be evaluated together with the corresponding ontology, so such systems can be considered as integrated systems.

The task-based evaluation approach has also been utilised for core or domain ontologies. For example, in DARPA [Cohen *et al.*, 1998] an evaluation approach is designed for a crisis management scenario where evaluators devised test questions and answer keys similar to an exam. The question and answer keys are rated along with justifications of answers. Porzel & Malaka [2004] describe a scenario where the ontology including its concepts and relations (i.e. `is-a` and semantic relations) are used to discover the closeness of the meaning for the corresponding concepts. The task is a speech recognition problem, where the final output of the task is also compared with a gold standard.

In the Rapid Knowledge Formation Project[3], a similar approach is utilised to evaluate a knowledge framework. In evaluations, domain experts contributed knowledge about DNA transcription derived from ten pages from a standard textbook on ontological systems (i.e. CYC and SHAKEN) [Barker *et al.*, 2004]. Independent experts performed subjective ratings on acquired answers. Another notable aspect of RFK was the employment of complicated explanation questions.

A task-based evaluation ought to reveal the subsequent weak points depending on `is-a` and semantic relations: *i*) insertion errors signifying unnecessary concepts; *ii*) deletion errors indicating absent concepts; *iii*) substitution errors indicating off-target or ambiguous concepts. However, task-

[3] http://www.cs.utexas.edu/users/mfkb/RKF/projects/rkf.html

based evaluation has several drawbacks: *i*) the evaluations are based on a particular method for a particular task, so it is difficult to generalise observations; *ii*) the ontology may be only a small component of the application and its effect on the outcome may be relatively small and indirect; *iii*) evaluating a large number of ontologies is only possible if they can all be plugged into the same application.

### 7.1.3 Data-driven ontology evaluation

An ontology may also be evaluated by comparing its terminology to existing data (i.e. a collection of text) in the application domain. The principle is to assess whether an ontology completely covers the domain of interest. A way to evaluate the completeness of an ontology is to apply natural language processing techniques to discover all relevant terms in a text corpus. Subsequently, the extracted terms are used to assess the domain coverage of the corresponding ontology. For instance, Brewster *et al.* [2004] utilised such an approach for ontology evaluation in which a set of relevant domain-specific terms are extracted from a text corpus applying latent semantic analysis. Next, the obtained terms are compared with the terms appearing in the ontology in order to assess the association between the ontology and the domain specific corpus. But, this approach ignores the relationships between concepts, and is subject to the standard problems with term-matching. Some coverage issues regarding term-matching methods are addressed in [Blaschke *et al.*, 2004], where Gene Ontology[4] has been evaluated by mapping onto other classification and database systems. Also, it is worth noting that entity normalisation is non-trivial in biological domains: for example, authors have pointed out that the length of the names and ambiguity in the vocabulary have yielded poorer results for mouse genes compare to yeast genes.

Raimond [2008] introduced a method combining task-based and data driven techniques. The task was simply to answer a set of musical queries. Likewise other data-driven approaches, it started from a text corpus and mapped the extracted terms on to the proposed knowledge representation

[4] http://www.geneontology.org/

framework. Similarly, Baumann *et al.* [2002] described an evaluation approach for an ontology based music search system, whereby they collected 1500 verbalised queries, and clustered them in several categories in order to analyse them qualitatively.

### 7.1.4 Gold-Standard — Similarity Based Evaluation

The gold-standard evaluation approach builds on the idea of utilising a similarity measure to compare an ontology with an existing ontology that serves as a reference. This approach is particularly useful to evaluate automatically obtained ontologies using a golden-standard ontology as a reference. We have used this approach in our evaluations. A set of measures may be used in order to describe at the lexical and conceptual levels.

For the lexical term layer, binary measures are often chosen over to assess acquired terms. They are typically performed depending on an exact match of strings by applying precision and recall, such as in [Sabou *et al.*, 2005b]. This is known as lexical precision and recall [Sabou *et al.*, 2005a]. A different illustration of a lexical-level evaluation is the String Matching method which is presented in [Maedche & Staab, 2002]. This measure is calculated through the Levenshtein edit distance [Levenshtein, 1966], normalised to [0, 1]. A string-matching measure between two sets of strings is defined by taking each string of the first set, finding its similarity to the most similar string in the second set, and averaging this over all strings of the first set. The second set may be regarded as the terminology of the gold-standard ontology which is considered to be a good representation of the application domain being assessed.

Taxonomic comparison consists of two parts: *i*) a local measure is obtained by comparing the positions of concepts between the acquired ontology and reference ontology; *ii*) the global measure is calculated by computing the average of the concept pairs which are obtained by local measurements. Maedche & Staab [2001] proposed several measures for taxonomic comparison of ontologies, such as semantic cotopy which allows us to assess structural aspects of two ontologies. With a gold-standard ontology, these measures can be used for ontology evaluation. The gold-standard can be another ontology, as in this study, based on a document-corpus or provided by experts. More details will be provided about the similarity measures of ontologies in the next section. Further details on ontology evaluation techniques can be found in [Vrandecic, 2010; Obrst *et al.*, 2007].

## 7.2 Similarity Measures for Ontology Evaluation

Entities need to have common characteristics in order to be considered as similar, and the calculation of the similarity between two items needs to be carried out very carefully to ensure that all the retrieved items are relevant. A few approaches have been developed in order to calculate the similarity between ontologies, but first to formalise the notion of similarity, we refer to the definition of a similarity function of Ehrig *et al.* [2005]:

$$sim : \mathcal{O} \times \mathcal{O} \rightarrow [0,1] \tag{7.1}$$

where *sim* calculates the degree of similarity between two ontologies, $\mathcal{O}$, in order to obtain a value between $[0,1]$. Commonly, these kind of measures are reflexive and symmetric. In an effort to perform an evaluation, we can compare various ontologies utilising similarity functions at two different levels: lexical and taxonomic levels. We will first highlight the precision and recall that constitutes the foundation of the used ontology evaluation metrics; we will then describe the evaluations at the lexical and taxonomic levels through illustrations.

### 7.2.1 Precision and Recall

To assess the performance of ontology generation algorithms, standard evaluation metrics are usually utilised, such as F-measure (F), Recall (R) and Precision (P). Recall is the proportion of relevant material retrieved in answer to a search request, and precision is the proportion of retrieved material that is actually relevant [van Rijsbergen, 1979; Richter, 1992; Euzenat, 2007]. Precision and recall are defined as follows:

$$P(Ref, Comp) = \frac{\mid Comp \cap Ref \mid}{\mid Comp \mid} \tag{7.2}$$

$$R(Comp, Ref) = \frac{\mid Comp \cap Ref \mid}{\mid Ref \mid} \tag{7.3}$$

where $Ref$ is the reference ontology that we assume to be a gold-standard ontology, and $Comp$ is the ontology which is being compared to the reference ontology. Additionally, F-measure is the harmonic mean of recall and precision. In order to achieve a high F-measure score, the classifier must achieve both high precision and high recall where 1 indicates perfect correlation, and 0 indicates no correlation. The equation for the F-measure is given below[5]:

$$F = \frac{2 \times P(Comp, Ref) \times R(Comp, Ref)}{P(Comp, Ref) + R(Comp, Ref)} \tag{7.4}$$

### 7.2.2 Lexical Comparison

Lexical comparison assesses the similarity between lexicons (set of terms denoting concepts) of the automatically generated and gold-standard ontology. The metric is called Lexical Overlap (LO) which is usually evaluated applying precision and recall, which are well-known in information retrieval. In this context, precision is the fraction of the successfully computed lexical entries to the overall computed lexical entries, and recall is the fraction of the successfully computed lexical entries to the gold-standard lexical entries citepZouaq:2007ly. The equations for lexical precision (*LP*) and lexical recall (*LR*) are as follows:

[5]The F-measure calculation does not take the true negative rate into account which allows to measure various coefficients such as the Phi coefficient and Matthews correlation coefficient.

$$LP(\mathcal{O}_C, \mathcal{O}_R) = \frac{|\mathcal{C}_C \cap \mathcal{C}_R|}{|\mathcal{C}_C|} \tag{7.5}$$

$$LR(\mathcal{O}_C, \mathcal{O}_R) = \frac{|\mathcal{C}_C \cap \mathcal{C}_R|}{|\mathcal{C}_R|} \tag{7.6}$$

where $\mathcal{O}_C$ refers to the computed ontology and $\mathcal{O}_R$ refers to the reference ontology. Lexical precision and recall, therefore, reflect how well the computed lexical terms cover the target domain. It is worth noting that these definitions are based on exact match of the labels, so the comparison does not deal with different use of hyphens in multi-word phrases.

### 7.2.3 Taxonomic Comparison

Taxonomic comparison assesses the similarity between the taxonomic structures and the relations between automatically generated and gold-standard ontologies at the conceptual level. We use the evaluation metrics proposed by [Dellschaft & Staab, 2006]. This approach is modified version of one of the most popular approaches in the ontology learning field [Maedche & Staab, 2002]. It applies *Taxonomic Overlap* to find the similarity measure, taking into consideration the taxonomic structures of ontologies. In particular, each concept in a computed taxonomy and a corresponding concept in a reference ontology are compared based on the similarity of their ancestors and descendants, as described in [A. & K., 2009].

The idea is based on two different measures: $i)$ local taxonomic measure, and $ii)$ global taxonomic measure. The local taxonomic measure compares the positions of two concepts, and the global taxonomic measure compares the entire concept hierarchy of the two ontologies. The local taxonomic precision is given by the following equation:

$$tp_{ce}(c_1, c_2, \mathcal{O}_C, \mathcal{O}_R) := \frac{|ce(c, \mathcal{O}_C) \cap ce(c, \mathcal{O}_R)|}{|ce(c_1, \mathcal{O}_C)|} \tag{7.7}$$

where $ce$ is the character extraction that gives the characteristic objects for the position of a concept $c$ in the hierarchies $\mathcal{O}_C$ and $\mathcal{O}_R$. For the taxonomic overlap measure the semantic cotopy, $sc$, and common semantic cotopy, $csc$ are given by:

$$sc(c, \mathcal{O}) := \{c_i | c_i \in \mathcal{C} \wedge (c_i \leq c \vee c \leq c_i)\} \tag{7.8}$$

$$csc(c, \mathcal{O}_1, \mathcal{O}_2) := \{c_i | c_i \in \mathcal{C}_1 \cap \mathcal{C}_2 \wedge (c_i <_{C1} c \vee c <_{C1} c_i)\} \tag{7.9}$$

where $\wedge$ and $\vee$ represent AND and OR logical commands. $c_i \leq c$ returns all the descendants and $c \leq c_i$ returns all the ancestors for the concept $c$ in taxonomy. The corresponding ontology is defined by $O$ whereas the corresponding set of concepts for the taxonomy is defined by $C$. The common semantic cotopy, $csc$, is another taxonomy overlap measure which excludes the corresponding concept from its common semantic cotopy, as well as all the concepts that are not included in the concept set of the other ontology. The set of concepts for the corresponding ontologies ($\mathcal{O}_1$ and $\mathcal{O}_2$) are defined as $C_1$ and $C_2$, respectively. In the optimistic assessment, as in [Maedche & Staab, 2002], the current concept is compared with all the concepts from the reference ontology and the highest precision is chosen by picking the best match of $c$ in $\mathcal{O}_1$. The global taxonomic precision $TP$ and recall $TR$ are defined by the following equations:

$$TP_{sc}(\mathcal{O}_C, \mathcal{O}_R) := \frac{1}{|\mathcal{C}_C|} \sum_{c \in \mathcal{C}_C} \begin{cases} tp_{sc}(c, \mathcal{O}_C, \mathcal{O}_R), & \text{if } c \in \mathcal{C}_R \\ 0, & \text{if } c \notin \mathcal{C}_R \end{cases} \tag{7.10}$$

$$TR_{sc}(\mathcal{O}_C, \mathcal{O}_R) := TP_{sc}(\mathcal{O}_R, \mathcal{O}_C) \tag{7.11}$$

where $TP_{sc}$ represents the local taxonomic precision and $TR_{sc}$ represents the taxonomic recall of the corresponding ontology based on the semantic cotopy, $sc$. The local taxonomic precisions are summed up and averaged over all the taxonomic overlaps for a set of concepts $C_C$ in the corresponding ontology $\mathcal{O}_C$. The common semantic cotopy of the global taxonomic overlap is computed as follows:

$$TP_{csc}(\mathcal{O}_C,\mathcal{O}_R) := \frac{1}{|\mathcal{C}_C \cap \mathcal{C}_R|} \sum_{c \in \mathcal{C}_C \cap \mathcal{C}_R} tp_{csc}(c,\mathcal{O}_C,\mathcal{O}_R) \tag{7.12}$$

$$TR_{csc}(\mathcal{O}_C,\mathcal{O}_R) := TP_{csc}(\mathcal{O}_R,\mathcal{O}_C) \tag{7.13}$$

In the equation (7.12) and (7.13), the local $TP$ and $TR$ are summed up and averaged over all the taxonomic overlaps according to a common set of concepts of the ontologies. Finally, we used the taxonomic F-measure ($TF$) calculating the harmonic average of taxonomic overlap in both $\mathcal{O}_{auto}$ and $\mathcal{O}_{ref}$: the automatically generated and reference ontologies, respectively. The equation for the taxonomic F-measure ($TF$) is given by:

$$TF(\mathcal{O}_C,\mathcal{O}_R) := \frac{2 \cdot TP(\mathcal{O}_R,\mathcal{O}_C) \cdot TR(\mathcal{O}_R,\mathcal{O}_C)}{TP(\mathcal{O}_R,\mathcal{O}_C) + TR(\mathcal{O}_R,\mathcal{O}_C)} \tag{7.14}$$

In addition to the taxonomic F-measure, there is a need for a higher-level metric that involves not only the quality-of-concept hierarchy but also the lexical measure of the ontologies. Therefore, a higher-level F-measure, $TF'$, has been used in conjunction with the lexical measures to evaluate ontologies. The equations for the higher level F-measure, $TF'$, are given by :

$$TF'(\mathcal{O}_C,\mathcal{O}_R) = \frac{2 \cdot LR(\mathcal{O}_C,\mathcal{O}_R) \cdot TF(\mathcal{O}_C,\mathcal{O}_R)}{LR(\mathcal{O}_C,\mathcal{O}_R) + TF(\mathcal{O}_C,\mathcal{O}_R)} \tag{7.15}$$

## 7.3 Statistical Analysis of the Content-based Analysis System

To determine the level of accuracy of the musical instrument recognition system, F-Measures were computed for various combinations of classifiers, audio spectral features, and codebook dimensions (i.e. no. of coefficients and no. of clusters). The F-measure was traditionally obtained based on the precision and recall of the identification system.

The effects of the factors along with interaction effects were analysed from one-way up to four-ways by using the partial eta squared index of effect size. A multiple comparison procedure (MCP) was conducted in order to identify the interaction affects between factors. The Holm-Sidak procedure [Holm, 1989] was used here, as in [Barthet *et al.*, 2010a]. The definitions in [Cohen, 1977] have been adopted to discuss the effect sizes in the following cases: small effect size ($\eta^2 \leq .01$), medium effect size ($.01 \leq \eta^2 \leq .06$) and large effect size ($.06 \leq \eta^2 \leq .14$). ANOVA levels of significance are reported using the F-statistics and probability the $p$. A risk $\alpha$ of .05 was used in all statistical tests.

### 7.3.1 Results of the Content-based Analysis

The performances of six systems using LSFs and MFCCs are reported in Table 7.1. The total average correct identification rates ranges from 67.5% to 87.5% for the solo music dataset, from 38.5% to 90.3% for the isolated notes dataset. For the MLP classifier, the average correct identification rates were 76 % and 46.7% for the solo music and isolated notes, respectively.

Overall, the best results were found by using SVM polynomial at 3rd degree for both datasets: for instance, the average correct identification rate was slightly increased up to 83.0% and 86.3% for the solo music and the isolated notes, respectively. The highest performance for this classifier was obtained with 32 coefficients and 64 codevectors for both of the feature sets, LSF (87.5%) and MFCC (83.1%), on the solo music dataset. For the isolated notes, although the highest accuracy (90.3%) was obtained with the same settings, 32 coefficients and 64 codevectors, for the LSF feature set, the best performance was obtained with 8 coefficients and 64 codevectors (87.7%) for the MFCCs feature set.

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 75.2 | 76.2 | 75.4 | 77.5 | **77.9** | 76.5 | 76.2 | 76.4 |
| 16 | 72.2 | 77.2 | 74.9 | 78.0 | 73.3 | 77.9 | 74.9 | 79.5 |
| 24 | 69.9 | 79.6 | 73.2 | 77.0 | 74.0 | 77.3 | 73.2 | **80.3** |
| 32 | 73.0 | 79.7 | 73.9 | 79.1 | 72.9 | 78.9 | 74.0 | 79.6 |

(a) MLP on the Solo Music Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 43.2 | 45.9 | 38.8 | 43.3 | 37.8 | 45.9 | 39.2 | 47.7 |
| 16 | **48.1** | 46.3 | 43.1 | 52.9 | 42.9 | 52.0 | 47.0 | 48.8 |
| 24 | 46.4 | 51.0 | 42.7 | 48.3 | 42.8 | 49.0 | 46.1 | 47.0 |
| 32 | 43.5 | 52.2 | 46.3 | **56.4** | 41.6 | 54.3 | 42.8 | 53.3 |

(b) MLP on the Isolated Notes Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 82.9 | 69.4 | 83.9 | 69.1 | 84.3 | 67.5 | **84.5** | 68.4 |
| 16 | 81.9 | 76.4 | 82.0 | 77.6 | 83.0 | 77.7 | 83.7 | 76.9 |
| 24 | 80.3 | 81.5 | 82.2 | 81.5 | 82.3 | 81.1 | 82.7 | 81.9 |
| 32 | 80.5 | 82.3 | 82.2 | 81.3 | 82.0 | 81.9 | 83.0 | **82.2** |

(c) SVM w/ 2nd degree Polynomial Kernel on the Solo Music Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 84.5 | 61.1 | 84.9 | 60.7 | 85.7 | 62.0 | 87.7 | 62.1 |
| 16 | 84.7 | 65.4 | 84.8 | 65.7 | 85.7 | 67.7 | 86.3 | 68.3 |
| 24 | 85.0 | 69.4 | 84.9 | 72.7 | 85.1 | 73.3 | 86.3 | **73.7** |
| 32 | 84.5 | 70.0 | 84.2 | 71.6 | 84.7 | 72.7 | **87.7** | **73.7** |

(d) SVM w/ 2nd degree Polynomial Kernel on the Isolated Notes Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 81.0 | 79.1 | 80.6 | 79.7 | 81.7 | 79.3 | 81.4 | 79.1 |
| 16 | 80.2 | 85.1 | 80.0 | 84.9 | 81.4 | 86.8 | 82.4 | 85.8 |
| 24 | 80.3 | 86.9 | 81.4 | 86.6 | 81.5 | 86.7 | 82.2 | 86.8 |
| 32 | 80.0 | 87.4 | 81.6 | 85.7 | 81.6 | 86.5 | **83.1** | **87.5** |

(e) SVM w/ 3rd degree Polynomial Kernel on the Solo Music Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| COEF | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 84.5 | 81.7 | 84.9 | 82.2 | 85.7 | 83.8 | **87.7** | 84.4 |
| 16 | 84.7 | 86.2 | 84.8 | 87.5 | 85.7 | 88.1 | 86.3 | 88.1 |
| 24 | 85.0 | 88.0 | 84.9 | 89.1 | 85.1 | 89.4 | 86.3 | 90.0 |
| 32 | 84.5 | 87.6 | 84.2 | 88.8 | 84.7 | 90.0 | 86.8 | **90.3** |

(f) SVM w/ 3rd degree Polynomial Kernel on the Isolated Notes Dataset

Table 7.1: Performance of the Musical Instrument Recognition Systems for the Isolated Notes and Solo Music Datasets. In each case, the best performance is reported in bold. The ontology outputs of MLP using 16 MFCC features and 8 codevectors corresponds to $\mathcal{O}_{C2}$, and MLP using 32 LSF features and 16 codevectors corresponds to $\mathcal{O}_{C3}$ in Figure 7.2.

### 7.3.2 Comparison of the Classifiers and the Spectral Feature Sets

The results of the four-way analyses of variance for the musical instrument recognition system are reported in Table 7.2. Highly significant effects of the classifiers (*CLSR*) were found for both the solo music and isolated note datasets, $F(2,6415) = 243.135, p < .001$ and $F(2,8352) = 2415.081, p < .001$, respectively. The effect size of the classifier factor was medium size ($\eta^2 = .070$) on the solo music dataset, whereas a very large effect size ($\eta^2 = .366$) was found for the isolated notes.

We also found a significant effect for the audio spectral feature sets (*ASFT*) on both of the datasets, $F(1,6415) = 9.103, p = .003$, for the solo music, and $F(1,8352) = 44.006, p < .001$, for the isolated notes. The effects were of small size for both datasets ($\eta^2 = .001$ and $\eta^2 = .005$), respectively.

The posthoc analyses (multiple comparison procedure) conducted for the solo music and the isolated notes showed that the SVM classifier was significantly better than the MLP classifier ($p <.001$) independently of the polynomial degree and the dataset. The average differences between the MLP classifier and the SVM with polynomial kernels of degree 2 (3.1% $\leq \Delta$ F-measure $\leq$ 4.6%), degree 3 (6.2% $\leq \Delta$ F-measure $\leq$ 7.7%) was significant at the .05 level for the solo music dataset. Although a significant difference occurred for the isolated notes as well, the average F-measure differences between the MLP and SVM-based cases surprisingly increased (28.6% $\leq$ $\Delta$ F-measure $\leq$ 31.4% for degree 2, and 38.1% $\leq \Delta$ F-measure $\leq$ 40.9% for degree 3), respectively. A significant difference between the performance of SVM polynomial kernels of 2nd and 3rd degrees was also found. The SVM with a 3rd order polynomial kernel performed significantly better than the SVM with a 2nd order polynomial kernel for both datasets: 2.3% $\leq \Delta$ F-measure $\leq$ 3.8% for solo music, and 8.1% $\leq \Delta$ F-measure $\leq$ 10.9% for the isolated notes.

The LSF feature sets performed slightly better than the MFCC feature set for the solo music dataset, whereas MFCCs performed slightly better than LSFs on the isolated notes dataset. The average F-measure differences between the LSF and the MFCC feature sets were in the range $0.3\% \leq \Delta$ F-measure $\leq 1.3\%$ for the solo music dataset, $2.3\% \leq \Delta$ F-measure $\leq 4.2\%$ for the isolated notes dataset.

### 7.3.3 Influence of the Codebook Dimensions

The analysis of variance showed that the effect of the number of coefficients (*COEN*) was highly significant for the solo music and isolated notes datasets, $F(3,6415) = 30.628, p < .001$ and $F(3,8352) = 20.237, p < .001$, respectively. The effect sizes were found to be small for both datasets ($\eta^2 = .014$, solo music, and $\eta^2 = .007$, isolated notes). There was also a significant effect of the number of codebook clusters (*CLUN*) related to the K-means algorithm, on both datasets. The effect sizes were small ($\eta^2 = .002$, solo music, and $\eta^2 = .001$, isolated notes). Conversely, there was no significant mean differences in F-measures when the number of clusters was varied. With regard to the codebook dimensions in the case of solo music, the posthoc test revealed that there was no significant differences, except for 8 coefficients, when the number of feature coefficients varied by only 8. However, there was a significant difference ($p = .010$) between experiments where the coefficient number differed from 16 (e.g. between 16 and 32) with a small average difference, $.2\% \leq \Delta$ F-measure $\leq 2.1\%$. Highly significant differences ($p < .001$) were found when the number of coefficient was small (8) compared to other cases (16, 24, 32) for both type of spectral features, with small average differences ($1.1\% \leq \Delta \leq 3.0\%$, $1.8\% \leq \Delta$ F-measure $\leq 3.7\%$, and $2.2\% \leq \Delta F - measure \leq 4.1\%$, respectively).

For isolated notes, the same pattern occured. The average difference between using 16 and 32 feature coefficients was $0\% \leq \Delta$ F-measure $\leq 3.6\%$, and the average differences between using 8 feature coefficients and a higher number of coefficients (i.e. 16, 24 and 32) were $1.3\% \leq \Delta$ F-measure $\leq 4.9\%$, $2.4\% \leq \Delta$ F-measure $\leq 6.1\%$, and $3.1\% \leq \Delta F - measure \leq 6.7\%$, respectively.

| | Solo Music | | | Isolated Notes | | |
|---|---|---|---|---|---|---|
| Source | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ |
| CLSR | 2 | 243.135*** | .070 | 2 | 2415.081*** | .395 |
| ASFT | 1 | 9.103** | .001 | 1 | 44.006*** | .005 |
| COEN | 3 | 30.628*** | .014 | 3 | 20.237*** | .007 |
| CLUN | 3 | 3.488* | .002 | 3 | 2.844** | .001 |
| CLSR×ASFT | 2 | 141.388*** | .042 | 2 | 214.478*** | .049 |
| CLSR×COEN | 6 | 11.197*** | .010 | 6 | 1.578 | .001 |
| CLSR×CLUN | 6 | .143 | .000 | 6 | .337 | .000 |
| ASFT×CLUN | 3 | 2.594 | .001 | 3 | .984 | .000 |
| ASFT×COEN | 3 | 72.736*** | .033 | 3 | 8.948*** | .003 |
| CLUN×COEN | 9 | .370 | .001 | 9 | .463 | .000 |
| CLSR×ASFT×CLUN | 6 | .265 | .000 | 6 | .219 | .000 |
| CLSR×ASFT×COEN | 6 | 7.752*** | .007 | | 3.286** | .002 |
| CLSR×CLUN×COEN | 18 | .240 | .001 | 18 | .559 | .001 |
| ASFT×CLUN×COEN | 9 | .732 | .001 | 9 | .419 | .000 |
| CLSR×ASFT×CLUN ×COEN | 18 | .217 | .001 | 18 | .473 | .001 |
| Error | 6415 | | | 8352 | | |

Table 7.2: Results of the Four-Way Analysis of Variance for the Musical Instrument Recognition System. $\eta^2$ is the partial eta squared measure of effect size. $^{*}p < .05, ^{**}p < .01, ^{***}p < .001$. CSLR: classifier; ASFT: audio spectral feature set; COEF: number of coefficients; CLUN: number of clusters.

### 7.3.4 Relationships Between the Factors

The interaction between the classifier factor, the spectral feature sets, and the dimensions of the feature vector were highly significant for solo music, $F(2,6415) = 141.388, p < .001$ ($CLSR \times ASFT$) and $F(6,6415) = 11.197, p < .001$ ($CLSR \times COEN$), $F(6,6415) = 7.752, p < .001$ ($CLSR \times ASFT \times Coen$), respectively. Although the effect size of the interaction between classifier and spectral feature sets was larger than the interaction between the classifier factor and the coefficient factor, both interaction effects were small ($\eta^2 = .042$, $\eta^2 = .010$). There was also a highly significant interaction between the classifier and the spectral feature set factors, $F(2,8352) = 214.478, p < .001$ for isolated notes, with a small effect size ($\eta^2 = .049$). Nevertheless, there was no significant interaction effect between the classifier and the dimensions of the feature vectors for the isolated notes.

The interaction between spectral features and the number of coefficients yielded an $F$ ratio of $F(3,6415) = 72.736, p < .001$, and $F(3,8352) = 8.948, p < .001$ for solo music and isolated notes respectively, indicating that there were highly significant effects of interaction on both datasets.

The effect sizes of the interaction were very small ($\eta^2 = .033$ and $\eta^2 = .003$) for both datasets.

Finally, the interaction between the classifier, the spectral feature set, and the dimensions of the feature vectors were highly significant on the solo music dataset, $F(6,6415) = 7.752, p < .001$, and significant on the isolated notes dataset, $F(6,8352) = 3.286, p = .003$. The effect sizes were small ($\eta^2 = .007$) for the solo music, and very small ($\eta^2 = .002$) for the isolated notes.

### 7.3.5 Discussion

The results of the experiments show that in most cases, classifying the samples with the SVM classifier increases the performance of the identification for both the isolated notes and the solo music datasets. This can be explained by the characteristics of the SVM classifier such the use of a kernel function to map the data into a higher-dimensional space where the classes become linearly separable as well as the determination of an optimal hyperplane that minimises the probability of misclassifications. Using inherent nonlinear approximation properties provide the capability to model very complex patterns for SVM. This is unlike the traditional MLP which relies on linear basis function architectures with inputs weighted before being summed and have sigmoidal or step activation functions. MLP obtained good identification rate (76.0%) for the majority of the instruments in the solo music dataset, whereas there was a very low performance (46.7%) for the isolated notes dataset. Although reverberation may affect the instrument recognition system [Barthet & Sandler, 2010a], in this case, it may be inferred that the contextual information obtained analysing musical phrases rather than isolated notes substantially improved the identification rate. Thus, there will be reasonably good ontologies with the MLP classifier applied on the solo music dataset, as we will see in the next section. It could be assumed that ontologies that obtained with MLP classifier for the isolated notes dataset will be less than ideal compared to the SVM classifiers, since the identification error will intrinsically be propagated to the conceptual analysis part of the system. This could be attributed to the characteristics of the SVM classifier, as we discussed previously.

Regarding the SVM classifiers, there was also a small difference between the 2nd and 3rd degree polynomial kernels: $2.3\% \leq \Delta$ F-measure $\leq 3.8\%$ for solo music, and $8.1\% \leq \Delta$ F-measure $\leq 10.9\%$. This could be explained by the fact that there was a very low average identification rate in case of SVM polynomial kernels of 2nd degree with LSFs for a few instrument categories: *reeds no:2* (4%), *Edge Instruments* (20%), *Flute* (19%) and *TrueFlutes* (21%), while the average identification rates were low only for the *Clarinet* (46%) and *Cello* (56%) on the solo music dataset. The low scores obtained for the edge and reed pipe instruments may be due to the fact that the source/-filter model of sound production from which the LSFs are derived is not well adapted for these instruments. Indeed, for edge instruments such as the flute and clarinet, the coupling between the exciter (the closed-end-of-the flute) and the resonator (the bore) plays an important role regarding timbre, as for the clarinet [Barthet *et al.*, 2010b]. It is shown that timbral melody and expression can advantageously be taken into account to build more robust instrument models. However, additional information on the datasets (i.e. recording conditions, attacks and decays of the notes) should be required before a complete understanding of the classifiers' performance can be reached.

Even though there was a significant difference between LSFs and MFCCs for the SVM with 2nd degree polynomial on the isolated notes, there was small difference depending on classifier for the rest of the cases. The small difference can be explained by the fact that they botch characterise the spectral envelope of the sounds. This is in line with the results of Chétry [2006], where MFCCs performed better than LSFs with Gaussian Mixture Model (GMM), and vice versa with the codebook-to-codebook similarity measure based on the K-means clustering algorithm.

We found that the highest performance has been shown by the higher number of coefficients (*COEN*) for the datasets This can be attributed to the fact that using a high coefficient number tends to model too much spectral information such as salient partials and overtones, whereas a small number of coefficient numbers cannot capture the detailed information carried by the formant structure. Subsequently, the insignificant effect of the number of clusters (*CLUN*) could be associated with the fact that we used the average and variance of the codebook dimensions. This is likely to affect the accuracy obtained with the codebook dimensions. However, it should be noted that the use of full codebooks would require implementations of the MLP and SVM classifiers as current machine learning toolboxes can't handle multidimensional data structure.

## 7.4 Quantitative Comparison for the Conceptual Analysis

In this section the automatically constructed concept hierarchies will be evaluated analytically using measures presented in section 7.2. The level of accuracy estimated by these metrics were analysed utilising a three-way multivariate analysis of variance. The independent variables were the classifiers (i.e. MLP, SVM with 2nd degree polynomial, SVM with 3rd degree polynomial), the audio spectral features (i.e. LSFs and MFCCs), and the number of coefficients (i.e. 8, 16, 24, 32). The dependent variables were the following evaluation metrics: Lexical Recall (*LR*), higher level F-measure ($TF'_{csc}$), taxonomic F-measure ($TF_{csc}$) of the common semantic cotopy, higher level F-measure ($TF'_{sc}$), taxonomic F-measure ($TF_{sc}$) of the semantic cotopy metric.

Content-based analysis provides an output for each instrument concept or property. We, therefore, obtained a large set of output data in line with the amount of our experimental dataset. An ontology consists of all the corresponding concepts and properties, yet, there is only one output to be analysed. Thus, there was a small set of output data for the conceptual analysis. Considering the fact that the number of clusters factor ($COUN$) did not produce a significant effect in the content-based analysis part, it was not considered as a dependent variable in order to get a sufficient degree of freedom to run MANOVA test.

The effects of the factors along with interactions effects were analysed in range of one-way to three-ways by using the partial eta squared index of effect size. Subsequently, a multiple comparison procedure (MCP) was also conducted for the conceptual analysis results in order to identify whether there was a significant difference between the parameters of the factors. Similar to the

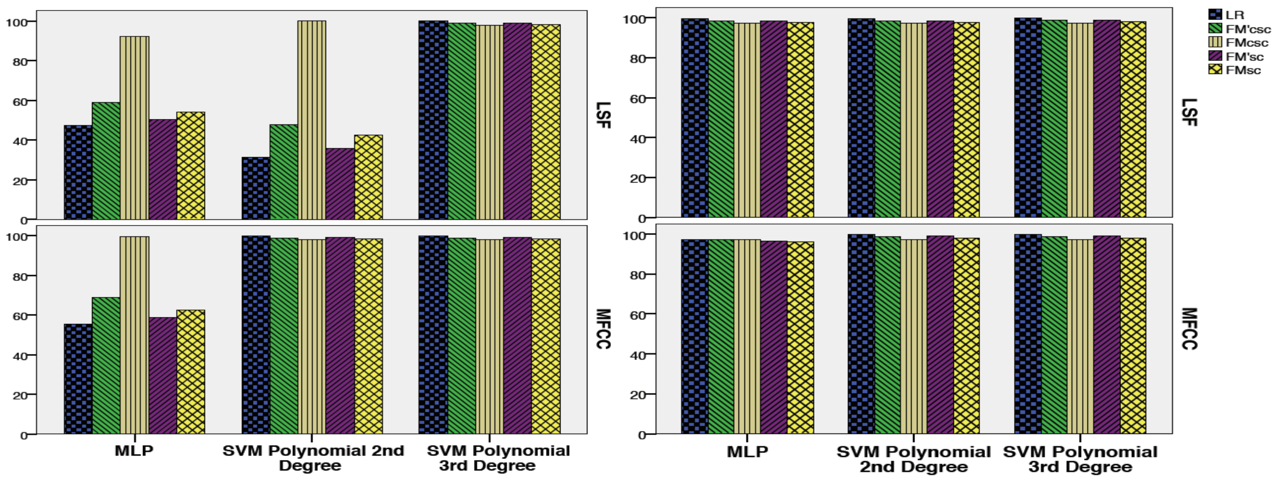
ANOVA test conducted for content-based analysis results, the Holm-Sidak procedure and a risk $\alpha$ of .05 were used in the MANOVA tests. In addition, we have used the same definitions in our interpretations of the effect sizes: small effect size ($\eta^2 \leq .01$), medium effect size ($.01 \leq \eta^2 \leq .06$) and large effect size ($.06 \leq \eta^2 \leq .14$).

### 7.4.1 Results of the Conceptual Analysis

The performance of six systems using LSFs and MFCCs are reported in Figure 7.1. For more detailed results regarding the performance of the systems, see Appendix A. For the isolated notes, overall, the total average identification rates were reasonably good for *LR* (72.33%). In parallel, the total average identification rates of the taxonomic relations were slightly lower than the correct lexical entries for $TF_{sc}$ (75.8%) and $TF'_{sc}$ (73.8%). The results shows that the highest total average identification rates are obtained for the $TF_{csc}$ (97.6%). But this improvement on the taxonomic layer of the ontologies is accompanied by a decrease on $TF'_{csc}$ (78.7%), after interaction with the correct lexical entry outputs (see Table A.2).

For the solo music dataset, on the other hand, the total average identification rates were higher than for the isolated notes dataset. For instance, the total average identification rate was very high on the lexical layer, *LR* (99.4%). Similarly, there were also very high performances for the taxonomic relations of the computed ontologies. For instance, the average identification rates of $TF_{sc}$ and $TF'_{sc}$, were 97.6% and 98.5%, respectively. In parallel, the average identification rates for $TF_{csc}$, and $TF'_{csc}$, were 97.4% and 98.3%, respectively (see Table A.3).

Regarding the average identification rates of the classifiers, the MLP obtained reasonably good results for *LR* (51.4%) on the lexical layer. There was slightly higher performances for $TF_{sc}$ (58.5%) and $TF'_{sc}$ (54.5%) on the taxonomic layer. However, the taxonomic relations were very high for $TF_{csc}$ (95.9%), whereas the results again dramatically dropped from an initial high value to a much lower value for $TF'_{csc}$ (54.5%) on the isolated notes dataset. On the contrary, the MLP performed very well on the solo music dataset. The average correct identification rate of *LR* was




(a) Isolated Notes Dataset

(b) Solo Music Dataset

Figure 7.1: Summary of the evaluation results for the isolated notes (Figure 7.1a) and solo music (Figure 7.1b) datasets. The bars refer to the following evaluation metrics: Lexical Recall (*LR*), higher level F-measure ($TF'_{csc}$), taxonomic F-measure ($TF_{csc}$) of the common semantic cotopy, higher level F-measure ($TF'_{sc}$), taxonomic F-measure ($TF_{sc}$) of the semantic cotopy metric, respectively.

98.3%. Similarly, there was a very high performance for $TF_{sc}$ (96.8%) and $TF'_{sc}$ (97.6%) on the taxonomic layer. In parallel with the semantic cotopy results, the MLP also obtained a very high performance for $TF_{csc}$ (97.3%) and $TF'_{csc}$ (97.8%).

The results show that the MLP classifier obtained the highest performance with 32 LSF and 16 codevectors, and 16 MFCCs and 64 codevectors on the isolated notes dataset. The highest performance of the MLP classifier was obtained with various codevector dimensions, for both LSFs and MFCCs on the solo music dataset. When using the SVM with a polynomial kernel of 2nd degree, the average correct identification rate increased up to 65.6% for *LR* on the lexical layer. There was also a reasonable increase for $TF_{sc}$ (75.8%) and $TF'_{sc}$ (73.8%) in terms of semantic cotopy, and a reasonable increase for $TF_{csc}$ (97.6%) and $TF'_{csc}$ (78.7%) in terms of common semantic cotopy, on the taxonomic layer of the generated ontologies for the isolated notes dataset. For the solo music dataset, there was a very small increase up to 99.7% for *LR* on the lexical layer of the generated ontologies. Likewise, there was also a very small increase up to 97.6% and 98.5% for $TF_{sc}$ and $TF'_{sc}$ in terms of semantic cotopy, and up to 97.4% and 98.3% for $TF_{csc}$ and $TF'_{csc}$, in terms of common semantic cotopy, on the taxonomic layer of the generated ontologies for the solo music dataset.

Overall, the best performance was obtained with the SVM using a polynomial kernel of 3rd degree. The results are almost identical for both the isolated notes and the solo music datasets. On the lexical layer, the average identification rate is 100% for *LR* on both of the datasets. On the taxonomic layer, the average identification rates were 98.4% and 99.2% for the $TF_{sc}$ and the $TF'_{sc}$, in terms of semantic cotopy, and 97.9% and 98.9% for $TF_{csc}$ and $TF'_{csc}$, in terms of the common semantic cotopy on the isolated notes dataset. Likewise, the average identification rates were 98.0% and 99.0% for $TF_{sc}$ and $TF'_{sc}$, in terms of semantic cotopy, and 97.4% and 98.7% for $TF_{csc}$ and $TF'_{csc}$, in terms of the common semantic cotopy on the solo music dataset.

With regards to the $ASFT$, the best overall performance was obtained with the MFCC on the isolated notes dataset, while the average identification rates were similar for both the MFCC and LSF on the solo music dataset. For instance, the best performance of LSF and MFCC feature sets were found mostly with 16 MFCCs and 64 codevectors and 32 LSFs and 16 codevectors for the isolated notes; for the solo music dataset MFCCs and LSFs yielded the best performance with various numbers of coefficients and codevectors dimensions.

### 7.4.2 Comparison of the Classifiers and the Spectral Feature Sets

The results of the three-way analyses of variance results for the conceptual analysis is reported in Table 7.3. The interaction between *CLSR* and $ASFT$ was highly significant for the isolated notes dataset, whereas only *CLSR* and its interaction with the spectral feature sets ($CLSR \times ASFT$) had a significant effect on the solo music dataset. These factors and the influence of their interactions are detailed in this section.

| Solo Music | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $LR$ | | | $TF'_{csc}$ | | | $TF_{csc}$ | | | $TF'_{sc}$ | | | $TF_{sc}$ | | |
| Source | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ |
| CLSR | 2 | 4.962* | .121 | 2 | 4.926* | .120 | 2 | 4.875* | .119 | 2 | 4.901* | .120 | 2 | 4.816* | .118 |
| ASFT | 1 | 1.846 | .025 | 1 | 1.862 | .025 | 1 | 1.880 | .025 | 1 | 1.836 | .025 | 1 | 1.818 | .025 |
| COEN | 3 | .615 | .025 | 3 | .621 | .025 | 3 | .629 | .026 | 3 | .623 | .025 | 3 | .634 | .026 |
| CLSR×ASFT | 2 | 3.577* | .090 | 2 | 3.566* | .090 | 2 | 3.549* | .090 | 2 | 3.568* | .090 | 2 | 3.553* | .090 |
| CLSR×COEN | 6 | .500 | .040 | 6 | .519 | .041 | 6 | .542 | .043 | 6 | .517 | .041 | 6 | .539 | .043 |
| ASFT×COEN | 3 | 1.231 | .049 | 3 | 1.242 | .049 | 3 | 1.255 | .050 | 3 | 1.246 | .049 | 3 | 1.266 | .050 |
| CLSR×ASFT×COEN | 6 | .654 | .052 | 6 | .674 | .053 | 6 | .698 | .055 | 6 | .669 | .053 | 6 | .687 | .054 |
| Error | 72 | | | 72 | | | 72 | | | 72 | | | 72 | | |

| Isolated Notes | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $LR$ | | | $TF'_{csc}$ | | | $TF_{csc}$ | | | $TF'_{sc}$ | | | $TF_{sc}$ | | |
| Source | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ | $df$ | $F$ | $\eta^2$ |
| CLSR | 2 | 98.767*** | .733 | 2 | 63.753*** | .639 | 2 | .717 | .020 | 2 | 90.949*** | .716 | 2 | 81.545*** | .694 |
| ASFT | 1 | 77.982*** | .520 | 1 | 60.564*** | .457 | 1 | .743 | .010 | 1 | 73.634*** | .506 | 1 | 66.767*** | .481 |
| COEN | 3 | .999 | .040 | 3 | 1.169 | .046 | 3 | .797 | .032 | 3 | 1.083 | .043 | 3 | 1.210 | .048 |
| CLSR×ASFT | 2 | 55.702*** | .607 | 2 | 36.186*** | .501 | 2 | 1.959 | .052 | 2 | 50.704*** | .585 | 2 | 44.223*** | .551 |
| CLSR×COEN | 6 | .999 | .077 | 6 | 1.169 | .089 | 6 | .797 | .062 | 6 | 1.083 | .083 | 6 | 1.210 | .092 |
| ASFT×COEN | 3 | .640 | .026 | 3 | .521 | .021 | 3 | .868 | .035 | 3 | .555 | .023 | 3 | .452 | .018 |
| CLSR×ASFT×COEN | 6 | .640 | .051 | 6 | .521 | .042 | 6 | .868 | .067 | 6 | .555 | .044 | 6 | .452 | .036 |
| Error | 72 | | | 72 | | | 72 | | | 72 | | | 72 | | |

Table 7.3: Results of Three-Way Analyses of Variance for the Automatic Ontology Generation System based on the isolated notes and solo music datasets. $\eta^2$ is the partial eta squared measure of effect size. $^{*}p < .05, ^{**}p < .01, ^{***}p < .001$. CSLR: classifier; ASFT: audio spectral feature set; COEF: coefficient number.

*Classifier*

For the isolated notes dataset, highly significant effect of the classifiers (*CLSR*) was found on *LR* of the generated ontologies on the lexical layer with very large effect size (($p < .001, \eta^2 = .733$). With regard to the taxonomic relations, there were also highly significant effects of the classifier on the $TF_{sc}$ and $TF'_{sc}$ with very large effect sizes for the semantic cotopy, and highly significant effect only on $TF'_{csc}$ with a very large effect size for the common semantic cotopy (all $p < .001$). The post-hoc analysis, however, revealed that there was no significant effect of the classifiers on $TF_{csc}$ ($p > .05$).

For the solo music dataset, the post-hoc test also revealed that there was significant effect of the classifiers on *LR* of the generated ontologies with a large effect size. Regarding the taxonomic relations, similarly, there was also significant effect of the classifiers on $TF_{sc}$ and $TF'_{sc}$ with large effect sizes for the semantic cotopy, and significant effect on $TF_{csc}$ and $TF'_{csc}$ with large effect sizes for the common semantic cotopy (all $p < .05$).

On the lexical layer, the post-hoc analyses (multiple comparison procedure) conducted for the isolated notes dataset revealed that the SVM classifier with a polynomial kernel of degree 2 was significantly better than the MLP classifier ($p <.05$), and the SVM classifier with a polynomial kernel of degree 3 was highly significantly better than the MLP classifier ($p <.001$) on *LR*. Consequently, the average difference on *LR* was large between the MLP and the SVM classifiers with both 2nd order and 3rd order polynomials for the isolated notes dataset. The results indicates that there was also significant difference on *LR* between the MLP and the SVM classifiers with 2nd order and 3rd order polynomials for the solo music dataset ($p < .05$).

On the taxonomic layer, subsequently, post-hoc analysis revealed that the MLP was significantly different than the SVM classifier with a 2nd order polynomial kernel ($p <.05$), and highly significantly different than the SVM with 3rd order polynomial on $TF_{sc}$, $TF'_{sc}$ and $TF'_{csc}$ ($p <.001$) . There was no significant effect on the $TF_{csc}$ for any classifier utilised on the isolated notes dataset

($p > 0.5$). Additionally, the difference for these classifiers was large on $TF_{sc}$, $TF'_{sc}$, and $TF'_{csc}$, and smaller on the $TF_{csc}$.

There was a significant difference between the MLP and the SVM classifiers ($p < 0.5$) independent from the polynomial degree on all the outcome variables for the solo music dataset. Compared to the isolated notes dataset, the difference was small between the MLP and the SVM classifiers with a 2nd order polynomial $TF_{sc}$, $TF'_{sc}$, $TF_{csc}$, and $TF'_{csc}$ for the solo music dataset. Similarly, there was also a small difference between the MLP and the SVM classifier with degree 3 on $TF_{sc}$, $TF'_{sc}$, $TF_{csc}$, and $TF'_{csc}$.

*Audio Spectral Feature Set*

The post-hoc test has not been used for the examination of the *ASFT*, since the number of groups (MFCCs and LSFs) were less than three. Instead, the pairwise comparison test has been used to analyse the effect of the Spectral Feature Sets at the .05 level based on Holm-Sidak procedure. There was a highly significant effect of the *ASFT* on *LR*, $TF_{sc}$, $TF'_{sc}$, and $TF'_{csc}$ (all $p < 001$), and no significant effect on $TF_{csc}$ ($p > 0.5$). Unlike the isolated notes dataset, there was a significant effect of the $ASFT$ on *LR*, $TF_{sc}$, $TF'_{sc}$, $TF_{csc}$ and $TF'_{csc}$ (all $p < 05$).

Contrary to the results obtained with content-based analysis, the LSF feature sets performed slightly better than the MFCC feature set on the isolated music dataset, whereas MFCCs performed slightly better than LSFs for the conceptual analysis on the solo music dataset.

### 7.4.3 Relationships Between the Factors

Interaction between the Classifier (*CLSR*) and Audio Spectral Feature Set ($ASFT$) factors was highly significant on *LR* with a very large effect size for the isolated notes and significant on *LR* with a large effect size for the solo music dataset.

Regarding the taxonomic relations, there was also highly significant effect of the interaction of *CSLR* and $ASFT$ with very large effect sizes on $TF_{sc}$ and $TF'_{sc}$ in terms of the semantic cotopy, and

$TF'_{csc}$ in terms of the common semantic cotopy. There was no significant effect on the $TF_{csc}$ for the isolated notes. There was significant effect of interaction with large effect sizes on $TF_{sc}$ and $TF'_{sc}$ in terms of the semantic cotopy, and similarly, significant effects with large effect sizes on $TF_{csc}$ and $TF'_{csc}$ in terms of the common semantic cotopy.

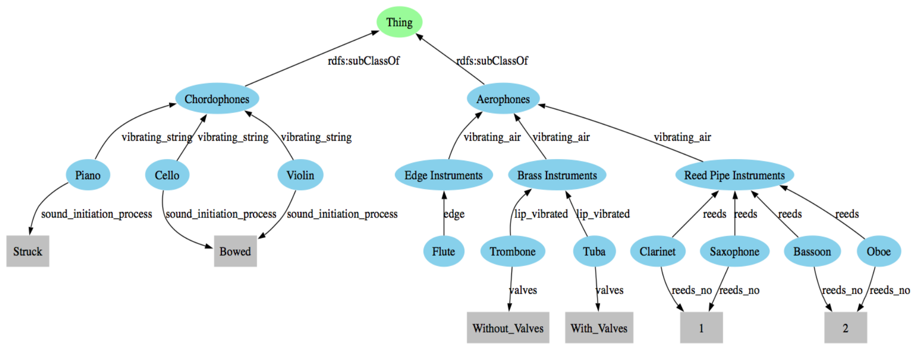
Contrary to the content-based analysis results, there was no significant effects of interaction between factors *CLSR* and *COEN*, as well as *ASFT* and *COEN* for the conceptual analysis results on datasets. Although both of the factors have indicated significant effects for instrument recognition, these do not affect the conceptual analysis as the identifications are good enough in all cases.

### 7.4.4 Discussion

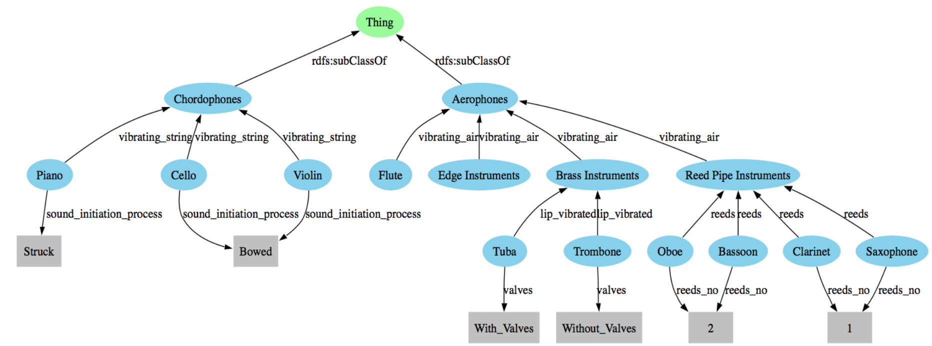
Instrument recognition performances in range of 43.5% and 90.3% for the isolated notes dataset, and 69.9% and 87.5% for the solo music dataset, have been obtained. The hand-crafted reference concept hierarchy (based on Hornbostel and Sachs' system) is denoted, $\mathcal{O}_{Ref1}$, and the automatically-generated concept hierarchies associated with the highest to lowest recognition system performance are denoted as $\mathcal{O}_{C1}$, $\mathcal{O}_{C2}$ and $\mathcal{O}_{C3}$ for the isolated notes. Likewise, the $\mathcal{O}_{Ref2}$ denote the hand-crafted reference concept hierarchy, and $\mathcal{O}_{C4}$, $\mathcal{O}_{C5}$ and $\mathcal{O}_{C6}$ the automatically generated concept hierarchies for the solo music dataset. Figure 7.2 and Figure 7.3 illustrate the automatically generated and reference ontologies for isolated notes and solo music, respectively.

Compared to $\mathcal{O}_{Ref1}$ and $\mathcal{O}_{Ref2}$, the values of the taxonomic measures are slightly lower than the corresponding values of the lexical measures of the isolated notes and the solo music datasets, since there is no error on the lexical term layer. It should be noted that the {*Brass Instruments, Tuba, Trombone*} concepts was not taken into account in the evaluations of solo music, since these instruments were not present in the solo music dataset.

As can be seen in Table A.1, the semantic cotopy of the ontology $\mathcal{O}_{C1}$ is almost identical to the reference ontology $\mathcal{O}_{Ref1}$. For example, the semantic cotopy of the concept *Aerophones* in the hand-crafted ontology ($\mathcal{O}_{ref}$) in Fig. 7.2a is {*Thing, Aerophones, Edge instruments, Brass Instru-*

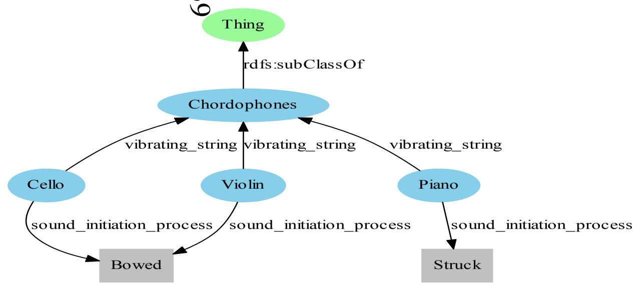



(a) $\mathcal{O}_{Ref1:}$, ground-truth based on Hornbostel and Sachs' instrument classification terminology

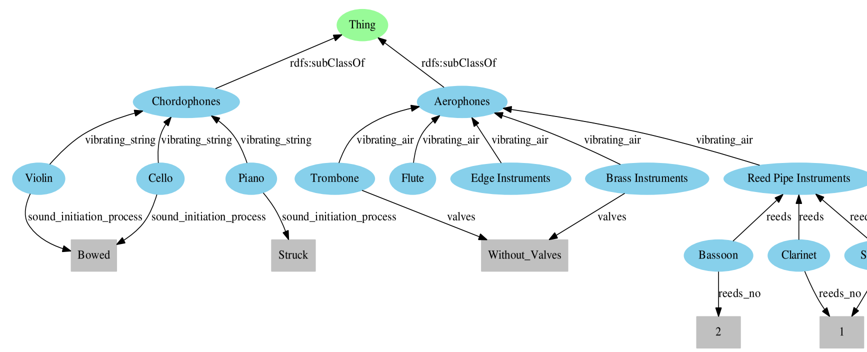



(b) $\mathcal{O}_{C1:}$, SVM with the 3rd. degree polynomial kernel using 32 LSF features and 64 codevectors for the isolated notes






(c) $\mathcal{O}_{C2:}$, MLP using 16 MFCC features and 8 codevectors, and SVM with the 2nd degree polynomial kernel using 24 LSF and 64 codevectors for the isolated notes.




(d) $\mathcal{O}_{C3:}$, MLP using 32 LSF features and 16 codevectors for the isolated notes.

Figure 7.2: Automatically generated concept hierarchies $\mathcal{O}_{C1,C2,C3}$, compared to the reference concept hierarchy $\mathcal{O}_{Ref-1}$

*ments, Reed Pipe Instruments, Flute, Trombone, Tuba, Clarinet, Saxophone, Bassoon, Oboe*} and the semantic cotopy of the *Aerophones* in the automatically generated ontology in Fig 7.2b, $\mathcal{O}_{C1}$, is {*Thing, Aerophones, Edge instruments, Brass Instruments, Reed Pipe Instruments, Flute, Trombone, Tuba, Clarinet, Saxophone, Bassoon, Oboe*}. That is, the semantic cotopy of the concept *Aerophones* is identical in both ontologies. It is possible to see the same identical overlap for other concepts as well. However, for the concepts *Edge Instruments* and *Flute* the semantic cotopy in $\mathcal{O}_{ref}$ is {*Thing, Aerophones, Edge Instruments, Flute*}, whereas for the concept *Edge Instruments*, the semantic cotopy in $\mathcal{O}_{C1}$ is {*Thing, Aerophones, Edge Instruments*}, and for the *Flute*, it is }*Thing, Aerophones, Flute*}.

Parallel to this, there were slightly lower results for the common semantic cotopy of ontology $\mathcal{O}_{C1}$. It is possible to see the same problem with this metric as well, for example, the common semantic cotopies of the concept *Aerophones* and many other concepts are identical (e.g., {*Thing, Edge instruments, Brass Instruments, Reed Pipe Instruments, Flute, Trombone, Tuba, Clarinet, Saxophone, Bassoon, Oboe*}). However, the conceptual hierarchy dissimilarity of the *Edge Instruments* and the *Flute* concepts were also reflected on common semantic cotopy measurements; for instance, for the *Edge Instruments* and the *Flute* in $\mathcal{O}_{ref}$, the common semantic cotopies are *Thing, Aerophones, Flute* and *Thing, Aerophones, Edge Instruments*, whereas in $\mathcal{O}_{C1}$, for both of the concepts, it is *Thing, Aerophones*.

In fact, except the concepts *Edge Instruments* and *Flute*, every leaf concept in the reference concept hierarchy has a maximum overlap with the corresponding concept hierarchy in $\mathcal{O}_{C1}$. Thus, it is evident from the results shown by $\mathcal{O}_{C1}$, in terms of arrangement of the leaf nodes and by making abstraction of the inner nodes, obtained fairly high results as shown in Fig. 7.2b. The good correspondences obtained from the instrument identification system lead to a high precision and recall with respect to the taxonomic overlap.

There was an important distinction for the conceptual analysis results shown by $\mathcal{O}_{C2}$ and $\mathcal{O}_{C3}$, as the obtained musical instrument identification results were 48.1% and 56.4% for $\mathcal{O}_{C2}$ and $\mathcal{O}_{C3}$, respectively. In the case of the MLP classifier for $\mathcal{O}_{C2}$, the incorrect instrument identifications led to some contradictions in the pruning stage for the concept *Aerophones*. The contradiction was mainly caused by the concept *Edge Instruments* which did not possess the {*vibrating_air*} property. The system, therefore, couldn't identify the right infimum edge to establish necessary pruning on the lattice form. In parallel, in case of the SVM classifier for $\mathcal{O}_2$, a similar problem occurred for the concept *Aerophones* which did not possess the {*reed_no:2*} property.

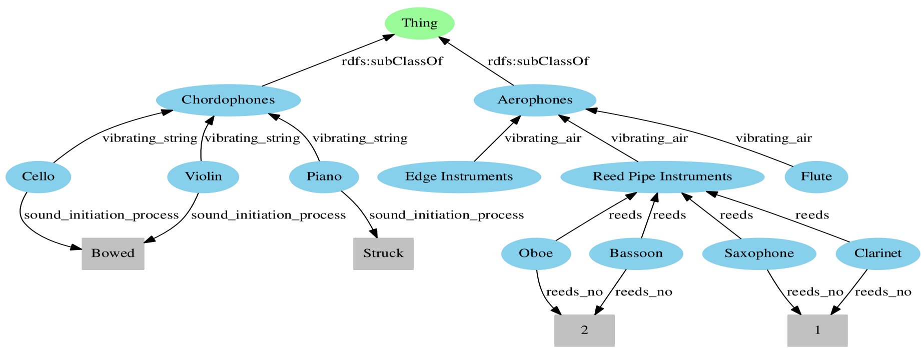
On the contrary, although the instrument identification results were low for the $\mathcal{O}_3$, there was no issues on removing the infimum edges in the lattice form. However, as a result of the low identification ratio, different issues occurred, such as missing associations, which resulted in the loss of two concepts (i.e. *Bassoon*, and *Tuba*). Since, there was low accuracy results for the concept *Tuba*, the system was not able to dentify the concept *Trombone* alone as a subclass of the concept *Brass Instruments*, despite the fact that the concept *Trombone* and *Brass Instruments* were predicted correctly along with their properties.

When compared to $\mathcal{O}_{Ref1}$, eleven concepts are missing in $\mathcal{O}_{C2}$, and two concepts are missing in $\mathcal{O}_{C3}$, but the hierarchy of the remaining concepts are not changed for $\mathcal{O}_{C2}$. This leads to very low (31.3%) Lexical Recall (*LR*), result for $\mathcal{O}_{C2}$, and very good *LR* result (87.5%) for $\mathcal{O}_{C3}$. On the other hand, a perfect ($TF_{csc}$=100%) taxonomic F-measure for the common semantic cotopy of $\mathcal{O}_{C2}$ and an excellent taxonomic F-measure ($TF_{csc}$=95%) for $\mathcal{O}_{C3}$ were obtained. For instance, the common semantic cotopy of the concept *Chordophones* for $\mathcal{O}_{Ref1}$ in Fig.7.2a is {*Thing, Cello, Violin, Piano*} and the common semantic cotopy of the *Chordophones* in the automatically-generated ontology, $\mathcal{O}_{C2}$ in Fig 7.2c, is {*Thing, Cello, Violin, Piano*}; it is identical in both ontologies. It is possible to see the same identical overlap for other concepts, such as *Thing, Cello, Violin* and *Piano* of $\mathcal{O}_{C2}$. This pattern is repeated for all the concepts of $\mathcal{O}_{C3}$ as well. The errors on the lexical term layer of the learned ontologies are fairly smaller than the $TF_{csc}$ result for $\mathcal{O}_{C2}$ and slightly smaller than the

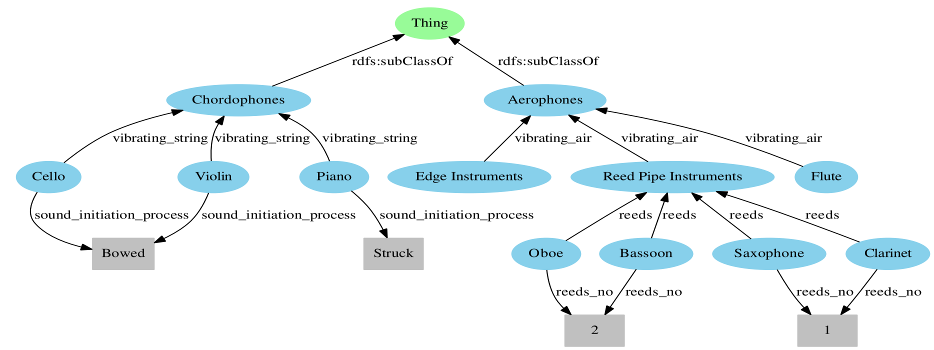
$TF_{csc}$ result for $\mathcal{O}_{C3}$. Consequently, $TF'_{csc}$ of $\mathcal{O}_{C2}$ dramatically dropped from 100% $TF_{csc}$ to 47.6% $TF'_{csc}$ with the influence of the lexical measure in the calculation. On the contrary, the effect of the lexical measure was slightly smaller on $\mathcal{O}_{C3}$ which led to a 91.1% $TF'_{csc}$ performance.

However, it is worth pointing out that the influence of the lexical measure is higher on the common semantic cotopy compared to the semantic cotopy taxonomy measures in $\mathcal{O}_{C2}$. As a result, 42.5% $TF_{sc}$ and 36.0% $TF'_{sc}$ performances were obtained for $\mathcal{O}_{C2}$, and 87.4% $TF_{sc}$ and 87.4% $TF'_{sc}$ for $\mathcal{O}_{C3}$. This is mainly due to the fact that the common semantic cotopy considers only the common semantics for the reference ontology and the automatically generated ontology, whilst the semantic cotopy takes into account the complete semantics present in the corresponding ontologies. For instance, the local taxonomic recall and precision for the common semantic cotopy, $TR_{csc}$ and $TP_{csc}$, of the concept *Brass Instruments* in the automatically-generated ontology, $\mathcal{O}_{C3}$ in Fig 7.2d, are {*Thing, Aerophones, Trombone*} and {*Thing, Aerophones*}. This identical overlap led to 66.6% local taxonomic recall and 100% local taxonomic precision for the concept *Brass Instruments*. On the other hand, the local taxonomic recall for the semantic cotopy, $TR_{sc}$, of the concept *Brass Instruments* in the automatically-generated ontology, $\mathcal{O}_{C3}$, is {*Thing, Aerophones, Brass Instruments, Tuba, Trombone*}. And the local taxonomic precision for the semantic cotopy, $TP_{sc}$, of the concept *Brass Instruments* in $\mathcal{O}_{C3}$, is {*Thing, Aerophones, Brass Instruments*}. Consequently, $TR_{sc}$ dropped to 60% while $TP_{sc}$ was unaffected (100%).

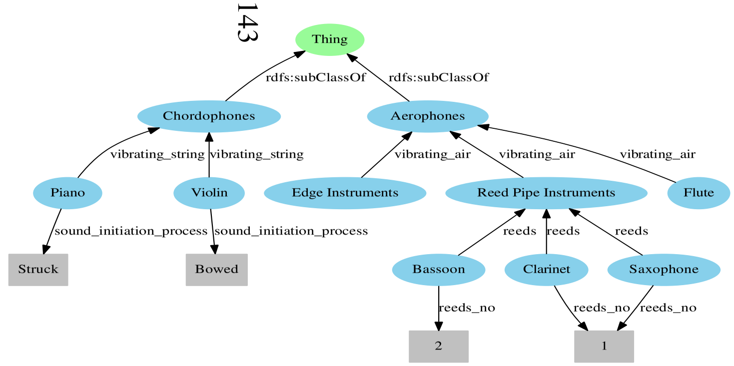
For the SVM classifier-based ontologies, independent from the degree of polynomial kernel, the average correct ontology identification rates were very high on the solo music dataset (see Table A.1). Therefore, it is highly probable that the conceptual terms successfully passed the predefined threshold (50%) regarding the binary context. This evidence suggests that the SVM-based ontologies generated from the solo music dataset were unaffected by the changes in spectral feature sets and the codebook dimensions. As can be seen in the fourth ontology ($\mathcal{O}_{C4}$) in Fig. 7.3b, the hierarchy of the ontology was not changed except for the concepts of *Edge Instruments*



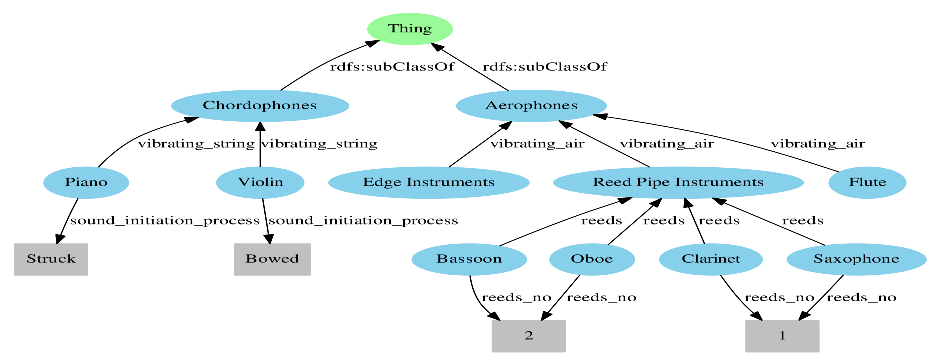

(a) $\mathcal{O}_{Ref2:}$, ground-truth based on Hornbostel and Sachs' instrument classification terminology




(b) $\mathcal{O}_{C4:}$, SVM with the 3rd. degree polynomial kernel using 32 LSF features and 64 codevectors for solo music




(c) $\mathcal{O}_{C5:}$, MLP using 24 MFCC features and 16 codevectors for solo music.




(d) $\mathcal{O}_{C6:}$, MLP using 8 MFCC features and 32 codevectors for solo music.

Figure 7.3: Automatically generated concept hierarchies $\mathcal{O}_{C4,C5,C6}$, compared to the reference concept hierarchy $\mathcal{O}_{Ref-2}$

and *Flute*. Due to the fact that there is only one leaf for the Edge Instruments, and that every data in the concept of Flute is most likely to be the same as for the Edge Instruments, it may be reasonable to assume that this case represent a challenge for generating ontologies by FCA using audio signals. The *Flute* could only be separated from *Edge Instrument* if there was at least one more instrument concept with an identical attribute. Thus, the problem could be solved in the lattice pruning phase by removing the corresponding lower bound, infimum edge, of the *Edge Instruments*.

For the MLP classifier-based ontologies, on the other hand, the average correct identification rates were reasonably lower to that obtained with the SVM classifier based systems (see Table A.2 and Table A.3). The average instrument identification rates were slightly higher (77.9%) for $\mathcal{O}_{C6}$, in Fig7.3d, than (73.2%) for $\mathcal{O}_{C5}$, in Fig. 7.3c, in line with the average ontology identification rates. The higher taxonomic F-measures $TF'_{csc}$ and $TF'_{sc}$ were 90.3% and 85.5% for $\mathcal{O}_{C5}$, respectively; whereas the higher taxonomic F-measures $TF'_{csc}$ and $TF'_{sc}$ were 94.7% and 92.5% for $\mathcal{O}_{C6}$, respectively.

When compared to $\mathcal{O}_{Ref2}$, two concepts are missing in $\mathcal{O}_{C5}$, and one concept is missing in $\mathcal{O}_{C6}$. This leads to 84.6% and 92.3% Lexical Recall (*LR*) for $\mathcal{O}_{C5}$ and $\mathcal{O}_{C6}$, respectively. For instance, the common semantic cotopy of the concept *Reed Pipe Instruments* for $\mathcal{O}_{Ref2}$ in Fig. 7.3a is {*Thing, Aerophones, Saxophone, Clarinet, Bassoon*} and the common semantic cotopy of the *Reed Pipe Instruments* in the automatically generated ontology, $\mathcal{O}_{C5}$ in Fig. 7.3c, is {*Thing, Aerophones, Saxophone, Clarinet, Bassoon*}. That is, the taxonomic recall and precision for the common semantic cotopy of the concept *Chordophones* are 100% and 100%, respectively. On the other hand, the concept *Oboe* is missing from the $\mathcal{O}_{C6}$, thus, the common semantic cotopy of the concept *Reed Pipe Instruments* for $\mathcal{O}_{Ref2}$ is {*Thing, Aerophones, Saxophone, Clarinet, Oboe, Bassoon*}, and the common semantic cotopy of the concept *Reed Pipe Instruments* for $\mathcal{O}_6$ is {*Thing, Aerophones, Saxophone, Clarinet, Oboe, Bassoon*}. Consequently, the taxonomic recall and precision for the common semantic cotopy of the concept *Reed Pipe Instruments* for $\mathcal{O}_6$ are 100% and 100%, respectively. Although the errors of the taxonomic overlaps look similar, overall more concepts are predicted for $\mathcal{O}_6$ on the lexical layer, which led to obtain slightly better results on $TF_{csc}$.

The problems in the lattice form of $\mathcal{O}_{C5}$ were caused by the low accuracy in the identification of associations, such as the fact that the concepts *Cello* and *Oboe* didn't pass the 0.5 threshold to be associated with their corresponding properties {*vibrating_string, sound initiation process:Bowed*} and {*vibrating_air, reeds_no:2*}, respectively. In case of $\mathcal{O}_6$, on the other hand, the missing concepts were caused by similar identification problems, yet only for the concept *Cello*. A complete listing of the generated ontologies are available in Appendix D.

## 7.5 Discussion

The proposed automatic ontology generation system is based on content-based audio analysis. Musical instrument identification has been investigated as an application domain. Given that musical instrument identification is a very difficult task on its own, extraction of ontologies from musical instruments makes the work even more complicated. For that reason, in an effort to accomplish a good performance, we grounded our system on the prior studies by Chétry [2006] and Barthet & Sandler [2010b], who investigated musical instrument identification using the LSF and MFCC. We have evaluated the system performance at identifying and conceptualising musical instruments extracted from isolated notes and solo music. The performance of our system was compared to a gold-standard ontology based on H-S terminology. The results revealed that the models comprising average spectral envelopes determined using K-means algorithm were able to capture the essential information about the musical instruments, and it has also indicated how well FCA performed in conceptualisation of musical instruments.

As we discussed in section 7.3.5 and 7.4.4, when we compared the performance of the SVM classifiers with the ones that can be achieved with the MLP classifier, the results has shown that a classification approach using SVM yielded better performance than a learning approach using MLP, and musical instrument identification based on the traditional MLP classifier was inappropriate to deal with perceptual scales for the isolated notes. On the one hand, this can be explained by the characteristics of the SVM classifier such the use of a kernel function to map the data into a higher-dimensional space where the classes become linearly separable as well as the determination of an optimal hyperplane that minimises the probability of misclassifications. Using inherent nonlinear approximation properties provide the capability to model very complex patterns for SVM. This is unlike the traditional MLP which relies on linear basis function architectures. On the other hand, while reverberation may affect the instrument recognition system [Barthet & Sandler, 2010a], in this case, it can also be explained by the contextual information obtained analysing musical phrases that substantially improved the identification rate compared to isolated notes.

Overall, we obtained promising ontology results for the system. In essence, the *CLSR* was the only factor that has shown significant effect with a very large effect size on the ontology outputs, whereas there was no significant effect for the rest of the factors (i.e. $ASFT$, $COEN$, $CLUN$), although they have established significant effects with small or medium effect size on the content-based analysis part of the system. However, the challenge in the proposed system is that it is dependant on a supervised learning. As a result, it requires a new training and determination of new instrument models once a new instrument has been included in the database. This should be further replicated with an adaptive and dynamic learning algorithm, as it is not how learning process is developed in real life: it is actually a dynamic and interactive process.

The acoustic timbral descriptors were also chosen based on a prior knowledge, and experimentally further verified for automatic ontology generation process. Spectral envelope descriptors, both the LFSs and MFCCs, have been found to be a significant contributor with a small effect size to modelling the timbre of sound as can be seen in the content-based analysis part of the system. Obtaining small difference between these audio descriptors can be explained by the fact that they both characterise the spectral envelope of the sounds. High performance of both descriptors could be also partly attributed to the utilisation of "5 s-long audio segments", which enabled to capture a better representation of cepstral features, as described in [Essid *et al.*, 2006]. While in this experiment we have obtained a very efficient instrument family discrimination, future work should include different audio descriptors to be able to identify different instrument families, for instance membraphones or idiophones. Moreover, it should be investigated with multi-descriptor based approaches including automatic feature selection algorithms.

Additionally, another important question remains for the evaluation of the threshold which was manually determined as 50%. During the experiments, we have realised that threshold value also offers a crucial impact on the ontology results. Even so, we haven't taken this factor into consideration due to excessive number of variables that have been used in the statistical analysis. Thus, possible research direction should also include an investigation for threshold value, for instance in the range 10-90%, to re-examine the effects of the factors that haven't shown any significant effect in the conceptual analysis part, thereby on the ontology outputs.

Our system can be applied to a wide range of content-based audio applications. For instance, machine recognition of animal sounds may be another interesting application domain for our system. Identification of animals by their sounds is a valuable resource for biological research and environmental monitoring applications, particularly in detecting, locating and ecological censusing animals. Due to the fact that birds and their sounds are in many ways important for our culture, and birdsong has long been a significant source of inspiration for many composers, musicians, and writers: identification of birdsong has also taken a particular interest in the MIR field. For birdsong identification similar techniques that are designed for human speech analysis are commonly used, such as Short-Time Fourier Transforms, power spectral density, linear predictive coding. and cepstral analysis. MFCCs are widely used in several studies [Chou *et al.*, 2008] and [Graciarena *et al.*, 2010]. For instance, Lee *et al.* [2006] used MFCCs and LPCCs as the vocalisation feature in the syllables; and exploited a codebook comprising a number of features to model the variant characteristics of different syllables segmented from the same bird songs. In another study, Fagerlund [2007] compared three cepstral representations based on linear prediction, Mel spectra or perceptual liner prediction, in which it has been suggested that there was very small difference between these features for a classification task. Of course a throughout review of birdsong identification is beyond the scope of this thesis, however, it provides an interesting research opportunity and a second example for the possible use of our audio analysis-based automatic ontology generation system. The birdsong identification has a rich literature, and a detailed overview can be found in [Stowell & Plumbley, 2011].

## 7.6 Summary

The evaluation approaches and the applied research fields have been reviewed in this chapter. In the core, we have highlighted the human assessment, task-based, data-driven, and gold-standard based ontology evaluation techniques. As a result, considering the limited number of instrument categories in the automatically generated instrument ontologies in our system, and the difficulty of finding labeled music pieces with a single instrument, we have chosen to use gold-standard base evaluation rather than a task-based evaluation.

Thus, we have subsequently described the gold-standard based evaluation metrics used in our system. The gold-standard evaluation metric have been explained by considering the lexical and the conceptual level analysis of ontologies. We have also given some examples along with the evaluation metrics.

Next, we provided the evaluations of the content-based analysis and conceptual analysis of the automatic ontology generation system presented in the previous chapter. The system has been evaluated with two different datasets. The system was tested using MLP and SVM classifiers by modelling different codebook dimensions for the timbre features, LSFs and MFCCs, on various instruments for the wind and string families in order to find the best performance.

We found that in all cases the system succeeded to generate a very good ontology for the solo music dataset. Contrarily, significant effects of the classifier type and the number of codebook vectors on the ontology generation were highlighted for the isolated notes dataset. Overall, the best results were obtained for the SVM with a polynomial kernel of the 3rd degree using 32 LSF and dictionaries of 64 codevectors in both the solo music (87.5%) and isolated notes (90.3%) datasets for the content based analysis.

In terms of the conceptual analysis, the highest performance were obtained with SVM classifiers independent from the degree of the polynomial on both the isolated notes and solo music datasets. For instance, the SVM classifiers were obtained 100% *LR* on the lexical layer, and 98.9% for $TF'_{csc}$ and 99.2% for $TF'_{sc}$ on the taxonomic layer of the isolated note-based ontologies. Similarly, the best results were obtained for both of the classifiers on the solo music dataset. On the lexical layer, the highest results were 100%, and 98.7% for $TF'_{csc}$ and 99.0% for $TF'_{sc}$ on the taxonomic layer of the solo music dataset based ontologies. Even though the results were varied depending on a few codebook dimensions for the MFCCs, generally there seemed to be very good results also for the MLP classifier on the solo music dataset.

With regards to the spectral feature set, the LSF feature set were yielded lower performance with the SVM classifier with a polynomial kernel of the 2nd degree on the isolated notes dataset, whereas the results were very good, like the MFCC feature set, except in the case of the 8 LSF with 8 codebook vectors. In general, reasonable results were obtained with regard to the hierarchical design of the instrument ontology on both datasets.

# Chapter 8

# Conclusions

---

Our research is motivated by the fact that current music ontology design processes do not incorporate automated learning systems. This makes the design of ontologies highly difficult and dependent on human supervision. To date, only manually-created ontologies have been proposed, therefore, we have presented a hybrid system to build an ontology supported by the OWL using timbre-based automatic instrument recognition techniques and Formal Concept Analysis.

In general, reasonable results were obtained with regard to the hierarchical design of the instrument ontology on the isolated notes and solo music datasets. In the context of the Semantic Web, these findings confirm that the proposed hybrid system enables a novel methodical design for automatic ontology generation. Most notably, to the author's knowledge, this is the first study to investigate automatic ontology generation in the context of audio and music analysis. In addition, the proposed system can be applied to a broad range of research fields investigating knowledge exploitation and management. In this chapter we will summarise our work, discuss its limitations and present ideas for future work.

## 8.1 Review of contents

In chapter 3, in order to better understand the connection between the Semantic Web and ontologies, we provide a survey on the semantic richness of ontology representation models, and then present the semantic web technologies for ontology representation that have been used throughout the thesis. Afterwards, we review the ontology engineering works to date on the semantic web.

In chapter 4, we investigate the heterogeneity and limitations in existing instrument classification schemes. We develop representations of the taxonomic instrument classification schemes, based on Hornbostel and Sach's classification scheme, using OWL and compare terminological and conceptual heterogeneity using SPARQL queries. We demonstrate that traditional designs based on taxonomy trees lead to ill-defined knowledge representation, especially in the context of an ontology for the Semantic Web.

In chapter 5, we review fundamental studies are related to the automatic ontology generation systems. We review the input data types and the state of the art frameworks for automatic ontology generation (section 5.1). Next, we review machine learning (section **??**), content-based audio analysis (section 6.3), and conceptual analysis techniques (section 5.3) that are employed for musical instrument identification to date.

In chapter 6, we present the proposed ontology generation framework describing each component along with examples using experimental datasets. We provide the analysis of parameters for the feature extraction and the audio classification components of the content-based analysis part of the system. We also illustrate the automatic process of generation of ontologies based on the proposed framework for isolated notes, since the isolated notes dataset involves slightly larger terminology compare to the solo music dataset. We present the OWL specifications used in our experiments together with a sample from a generated ontology. We treat every concepts as an OWL class and every property as either OWL object properties or OWL data properties. The instrument audio signals are modelled using our framework and are generated corresponding ontologies for each combination of the analysis parameters. We describe how our framework could provide a conceptual hierarchy along with properties as a solution to the ill-defined knowledge representation of instrument classification schemes presented in chapter 4.

In chapter 7, we review ontology evaluation techniques (section 7.1), and present the methodology used in experiments (section 7.2). The introduced methodology evaluates how well a machine-based ontology is similar to a ground truth ontology in terms of conceptual hierarchy. We also show illustrations of the various ontology metrics used for assessment of the system. Our framework is evaluated by Analysis of Variance for the content-based analysis results and Multivariate Analysis of Variance for the conceptual analysis results in order to assess the main effects and interactions occurs combinations of factors. Additionally, we discuss the quantitative evaluations together with the visual ontology graphs.

### 8.1.1 Contributions of this work

This research aims to investigate how the process of developing ontologies can be made less dependent on human supervision by exploring conceptual analysis techniques in a Semantic Web environment. The contributions of this work are as follows:

- It addresses the question, how to embed content-based audio analysis techniques in the ontology engineering process by providing a comprehensive framework for automatic ontology generation.
- It provides a detailed analysis on the effects of the various combinations of the classifiers, audio spectral feature sets, and codebook dimensions for the automatically generated ontologies. A particular emphasis is placed for each factor and its influence on the quality of ontology outputs.
- It shows how to evaluate automatically generated ontologies using several different measures applied within a gold standard setting.

- It gives suggestions on other possible application domains that audio analysis-based automatic ontology generation system can be used.
- It reveals knowledge representation issues of musical instruments on the SemanticWeb, by taking musical instrument classification schemes into account; and an assessment of the OWL representations of these classification schemes using SPARQL queries.
- It contributes to Linked Data by filtering and publishing a large set of music audio similarity features produced by the SoundBite playlist generator tool.

## 8.2 Limitations and future work

The study presented here has some limitations and there are also numerous possibilities for future work.

### 8.2.1 Formal Concept Analysis

In our proposed system we deal with conceptual hierarchies in order to automatically obtain ontologies for musical instruments. Modified version of the reduced labelling technique has been used in order to obtain conceptual hierarchies in the pruning stage. However, our system does not deal with very complex graphical knowledge representations. Even though the findings are promising, future work should be carried on by a more in-depth examination of the reduced labelling techniques on more complicated conceptual graphs to tackle knowledge management issues.

Moreover, It should be noted that our system does not automatically deal with many valued context issue. We provide information regarding the data properties simply using a colon between the property and data value. Future work could therefore be aimed at retrieving necessary characteristics for every properties through SPARQL queries made from available knowledge-bases data, such as DBpedia, on the Web.

### 8.2.2 The effect of threshold on the concept/property associations

The formal context was the phase in which outcome of content-based analysis were assessed by a threshold (50%) to determine the associations among instrumental categories and properties. During our preliminary results, we found that threshold variations might affect the system which may lead to obtain different ontology outputs. Therefore, although it is evident that the findings are promising, there is an important opportunity for future studies to examine the effects of threshold parameters.

### 8.2.3 Adaptive and dynamically learning system

The proposed system is based on batch learning, and it is an automatic system which carries out fixed functions on some available prior knowledge without the intervention of an ontology engineer. However, it is not a dynamic nor an adaptive system to learn about new instruments and re-design ontology. Future work should, therefore, include an investigation on which algorithms should be used to obtain an adaptive and dynamically learning system, since it is how learning process is developed in real life.

### 8.2.4 Multimodal data

In our experiments, we have used two different audio collections in order to examine our system. Further work, nonetheless, is needed towards incorporating a wider set of musical instruments and attributes and utilising more OWL language features. Additionally, there is also a great deal of advantage of utilising multimodal data (e.g. textual, audio and visual) in an effort to cope with the wide diversity of data available on the Web (e.g. text, sound, photos, and videos). This will likely encompass a greater understanding of the complex nature of human language, expressions and associated actions for potential smart devices.

### 8.2.5 Application Scenarios

As we pointed out within the introduction, exchanging data among diverse knowledge repositories has substantial positive aspects for the discovery of cultural heritages, such as musical instrument museums, libraries, and institutions. Thus, there is a huge potential for machines to exploit the growing quantity of information and ontologies on the internet. Having an explicitly defined knowledge repository is a necessity for an effective control system in which machines can share and re-use their data-driven knowledge representations through machine readable languages, such as Ontology Web Language (OWL). For these reasons, the system could be further developed by implementing autonomous and dynamic systems in which smart devices and robots can take advantage of the vast amount of data available on the Web.

As a possible use case scenario, the proposed system may be used as a plugin tab for ontology editor applications, such as Protege, in order to facilitate the ontology development process for ontology engineers. Due to the fact that different sounds can indicate different characteristics of musical instruments, the processing of information directly derived from audio content contains very important clues for the retrieval and management of musical instrument knowledge. Therefore, the proposed system may also be used to overcome issues in current musical instrument classification schemes developed in organology and musicology.

In addition, our system may be examined in the machine recognition of animal sounds, such as birdsong identification. Many animals generate sounds either for communication or their living activities such as eating, moving, or flying; most of the animal vocalisations have evolved to be species-specific. Identification of their sounds could offer a great deal of benefit for biological research and environmental monitoring. Since similar audio descriptors are used in birdsong identification (i.e. MFCCs), it provides an interesting research opportunity and a second example for the possible use of our audio analysis-based automatic ontology generation system.

# Appendix A

# Detailed results of the conceptual analysis

---

The performance of the six systems using LSFs and MFCCs are reported in Table A.1. The performance of MLP-based systems are reported for the isolated notes dataset in Table A.2, and for the solo music dataset in Table A.3.

| Clsr | Asft | Coen | Clun | $LR$ | $TF'_{csc}$ | $TF_{csc}$ | $TF'_{sc}$ | $TF_{sc}$ |
|---|---|---|---|---|---|---|---|---|
| MLP | ∀ | ∀ | ∀ | ∗ | ∗ | ∗ | ∗ | ∗ |
| SVM w/ 2nd deg. poly. | MFCC | ∀ | ∀ | 100 | 98.9 | 97.9 | 99.2 | 98.4 |
| | LSF | ∀ | ∀ | **31.3** | **47.6** | **100** | **36.0** | **42.5** |
| SVM w/ 3rd deg. poly. | ∀ | ∀ | ∀ | **100** | **98.9** | **97.9** | **99.2** | **98.4** |

(a) Isolated Notes Dataset

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| MLP | ∀ | ∀ | ∀ | ∗ | ∗ | ∗ | ∗ | ∗ |
| SVM w/ 2nd deg. poly. | MFCC | ∀ | ∀ | 100 | 98.7 | 97.4 | 99.0 | 98.0 |
| | LSF | 8 | 8 | 92.3 | 94.7 | 97.1 | 92.4 | 92.4 |
| | | 16-32 | 16-64 | 100 | 98.7 | 97.4 | 99.0 | 98.0 |
| SVM w/ 3rd deg. poly. | ∀ | ∀ | ∀ | **100** | **98.7** | **97.4** | **99.0** | **98.0** |

(b) Solo Music Dataset

Table A.1: Summary table for the evaluation results of the isolated notes and solo music datasets. The performance of factors given for all parameters: ∀ indicates all parameters of a factor and ∗ indicates that corresponding results are detailed in the Table A.2 and Table A.3. The ontology outputs of SVM 3rd order polynomial kernel correspond to $\mathcal{O}_{C1}$ for isolated notes in Figure 7.2, and $\mathcal{O}_{C4}$ for solo music in Figure 7.3. For the isolated notes dataset, the ontology output of the SVM with the 2nd degree polynomial kernel using 24 LSF and 64 codevectors and MLP using 16 MFCC features and 8 codevectors correspond to $\mathcal{O}_{C2}$ in Figure 7.2. In each case, the corresponding performance is reported in bold.

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 68.8 | 31.3 | 18.8 | 68.8 | 50.0 | 31.3 | 56.3 | - |
| 16 | **31.3** | 81.3 | 68.8 | 31.3 | 31.3 | 31.3 | 93.8 | 31.3 |
| 24 | 68.8 | 31.3 | 68.8 | 31.3 | 75.0 | 81.3 | 68.8 | 68.8 |
| 32 | 31.3 | 87.5 | 31.3 | **87.5** | 62.5 | 31.3 | 62.5 | 31.3 |

(a) *LR* — MLP on the Isolated Notes Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 81.5 | 47.6 | 31.6 | 81.5 | 66.7 | 47.6 | 72.0 | - |
| 16 | **47.6** | 87.7 | 81.5 | 47.6 | 47.6 | 47.6 | 95.7 | 47.6 |
| 24 | 81.5 | 47.6 | 81.5 | 47.6 | 84.6 | 88.6 | 81.5 | 80.4 |
| 32 | 47.6 | 88.5 | 47.6 | **91.1** | 76.9 | 47.6 | 76.9 | 47.6 |

(b) $TF'_{csc}$ — MLP on the Isolated Notes Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | - |
| 16 | **100** | 95.3 | 100 | 100 | 100 | 100 | 97.7 | 100 |
| 24 | 100 | 100 | 100 | 100 | 97.1 | 97.4 | 100 | 96.9 |
| 32 | 100 | 89.5 | 100 | **95.0** | 100 | 100 | 100 | 100 |

(c) $TF_{csc}$ — MLP on the Isolated Notes Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 73.2 | 36.0 | 19.4 | 72.0 | 53.3 | 36.0 | 59.4 | - |
| 16 | **36.0** | 82.5 | 72.0 | 36.0 | 36.0 | 36.0 | 93.3 | 36.0 |
| 24 | 71.1 | 36.0 | 71.5 | 36.0 | 77.3 | 82.6 | 72.3 | 71.1 |
| 32 | 36.0 | 84.9 | 36.0 | **87.4** | 66.9 | 36.0 | 65.6 | 36.0 |

(d) $TF'_{sc}$ — MLP on the Isolated Notes

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 78.2 | 42.5 | 20.1 | 75.6 | 57.2 | 42.5 | 62.8 | - |
| 16 | **42.5** | 83.7 | 75.6 | 42.5 | 42.5 | 42.5 | 92.8 | 42.5 |
| 24 | 73.7 | 42.5 | 74.6 | 42.5 | 79.8 | 84.0 | 76.3 | 73.7 |
| 32 | 42.5 | 82.5 | 42.5 | **87.4** | 71.9 | 42.5 | 69.1 | 42.5 |

(e) $TF_{sc}$ — MLP on the Isolated Notes Dataset

Table A.2: Performance of the Automatic Ontology Generation System for MLP on the isolated notes datasets. The ontology outputs of MLP using 16 MFCC features and 8 codevectors corresponds to $\mathcal{O}_{C2}$, and MLP using 32 LSF features and 16 codevectors corresponds to $\mathcal{O}_{C3}$ in Figure 7.2. In each case, the corresponding performance is reported in bold.

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 100 | 100 | 100 | 100 | **92.3** | 92.3 | 100 | 100 |
| 16 | 100 | 100 | 100 | 100 | 100 | 100 | 92.3 | 100 |
| 24 | 100 | 100 | **84.6** | 100 | 100 | 100 | 92.3 | 100 |
| 32 | 100 | 100 | 100 | 100 | 92.3 | 100 | 100 | 100 |

(a) $LR$ — MLP on the Solo Music Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 98.7 | 98.7 | 98.7 | 98.7 | **94.7** | 94.7 | 98.7 | 98.7 |
| 16 | 98.7 | 98.7 | 98.7 | 98.7 | 98.7 | 98.7 | 94.7 | 98.7 |
| 24 | 98.7 | 98.7 | **90.3** | 98.7 | 98.7 | 98.7 | 94.7 | 98.7 |
| 32 | 98.7 | 98.7 | 98.7 | 98.7 | 94.7 | 98.7 | 98.7 | 98.7 |

(b) $TF'_{csc}$ — MLP on the Solo Music Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 97.4 | 97.4 | 97.4 | 97.4 | **97.1** | 97.1 | 97.4 | 97.4 |
| 16 | 97.4 | 97.4 | 97.4 | 97.4 | 97.4 | 97.4 | 97.1 | 97.4 |
| 24 | 97.4 | 97.4 | **96.9** | 97.4 | 97.4 | 97.4 | 97.1 | 97.4 |
| 32 | 97.4 | 97.4 | 97.4 | 97.4 | 97.1 | 97.4 | 97.4 | 97.4 |

(c) $TF_{csc}$ — MLP on the Solo Music Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 99.0 | 99.0 | 99.0 | 99.0 | **92.5** | 92.5 | 99.0 | 99.0 |
| 16 | 99.0 | 99.0 | 99.0 | 99.0 | 99.0 | 99.0 | 92.5 | 99.0 |
| 24 | 99.0 | 99.0 | **85.5** | 99.0 | 99.0 | 99.0 | 92.5 | 99.0 |
| 32 | 99.0 | 99.0 | 99.0 | 99.0 | 92.5 | 99.0 | 99.0 | 99.0 |

(d) $TF'_{sc}$ — MLP on the Solo Music Dataset

| | No. of clusters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | | 16 | | 32 | | 64 | |
| Coen | MFCC | LSF | MFCC | LSF | MFCC | LSF | MFCC | LSF |
| 8 | 98.0 | 98.0 | 98.0 | 98.0 | **92.7** | 92.7 | 98.0 | 98.0 |
| 16 | 98.0 | 98.0 | 98.0 | 98.0 | 98.0 | 98.0 | 92.7 | 98.0 |
| 24 | 98.0 | 98.0 | **86.4** | 98.0 | 98.0 | 98.0 | 92.7 | 98.0 |
| 32 | 98.0 | 98.0 | 98.0 | 98.0 | 92.7 | 98.0 | 98.0 | 98.0 |

(e) $TF_{sc}$ — MLP on the Solo Music Dataset

Table A.3: Performance of the Automatic Ontology Generation System for MLP on the solo music datasets. The ontology outputs of MLP using 24 MFCC features and 16 codevectors corresponds to $\mathcal{O}_{C5}$, and MLP using 8 MFCC features and 32 codevectors corresponds to $\mathcal{O}_{C6}$ in Figure 7.3. In each case, the corresponding performance is reported in bold.

# Appendix B

# Namespaces

The following namespaces are used throughout this work:

```
@prefix : <http://www.semanticweb.org/ontologies/sio#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://www.semanticweb.org/ontologies/sio> .

<http://www.semanticweb.org/ontologies/sio> rdf:type owl:Ontology .
```

Listing B.1: The set of prefixes that are assumed in the turtle listings throughout this work.

# Appendix C

# Musical Instrument Taxonomies

---

Here we provide the corresponding turtle listings that model the examples described in Chapter 7. The complete modelling can be found at corresponding URL[1].

```
# prefixes in Appendix A are assumed.

#################################################################
# Classes
#################################################################

### http://www.semanticweb.org/ontologies/sio#Idiophones
:Idiophones rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Membraphones
:Membraphones rdf:type owl:Class .
```

[1] http://www.isophonics.net/content/musical-instrument-taxonomies/

```
### http://www.semanticweb.org/ontologies/sio#Strigilated
:Strigilated rdf:type owl:Class ;
       rdfs:subClassOf :Idiophones.

### http://www.semanticweb.org/ontologies/sio#Kettle
:Kettle rdf:type owl:Class ;

       rdfs:subClassOf :Membraphones.

### http://www.semanticweb.org/ontologies/sio#Single_Head
:Single_Head rdf:type owl:Class ;
       rdfs:subClassOf :Membraphones.

### http://www.semanticweb.org/ontologies/sio#Pair
:Pair rdf:type owl:Class ;
       rdfs:subClassOf :Kettle.

### http://www.semanticweb.org/ontologies/sio#Sets
:Sets rdf:type owl:Class ;
       rdfs:subClassOf :Kettle.

### http://www.semanticweb.org/ontologies/sio#Single
:Single rdf:type owl:Class ;
       rdfs:subClassOf :Single_Head.
```

Listing C.1: Turtle representation of $\mathcal{O}_{EX2}$ (Figure **??**)

```
# prefixes in Appendix A are assumed.
```

```
#################################################################
# Classes
#################################################################

### http://www.semanticweb.org/ontologies/sio#Idiophones
:Idiophones rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Membraphones
:Membraphones rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Shaken
:Shaken rdf:type owl:Class ;
       rdfs:subClassOf :Idiophones.

### http://www.semanticweb.org/ontologies/sio#Strigilated
:Strigilated rdf:type owl:Class ;
       rdfs:subClassOf :Idiophones.

### http://www.semanticweb.org/ontologies/sio#Kettle
:Kettle rdf:type owl:Class ;
       rdfs:subClassOf :Membraphones.

### http://www.semanticweb.org/ontologies/sio#Single_Head
:Single_Head rdf:type owl:Class ;
       rdfs:subClassOf :Membraphones.

### http://www.semanticweb.org/ontologies/sio#Double_Head
:Double_Head rdf:type owl:Class ;
```

```
        rdfs:subClassOf :Membraphones.

### http://www.semanticweb.org/ontologies/sio#Pair
:Pair rdf:type owl:Class ;
        rdfs:subClassOf :Kettle.

### http://www.semanticweb.org/ontologies/sio#Sets
:Sets rdf:type owl:Class ;
        rdfs:subClassOf :Kettle.

### http://www.semanticweb.org/ontologies/sio#Single
:Single rdf:type owl:Class ;
        rdfs:subClassOf :Kettle.
```

Listing C.2: Turtle representation of $\mathcal{O}_{JM2}$ (Figure **??**) reference ontology based on Jeremy Montagu & John Burton's instrument classification system.

# Appendix D

# Automatically Generated Ontologies

These are the automatically generated ontologies which are presented in Chapter 7. For more details regarding the complete OWL files generated in the experiment, see this link[1].

```
# prefixes in Appendix A are assumed.

#################################################################
# Object Properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#lip_vibrated
:lip_vibrated rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#reeds
:reeds rdf:type owl:ObjectProperty .
```

[1] http://www.isophonics.net/content/auto-ontology-generation

```
### http://www.semanticweb.org/ontologies/sio#vibrating_air
:vibrating_air rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#vibrating_string
:vibrating_string rdf:type owl:ObjectProperty .

#################################################################
# Data  properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds_no
:reeds_no rdf:type owl:DatatypeProperty .

### http://www.semanticweb.org/ontologies/sio#
   sound_initiation_process
:sound_initiation_process rdf:type owl:DatatypeProperty .

### http://www.semanticweb.org/ontologies/sio#valves
:valves rdf:type owl:DatatypeProperty .

#################################################################
# Classes
#################################################################

### http://www.semanticweb.org/ontologies/sio#Aerophone
:Aerophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Bassoon
:Bassoon rdf:type owl:Class ;
```

```
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                          owl:onProperty :reeds ;
                          owl:someValuesFrom :Reed_Pipe_Instruments
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :reeds_no ;
                          owl:hasValue "2"
                        ] .

### http://www.semanticweb.org/ontologies/sio#Brass_Instruments
:Brass_Instruments rdf:type owl:Class ;
              rdfs:subClassOf [ rdf:type owl:Restriction ;
                                owl:onProperty :vibrating_air ;
                                owl:someValuesFrom :Aerophone
                              ] .

### http://www.semanticweb.org/ontologies/sio#Cello
:Cello rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                        owl:onProperty :sound_initiation_process ;
                        owl:hasValue "Bowed"
                      ] ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty :vibrating_string ;
                        owl:someValuesFrom :Chordophone
                      ] .

### http://www.semanticweb.org/ontologies/sio#Chordophone
:Chordophone rdf:type owl:Class .
```

```
### http://www.semanticweb.org/ontologies/sio#Clarinet
:Clarinet rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                      owl:onProperty :reeds ;
                      owl:someValuesFrom :Reed_Pipe_Instruments
                    ] ,
                    [ rdf:type owl:Restriction ;
                      owl:onProperty :reeds_no ;
                      owl:hasValue "1"
                    ] .

### http://www.semanticweb.org/ontologies/sio#Edge_Instruments
:Edge_Instruments rdf:type owl:Class ;
              rdfs:subClassOf [ rdf:type owl:Restriction ;
                            owl:onProperty :vibrating_air ;
                            owl:someValuesFrom :Aerophone
                          ] .

### http://www.semanticweb.org/ontologies/sio#Flute
:Flute rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                   owl:onProperty :vibrating_air ;
                   owl:someValuesFrom :Aerophone
                 ] .

### http://www.semanticweb.org/ontologies/sio#Oboe
:Oboe rdf:type owl:Class ;
    rdfs:subClassOf [ rdf:type owl:Restriction ;
```

```
                 owl:onProperty :reeds ;
                 owl:someValuesFrom :Reed_Pipe_Instruments
                ] ,
                [ rdf:type owl:Restriction ;
                 owl:onProperty :reeds_no ;
                 owl:hasValue "2"
                ] .

### http://www.semanticweb.org/ontologies/sio#Piano
:Piano rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                 owl:onProperty :vibrating_string ;
                 owl:someValuesFrom :Chordophone
                ] ,
                [ rdf:type owl:Restriction ;
                 owl:onProperty :sound_initiation_process ;
                 owl:hasValue "Struck"
                ] .

### http://www.semanticweb.org/ontologies/sio#Reed_Pipe_Instruments
:Reed_Pipe_Instruments rdf:type owl:Class ;

                 rdfs:subClassOf [ rdf:type owl:Restriction ;
                             owl:onProperty :vibrating_air ;
                             owl:someValuesFrom :Aerophone
                            ] .

### http://www.semanticweb.org/ontologies/sio#Saxophone
:Saxophone rdf:type owl:Class ;
```

```
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                          owl:onProperty :reeds ;
                          owl:someValuesFrom :Reed_Pipe_Instruments
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :reeds_no ;
                          owl:hasValue "1"
                        ] .

### http://www.semanticweb.org/ontologies/sio#Trombone
:Trombone rdf:type owl:Class ;

        rdfs:subClassOf [ rdf:type owl:Restriction ;
                          owl:onProperty :lip_vibrated ;
                          owl:someValuesFrom :Brass_Instruments
                        ] ,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty :valves ;
                          owl:hasValue "Without_Valves"
                        ] .

### http://www.semanticweb.org/ontologies/sio#Tuba
:Tuba rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                       owl:onProperty :valves ;
                       owl:hasValue "With_Valves"
                     ] ,
                     [ rdf:type owl:Restriction ;
                       owl:onProperty :lip_vibrated ;
```

```
                    owl:someValuesFrom :Brass_Instruments
                  ] .

### http://www.semanticweb.org/ontologies/sio#Violin
:Violin rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                   owl:onProperty :vibrating_string ;
                   owl:someValuesFrom :Chordophone
                 ] ,
                 [ rdf:type owl:Restriction ;
                   owl:onProperty :sound_initiation_process ;
                   owl:hasValue "Bowed"
                 ] .
```

Listing D.1: OWL representation of $\mathcal{O}_{C1}$ (Figure 7.2b) for SVM with the 3rd order polynomial kernel using 32 LSF features and 64 codevectors for the isolated notes dataset.

```
# prefixes in Appendix A are assumed.

#################################################################
# Object Properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#vibrating_string
:vibrating_string rdf:type owl:ObjectProperty .

#################################################################
# Data properties
#################################################################
```

```
### http://www.semanticweb.org/ontologies/sio#
   sound_initiation_process
:sound_initiation_process rdf:type owl:DatatypeProperty .

#################################################################
# Classes
#################################################################

### http://www.semanticweb.org/ontologies/sio#Cello
:Cello rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                   owl:onProperty :sound_initiation_process ;
                   owl:hasValue "Bowed"
                  ] ,
                  [ rdf:type owl:Restriction ;
                   owl:onProperty :vibrating_string ;
                   owl:someValuesFrom :Chordophone
                  ] .

### http://www.semanticweb.org/ontologies/sio#Chordophone
:Chordophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Piano
:Piano rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                   owl:onProperty :sound_initiation_process ;
                   owl:hasValue "Struck"
                  ] ,
```

```
                 [ rdf:type owl:Restriction ;
                  owl:onProperty :vibrating_string ;
                  owl:someValuesFrom :Chordophone
                 ] .

### http://www.semanticweb.org/ontologies/sio#Violin
:Violin rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :sound_initiation_process ;
                  owl:hasValue "Bowed"
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty :vibrating_string ;
                  owl:someValuesFrom :Chordophone
                 ] .
```

Listing D.2: OWL representation of $\mathcal{O}_{C2}$ (Figure 7.2c) for MLP using 16 MFCC features and 8 codevectors — and SVM with the 2nd degree polynomial kernel using 24 LSF and 64 codevectors for the isolated notes dataset.

```
# prefixes in Appendix A are assumed.

#################################################################
# Object Properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds
:reeds rdf:type owl:ObjectProperty .
```

```
### http://www.semanticweb.org/ontologies/sio#vibrating_air
:vibrating_air rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#vibrating_string
:vibrating_string rdf:type owl:ObjectProperty .

#################################################################
# Data  properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds_no
:reeds_no rdf:type owl:DatatypeProperty .

### http://www.semanticweb.org/ontologies/sio#
   sound_initiation_process
:sound_initiation_process rdf:type owl:DatatypeProperty .

### http://www.semanticweb.org/ontologies/sio#valves
:valves rdf:type owl:DatatypeProperty .

#################################################################
# Classes
#################################################################

### http://www.semanticweb.org/ontologies/sio#Aerophone
:Aerophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Bassoon
:Bassoon rdf:type owl:Class ;
```

```
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds ;
                    owl:someValuesFrom :Reed_Pipe_Instruments
                   ] ,
                   [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds_no ;
                    owl:hasValue "2"
                   ] .

### http://www.semanticweb.org/ontologies/sio#Brass_Instruments
:Brass_Instruments rdf:type owl:Class ;

              rdfs:subClassOf [ rdf:type owl:Restriction ;
                           owl:onProperty :vibrating_air ;
                           owl:someValuesFrom :Aerophone
                          ] ,
                          [ rdf:type owl:Restriction ;
                           owl:onProperty :valves ;
                           owl:hasValue "Without_Valves"
                          ] .

### http://www.semanticweb.org/ontologies/sio#Cello
:Cello rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :sound_initiation_process ;
                  owl:hasValue "Bowed"
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty :vibrating_string ;
```

```
                owl:someValuesFrom :Chordophone
              ] .


### http://www.semanticweb.org/ontologies/sio#Chordophone
:Chordophone rdf:type owl:Class .


### http://www.semanticweb.org/ontologies/sio#Clarinet
:Clarinet rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :reeds_no ;
                  owl:hasValue "1"
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :reeds ;
                  owl:someValuesFrom :Reed_Pipe_Instruments
                ] .


### http://www.semanticweb.org/ontologies/sio#Edge_Instruments
:Edge_Instruments rdf:type owl:Class ;
              rdfs:subClassOf [ rdf:type owl:Restriction ;
                          owl:onProperty :vibrating_air ;
                          owl:someValuesFrom :Aerophone
                        ] .


### http://www.semanticweb.org/ontologies/sio#Flute
:Flute rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                owl:onProperty :vibrating_air ;
                owl:someValuesFrom :Aerophone
```

```
                ] .

### http://www.semanticweb.org/ontologies/sio#Piano
:Piano rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                 owl:onProperty :sound_initiation_process ;
                 owl:hasValue "Struck"
                ] ,
                [ rdf:type owl:Restriction ;
                 owl:onProperty :vibrating_string ;
                 owl:someValuesFrom :Chordophone
                ] .

### http://www.semanticweb.org/ontologies/sio#Reed_Pipe_Instruments
:Reed_Pipe_Instruments rdf:type owl:Class ;
                rdfs:subClassOf [ rdf:type owl:Restriction ;
                             owl:onProperty :vibrating_air ;
                             owl:someValuesFrom :Aerophone
                            ] .

### http://www.semanticweb.org/ontologies/sio#Saxophone
:Saxophone rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds_no ;
                    owl:hasValue "1"
                   ] ,
                   [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds ;
                    owl:someValuesFrom :Reed_Pipe_Instruments
```

```
                    ] .


### http://www.semanticweb.org/ontologies/sio#Trombone
:Trombone rdf:type owl:Class ;
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                         owl:onProperty :vibrating_air ;
                         owl:someValuesFrom :Aerophone
                       ] ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty :valves ;
                         owl:hasValue "Without_Valves"
                       ] .


### http://www.semanticweb.org/ontologies/sio#Violin
:Violin rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                        owl:onProperty :vibrating_string ;
                        owl:someValuesFrom :Chordophone
                      ] ,
                      [ rdf:type owl:Restriction ;
                        owl:onProperty :sound_initiation_process ;
                        owl:hasValue "Bowed"
                      ] .
```

Listing D.3: OWL representation of $\mathcal{O}_{C3}$ (Figure 7.2d) for MLP using 32 LSF features and 16 codevectors on the isolated notes dataset.

```
# prefixes in Appendix A are assumed.
```

```
#################################################################
# Object Properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds
:reeds rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#vibrating_air
:vibrating_air rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#vibrating_string
:vibrating_string rdf:type owl:ObjectProperty .

#################################################################
# Data  properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds_no
:reeds_no rdf:type owl:DatatypeProperty .

### http://www.semanticweb.org/ontologies/sio#
   sound_initiation_process
:sound_initiation_process rdf:type owl:DatatypeProperty .

#################################################################
# Classes
#################################################################
```

```
### http://www.semanticweb.org/ontologies/sio#Aerophone
:Aerophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Bassoon
:Bassoon rdf:type owl:Class ;
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                         owl:onProperty :reeds ;
                         owl:someValuesFrom :Reed_Pipe_Instruments
                       ] ,
                       [ rdf:type owl:Restriction ;
                         owl:onProperty :reeds_no ;
                         owl:hasValue "2"
                       ] .

### http://www.semanticweb.org/ontologies/sio#Cello
:Cello rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                       owl:onProperty :vibrating_string ;
                       owl:someValuesFrom :Chordophone
                     ] ,
                     [ rdf:type owl:Restriction ;
                       owl:onProperty :sound_initiation_process ;
                       owl:hasValue "Bowed"
                     ] .

### http://www.semanticweb.org/ontologies/sio#Chordophone
:Chordophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Clarinet
```

```
:Clarinet rdf:type owl:Class ;
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds_no ;
                    owl:hasValue "1"
                   ] ,
                   [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds ;
                    owl:someValuesFrom :Reed_Pipe_Instruments
                   ] .

### http://www.semanticweb.org/ontologies/sio#Edge_Instruments
:Edge_Instruments rdf:type owl:Class ;
             rdfs:subClassOf [ rdf:type owl:Restriction ;
                          owl:onProperty :vibrating_air ;
                          owl:someValuesFrom :Aerophone
                         ] .

### http://www.semanticweb.org/ontologies/sio#Flute
:Flute rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :vibrating_air ;
                  owl:someValuesFrom :Aerophone
                 ] .

### http://www.semanticweb.org/ontologies/sio#Oboe
:Oboe rdf:type owl:Class ;
    rdfs:subClassOf [ rdf:type owl:Restriction ;
                 owl:onProperty :reeds_no ;
                 owl:hasValue "2"
```

```
                ] ,
                [ rdf:type owl:Restriction ;
                 owl:onProperty :reeds ;
                 owl:someValuesFrom :Reed_Pipe_Instruments
                ] .

### http://www.semanticweb.org/ontologies/sio#Piano
:Piano rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :sound_initiation_process ;
                  owl:hasValue "Struck"
                 ] ,
                 [ rdf:type owl:Restriction ;
                  owl:onProperty :vibrating_string ;
                  owl:someValuesFrom :Chordophone
                 ] .

### http://www.semanticweb.org/ontologies/sio#Reed_Pipe_Instruments
:Reed_Pipe_Instruments rdf:type owl:Class ;
                 rdfs:subClassOf [ rdf:type owl:Restriction ;
                              owl:onProperty :vibrating_air ;
                              owl:someValuesFrom :Aerophone
                             ] .

### http://www.semanticweb.org/ontologies/sio#Saxophone
:Saxophone rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds ;
                     owl:someValuesFrom :Reed_Pipe_Instruments
```

```
                    ] ,
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds_no ;
                     owl:hasValue "1"
                    ] .

### http://www.semanticweb.org/ontologies/sio#Violin
:Violin rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                     owl:onProperty :vibrating_string ;
                     owl:someValuesFrom :Chordophone
                    ] ,
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :sound_initiation_process ;
                     owl:hasValue "Bowed"
                    ] .
```

Listing D.4: OWL representation of $\mathcal{O}_{C4}$ (Figure 7.3b) for SVM with the 3rd. degree polynomial kernel using 32 LSF features and 64 codevectors on the solo music dataset.

```
# prefixes in Appendix A are assumed.

#################################################################
# Object Properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds
:reeds rdf:type owl:ObjectProperty .
```

```
### http://www.semanticweb.org/ontologies/sio#vibrating_air
:vibrating_air rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#vibrating_string
:vibrating_string rdf:type owl:ObjectProperty .

#################################################################
# Data  properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds_no
:reeds_no rdf:type owl:DatatypeProperty .

### http://www.semanticweb.org/ontologies/sio#
   sound_initiation_process
:sound_initiation_process rdf:type owl:DatatypeProperty .

#################################################################
# Classes
#################################################################

### http://www.semanticweb.org/ontologies/sio#Aerophone
:Aerophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Bassoon
:Bassoon rdf:type owl:Class ;
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds ;
                    owl:someValuesFrom :Reed_Pipe_Instruments
```

```
                    ] ,
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds_no ;
                     owl:hasValue "2"
                    ] .

### http://www.semanticweb.org/ontologies/sio#Chordophone
:Chordophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Clarinet
:Clarinet rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds_no ;
                     owl:hasValue "1"
                    ] ,
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds ;
                     owl:someValuesFrom :Reed_Pipe_Instruments
                    ] .

### http://www.semanticweb.org/ontologies/sio#Edge_Instruments
:Edge_Instruments rdf:type owl:Class ;
              rdfs:subClassOf [ rdf:type owl:Restriction ;
                           owl:onProperty :vibrating_air ;
                           owl:someValuesFrom :Aerophone
                          ] .

### http://www.semanticweb.org/ontologies/sio#Flute
:Flute rdf:type owl:Class ;
```

```
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                      owl:onProperty :vibrating_air ;
                      owl:someValuesFrom :Aerophone
                    ] .

### http://www.semanticweb.org/ontologies/sio#Piano
:Piano rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                      owl:onProperty :vibrating_string ;
                      owl:someValuesFrom :Chordophone
                    ] ,
                    [ rdf:type owl:Restriction ;
                      owl:onProperty :sound_initiation_process ;
                      owl:hasValue "Struck"
                    ] .

### http://www.semanticweb.org/ontologies/sio#Reed_Pipe_Instruments
:Reed_Pipe_Instruments rdf:type owl:Class ;
                  rdfs:subClassOf [ rdf:type owl:Restriction ;
                                   owl:onProperty :vibrating_air ;
                                   owl:someValuesFrom :Aerophone
                                 ] .

### http://www.semanticweb.org/ontologies/sio#Saxophone
:Saxophone rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                         owl:onProperty :reeds_no ;
                         owl:hasValue "1"
                       ] ,
```

```
                 [ rdf:type owl:Restriction ;
                   owl:onProperty :reeds ;
                   owl:someValuesFrom :Reed_Pipe_Instruments
                 ] .

### http://www.semanticweb.org/ontologies/sio#Violin
:Violin rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                   owl:onProperty :sound_initiation_process ;
                   owl:hasValue "Bowed"
                 ] ,
                 [ rdf:type owl:Restriction ;
                   owl:onProperty :vibrating_string ;
                   owl:someValuesFrom :Chordophone
                 ] .
```

Listing D.5: OWL representation of $\mathcal{O}_{C5}$ (Figure 7.3c) for MLP using 24 MFCC features and 16 codevectors on the solo music dataset.

```
# prefixes in Appendix A are assumed.

#################################################################
# Object Properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds
:reeds rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#vibrating_air
```

```
:vibrating_air rdf:type owl:ObjectProperty .

### http://www.semanticweb.org/ontologies/sio#vibrating_string
:vibrating_string rdf:type owl:ObjectProperty .

#################################################################
# Data  properties
#################################################################

### http://www.semanticweb.org/ontologies/sio#reeds_no
:reeds_no rdf:type owl:DatatypeProperty .

### http://www.semanticweb.org/ontologies/sio#
   sound_initiation_process
:sound_initiation_process rdf:type owl:DatatypeProperty .

#################################################################
# Classes
#################################################################

### http://www.semanticweb.org/ontologies/sio#Aerophone
:Aerophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Bassoon
:Bassoon rdf:type owl:Class ;
       rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty :reeds ;
                    owl:someValuesFrom :Reed_Pipe_Instruments
                  ] ,
```

```
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds_no ;
                     owl:hasValue "2"
                    ] .

### http://www.semanticweb.org/ontologies/sio#Chordophone
:Chordophone rdf:type owl:Class .

### http://www.semanticweb.org/ontologies/sio#Clarinet
:Clarinet rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds_no ;
                     owl:hasValue "1"
                    ] ,
                    [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds ;
                     owl:someValuesFrom :Reed_Pipe_Instruments
                    ] .

### http://www.semanticweb.org/ontologies/sio#Edge_Instruments
:Edge_Instruments rdf:type owl:Class ;
              rdfs:subClassOf [ rdf:type owl:Restriction ;
                           owl:onProperty :vibrating_air ;
                           owl:someValuesFrom :Aerophone
                          ] .

### http://www.semanticweb.org/ontologies/sio#Flute
:Flute rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
```

```
                  owl:onProperty :vibrating_air ;
                  owl:someValuesFrom :Aerophone
                ] .

### http://www.semanticweb.org/ontologies/sio#Oboe
:Oboe rdf:type owl:Class ;
     rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :reeds_no ;
                  owl:hasValue "2"
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :reeds ;
                  owl:someValuesFrom :Reed_Pipe_Instruments
                ] .

### http://www.semanticweb.org/ontologies/sio#Piano
:Piano rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :sound_initiation_process ;
                  owl:hasValue "Struck"
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :vibrating_string ;
                  owl:someValuesFrom :Chordophone
                ] .

### http://www.semanticweb.org/ontologies/sio#Reed_Pipe_Instruments
:Reed_Pipe_Instruments rdf:type owl:Class ;
                rdfs:subClassOf [ rdf:type owl:Restriction ;
```

```
                              owl:onProperty :vibrating_air ;
                              owl:someValuesFrom :Aerophone
                            ] .


### http://www.semanticweb.org/ontologies/sio#Saxophone
:Saxophone rdf:type owl:Class ;
        rdfs:subClassOf [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds ;
                     owl:someValuesFrom :Reed_Pipe_Instruments
                   ] ,
                   [ rdf:type owl:Restriction ;
                     owl:onProperty :reeds_no ;
                     owl:hasValue "1"
                   ] .


### http://www.semanticweb.org/ontologies/sio#Violin
:Violin rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                   owl:onProperty :sound_initiation_process ;
                   owl:hasValue "Bowed"
                 ] ,
                 [ rdf:type owl:Restriction ;
                   owl:onProperty :vibrating_string ;
                   owl:someValuesFrom :Chordophone
                 ] .
```

Listing D.6: OWL representation of $\mathcal{O}_{C6}$ (Figure 7.3d) for MLP using 8 MFCC features and 32 codevectors on the solo music dataset.

# References

A., P. & K., L. (2009). Constructing folksonomies from user-specified relations on flickr. *WWW '09 Proceedings of the 18th international conference on World wide web*. *Cited on page(s):* 121

ABULAISH, M. (2008). *Ontology Engineering For Imprecise Knowledge Management*. Lambert Academic Publishing. *Cited on page(s):* 16

AGOSTINI, G., LONGARI, M. & POLLASTRI, E. (2003). Musical instrument timbres classification with spectral features. *EURASIP Journal on Applied Signal Processing*. *Cited on page(s):* 77

ALLEMANG, D. & HENDLER, J. (2008). *Semantic Web for the Working Ontologists*. Morgan Kaufmann. *Cited on page(s):* 43

ANTONIOU, G. & CAN HARMELEN, F. (2008). *Semantic Web Premier 2nd Edition*. Massachusetts Institute of Technology. *Cited on page(s):* 45

ARABSHIAN, K., DANIELSEN, P. & AFROZ, S. (2012). Lexont: A semi-automatic ontology creation tool for programmable web. *The Association for the Advancement of Artificial Intelligence.*. *Cited on page(s):* 71

ARNDT, R., TRONCY, R., STAAB, S., HARDMAN, L. & VACURA, M. (2007). Comm: Designing a well-founded multimedia ontology for the web. In *Proceedings of the 6th International Semantic Web Conference (ISWC 2007)*, Busan, Korea. *Cited on page(s):* 32

ARTAC, M., JOGAN, M. & LEONARDIS, A. (2002). Incremental pca for on-line visual learning and recognition. *In Proceedings of International Conference on attern Recognition*, **3**, 781–784. *Cited on page(s):* 79

BANDARA, A., PAYNE, T., DE ROURE, D. & CLEMO, G. (2004). An ontological framework for semantic description of devices. *International Semantic Web Conference*. *Cited on page(s):* 32

BARKER, K., CHAUDHRI, V.K., CHAW, S.Y., CLARK, P.E., FAN, J., ISRAEL, D., MISHRA, S., PORTER, B., ROMERO, P., TECUCI, D. & YEH, P. (2004). A question-answering system for ap chemistry: Assesing kr & r technologies. *The Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004)*. *Cited on page(s):* 116

BARTHET, M. & SANDLER, M. (2010a). On the effect of reverberation on musical instrument automatic recognition. In *Proceedings of the 128th Convention*. *Cited on page(s):* 129, 146

BARTHET, M. & SANDLER, M. (2010b). Time-dependent automatic musical instrument recognition in solo recordings. In *7th Int. Symposium on Computer Music Modeling and Retrieval (CMMR 2010)*, 183–194, Malaga (Spain). *Cited on page(s):* 81, 98, 145

BARTHET, M., DEPALLE, P., KRONLAND-MARTINET, R. & YSTAD, S. (2010a). Accoustical correlates of timbre and expressiveness in clarinet performance. *Music Perception*, **28**, 135–153. *Cited on page(s):* 124

BARTHET, M., GUILLEMAIN, P., KRONLAND-MARTINET, R. & YSTAD, S. (2010b). From clarinet control to timbre perception. *Acta Acustica*, **96**, 678–689. *Cited on page(s):* 130

BAUMANN, S., KLUTER, A. & NORLIEN, M. (2002). Using natural language input and audio analysis for a human-oriented mir system. *Web Delivery of Music (WEDELMUSIC)*. *Cited on page(s):* 118

BERNERS-LEE, T. (????). Linked data. *Cited on page(s):* 46

BERNERS-LEE, T., HANDLER, J. & LASSILA, O. (2001). The semantic web. *Scientific American*. *Cited on page(s):* 15

BISHOP, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer, Inc. *Cited on page(s):* 77, 100

BIZER, C. & CYGANIAK, R. (2006). D2r server-publishing relational databases on the semantic web. *5th International Semantic Web Conference*. *Cited on page(s):* 51

BIZER, C., CYGANIAK, R. & HEATH, T. (????). How to publish linked data on the web. *Cited on page(s):* 46

BLASCHKE, C., HIRSCHMAN, L., VALENCIA, A. & YEH, A., eds. (2004). *A critical assessment of text mining methods in molecular biology*. *Cited on page(s):* 117

BLOEHDORN, S., HAASE, P., SURE, Y. & VOELKER, J. (2005). D.6.6.1 report on the integration of ml, hlt and om. Tech. rep., University of Karlsruhe). *Cited on page(s):* 26

BREWSTER, C., ALANI, H., DASMAHAPATRA, S. & WILKS, Y. (2004). Data driven ontology evaluation. *Proceedings of the International Conference on Language Resources and Evaluation*. *Cited on page(s):* 117

BROWN, E. & DEFFENBACHER, K. (1979). Perception and the senses. *Oxford University Press*. *Cited on page(s):* 74

CAMPBELL, K.E., OLIVER, D.E., SPACKMAN, K.A. & SHORTLIFFE, E.H. (1998). Representing thoughts, words, and things in the umls. *Journal of the American Medical Informatics Association*, **5**, 421–431. *Cited on page(s):* 24

CANU, S., GRANDVALET, Y., GUIGUE, V. & RAKOTOMAMONJY, A. (2005). SVM and kernel methods matlab toolbox. Perception Systemes et Information, INSA de Rouen, Rouen, France. *Cited on page(s):* 99

CHANG, Y.W., HSIEH, C.J., CHANG, K.W., RINGGAARD, M. & LIN, C.J. (2010). Low-degree polynomial mapping of data for svm. *Journal of Machine Learning Research*, 1471–1490. *Cited on page(s):* 78

CHÉTRY, N.D. (2006). *Computer Models for Musical Instrument Identification*. Ph.D. thesis, Queen Mary University of London. *Cited on page(s):* 97, 100, 130, 145

CHÉTRY, N.D., DAVIES, M. & SANDLER, M. (2005). Musical instrument identification using LSF and K-means. In *Proc. AES 118th Convention*. *Cited on page(s):* 81, 97, 98

CHOU, C.H., LIU, P.H. & CAI, B. (2008). On the studies of syllable segmentation and improving mfccs for automatic birdsong recognition. *IEEE Asia-Pacific Services Computing Conference*. *Cited on page(s):* 147

CIMIANO, P. (2006). *Ontology Learning and Population from Text Algorithms, Evaluation and Applications*. Springer. *Cited on page(s):* 84, 89

CIMIANO, P., HOTHO, A. & STAAB, S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*. *Cited on page(s):* 70, 83

COHEN, J. (1977). *Statistical power analysis for the behavioral sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates. *Cited on page(s):* 124

COHEN, P., SCHRAG, R., JONES, E., PEASE, A., LIN, A., STARR, B., GUNNING, D. & BURKE, M. (1998). The darpa high-performance knowledge bases project. *AI Magazine*, **19**, 25–49. *Cited on page(s):* 116

CORTES, C. & VAPNIK, V. (1995). Support vector networks. *Machine Learning*, **20**, 273–297. *Cited on page(s):* 77

CROFTS, N., DOERR, M., GILL, T., STEAD, S. & STIFF, M. (2010). Denition of the cidoc conceptual reference model (version 5.0.2). Tech. rep., First published by the ICOM/CIDOC Documentation Standards Group in 2003, continued by the CIDOC CRM Special Interest Group. *Cited on page(s):* 32

DACONTA, M.C., OBRST, L.J. & SMITH, K.T. (2003). *The Semantic Web: A Guide to the Future of XML Web Services, and Knowledge Management*. Wiley Publishing. *Cited on page(s):* 9, 38, 59

DAVULCU, H., VADREVU, S. & NAGARAJAN, S. (2004). Ontominer: Bootstrapping ontologies from overlapping domain specific web sites. *the13th International World Wide Web Conference*. *Cited on page(s):* 70

DAVY, M., DESOBRY, F., GRETTON, A. & DONCARLI, C. (2006). An online support vector machine for abnormal events detection. *Signal Processing*. *Cited on page(s):* 79

DELLSCHAFT, K. & STAAB, S. (2006). On how to perform a gold standard based evaluation of ontology learning. In *Lecture Notes in Computer Science*, vol. 4273, 228–241. *Cited on page(s):* 121

DIXON, S., JACOBSON, K., MESNAGE, C. & NORTON, B. (2011). Linkedbrainz. `http://linkedbrainz.c4dmpresents.org/`. *Cited on page(s):* 49

DOMINGUE, J., FENSEL, D. & HENDLER, J.A., eds. (2011). *Handbook of Semantic Web Technologies.*. Springer. *Cited on page(s):* 40

DRAĞER, H.H. (1948). Prinzip einer systematik der musikinstrumente. *Kassel und Basel:Barenreiter*. *Cited on page(s):* 56

EHRIG, M., HAASE, P., HEFKE, M. & STOJANOVIC, N. (2005). Similarity for ontologies - a comprehensive framework. *European Conference on Information Systems (ECIS)*, 1509–1518. *Cited on page(s):* 119

ELSAYED, A.E., EL-BELTAGY, S.R., RAFEA, M. & HEGAZY, O. (2007). Applying data mining for ontology building. *Proc. of ISSR*. *Cited on page(s):* 83

ELSCHEK, O. (1969). System of graphical and symbolic signs for the typology of aerophones. *Bratislava:VydatelstvS lovenskej Academi Vied.*. *Cited on page(s):* 56

ERONEN, A. (2001). *Automatic musical instrument recognition*. Master's thesis, Tempere University of Technology. *Cited on page(s):* 77

ERONEN, A. & KLAPURI, A. (2000). Musical instrument recognition using cepstral coefficients and temporal features. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 753–756. *Cited on page(s):* 82

ESSID, S., RICHARD, G. & DAVID, B. (2004). Musical instrument recognition on solo performances. *European Signal Processing Conference*. *Cited on page(s):* 100

ESSID, S., RICHARD, G. & DAVID, B. (2006). Musical instrument recognition by pairwise classification strategies. *IEE Transcatsions on Audio, Speech and Language Processing*, **14**, 1401–1412. *Cited on page(s):* 82, 100, 147

EUZENAT, J. (2007). Semantic precision and recall for ontology alignment evaluation. *International Joint Conference on Artificial Intelligence (IJCAI)*, 348–353. *Cited on page(s):* 119

FAGERLUND, S. (2007). Bird species recognition using support vector machines. *EURASIP Journal on Applied Signal Processing*. *Cited on page(s):* 148

FAZEKAS, G. & SANDLER, M. (2007). Structural decomposition of recorded vocal performances and its application to intelligent audio editing. In *in proc. 123rd Convention of the Audio Engineering Society, New York, NY, USA, 2007 October 5–8*. *Cited on page(s):* 73

FAZEKAS, G., RAIMOND, Y., JACOBSON, K. & SANDLER, M. (2010). An overview of semantic web activities in the omras2 project. *Journal of New Music Research special issue on Music Informatics and the OMRAS2 Project*, **39**, 295–311. *Cited on page(s):* 16

FERNANDEZ-LOPEZ, M. & GOMEZ-PEREZ, A. (2002). The integration of ontoclean in webode. *Evaluation of Ontology-based Tools Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002*. *Cited on page(s):* 115

FORTUNA, B., MLADENIC, D. & GROBELNIK, M. (2006). Semi-automatic construction of topic ontologies. *Lecture Notes in Computer Science.*, **4289**. *Cited on page(s):* 72

FREUND, Y. & SCHAPIRE, R.E. (1996). Experiments with a new boosting algorithm. In L. Saitta, ed., *International Conference on Machine Learning*, 148–156, Morgan Kaufmann, Bari, Italy. *Cited on page(s):* 83

FUJINAGA, I., MOORE, S., SULLIVAN, D.S. & JR. (1998). Implementation of exemplar-based learning model for music cognition. *In Proceedings of the International Conference on Music Perception and Cognition*, 171–179. *Cited on page(s):* 77

GANTER, B. (2006). *Finger Exercises in Formal Concept Analysis*. Dresden ICCL Summer School, Technishe Universitat Dresden. *Cited on page(s):* 107

GANTER, B. & WILLE., R. (1989). *Conceptual scaling.*. Springer-Verlag. *Cited on page(s):* 107

GANTER, B., STUMME, G. & WILLE, R. (2005). *Formal Concept Analysis Foundation and Applications*. Springer. *Cited on page(s):* 84

GILL, M.K., KAUR, R. & KAUR, J. (2010). Vector quantization based speaker identification. *International Journal of Computer Applications*, **4**. *Cited on page(s):* 76

GILLET, O. & RICHARD, G. (2004). Automatic transcription of drum loops. *International Conference on Acoustics, Speech, and Signal Processing*. *Cited on page(s):* 78

GILLET, O. & RICHARD, G. (2005). Drum loops retrieval from spoken queries. *Journal of Intelligent Information Systems*. *Cited on page(s):* 78

GODIN, R. & VALTCHEV, P. (2005). *Formal concept analysis-based hierarchy design in object-oriented software development*, 304–323. Springer Berlin Heidelberg. *Cited on page(s):* 83

GOTZMANN, D. & LINDIG, C. (????). `http://code.google.com/p/colibri-java/`, formal Concept Analysis implemented in Java. *Cited on page(s):* 101

GRACIARENA, M., DELPLANCHE, M., SHRIBERG, E., STOLCKE, A. & FERRER, L. (2010). Acoustic front-end optimization for bird species recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 293–296. *Cited on page(s):* 147

GRUBER, T.R. (1993). A translation approach to portable ontology specifications. Tech. rep., Knowledge Systems Laboratory. *Cited on page(s):* 23

GUARINO, N. & WELTY, C. (2000). A formal ontology of properties. *12th International Conference on Knowledge Engineering and Knowledge Management*. *Cited on page(s):* 114

GUARINO, N. & WELTY, C.A. (2004). *Handbook on Ontologies*, chap. An Overview of OntoClean, 151–172. Springer. *Cited on page(s):* 114

GUIZZARDI, G. (2007). On ontology, ontologies, conceptualizations, modeling languages, and (meta)models. In *Proceedings of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference DB&IS'2006*, 18–39, IOS Press, Amsterdam, The Netherlands, The Netherlands. *Cited on page(s):* 23

HAGAN, M.T. & MENHAJ, M.B. (1994). Training feedforward networks with the marquardt algorithm. *Neural Networks, IEEE*. *Cited on page(s):* 100

HAYKIN, S. (1998). *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall. *Cited on page(s):* 77

HEITTOLA, T. (2003). *Automatic Classification of Music Signals*. Master's thesis, Tampere University of Technology. *Cited on page(s):* 74

HELEN, M. & VIRTANEN, T. (2005). Separation of drums from polyphonic music using non-negative matrix factorisation and support vector machine. *In proceedings of Signal Processing Conference (EUSIPCO)*. *Cited on page(s):* 78

HEPP, M. (2005). Representing the hierarchy of industrial taxonomies in OWL:the gen/tax approach. *ISWC Workshop Semantic Web Case Studies and Best Practices for eBusiness (SWCASE05)*. *Cited on page(s):* 59

HEPP, M. & DE BRUIJN, J. (2007). Gentax: A generic methodology for deriving OWL and RDF-S ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. *The Semantic Web: Research and Applications, Lecture Notes in Computer Science*, **4519/2007**, 129–144. *Cited on page(s):* 59

HERRERA, P., YETERIAN, A., YETERIAN, R. & GOUYON, F. (2001). Automatic classification of drum sounds: A comparison of feature selection and classification techniques. In *Proceedings of 2nd International Conference on Music and Artificial Intelligence*, 69–80, Springer. *Cited on page(s):* 83

HOBBS, J.R. & PAN, F. (????). Time ontology in owl. *Cited on page(s):* 32

HOFFMAN, M.D., BLEI, D.M. & COOK, P.R. (2009). Easy as cba: A simple probabilistic model for tagging easy as cba: A simple probabilistic model for tagging music. In *Proceedings of the 10th International Society Proceedings of the 10th International Society for Music Information Retrieval Conference*, International Society for Music Information Retrieval. *Cited on page(s):* 79

HOLM, S. (1989). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, **6**, 65–70. *Cited on page(s):* 124

HORRIDGE, M. & BECHHOFER, S. (2010). University of Manchester and Clark & Parsia LLC and University of Ulm. `http://owlapi.sourceforge.net/`. *Cited on page(s):* 110

JENSEN, K. & ARNSPANG, J. (1999). Binary decision tree classification of musical sounds. In *International Computer Music Conference*, Beijing, China. *Cited on page(s):* 83

KABAL, P. & RAMACHANDRAN, R.P. (1986). The computation of line spectral frequencies using chebyshev polynomials. *IEEE Transactions on Audio Speech Language Processing*, **34**, 1419–1426. *Cited on page(s):* 73

KAMINSKYJ, I. & CZASZEJKO, T. (2005). Automatic recognition of isolated monophonic musical instrument sounds using knnc. *Journal of Intelligent Information Systems*, **24**, 199–221. *Cited on page(s):* 77

KANG, Y.K., HWANG, S.H. & YANG, K.M. (2009). Fca-based conceptual knowledge discovery in folksonomy. *World Academy of Science, Engineering and Technology*. *Cited on page(s):* 84

KAROUI, L., AUDE AUFAURE, M. & BENNACER, N. (2004). Ontology discovery from web pages: Application to tourism. *Knowledge Discovery and Ontologies.*. *Cited on page(s):* 70

KARTOMI, M. (2001). The classification of musical instruments: Changing trends in research from the latenineteenth century, with special reference to the 1990s. *Ethnomusicology*, **45**, 283–314. *Cited on page(s):* 16, 56

KIM, J.H., TOMASIK, B. & TURNBULL, D. (2009). Using artist similarity to propagate semantic using artist similarity to propagate semantic information. In *10th International Society for Music Information Retrieval Conference*. *Cited on page(s):* 79

KITAHARA, T., GOTO, M. & OKUNO, H.G. (2003). Musical instrument identification based on f0-dependent multivariate normal distribution. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 421–424. *Cited on page(s):* 82

KLAPURI, A. & DAVY, M., eds. (2006). *Signal Processing Methods For Music Transcription*. Springer. *Cited on page(s):* 81

KOLOZALI, S., BARTHET, M. & SANDLER, M. (2012). Knowledge management on the semantic web: A comparison of neuro-fuzzy and multi-layer perceptron methods for automatic music tagging. *9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)*, 220–231. *Cited on page(s):* 97

KOSTEK, B. (1999). *Soft Computing in Acoustics: Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics.*. Physica-Verlag. *Cited on page(s):* 77

KROHN, U., DAVIES, N.J. & WEEKS, R. (1999). Concept lattices for knowledge management. *BT Technology Journal*, **17**. *Cited on page(s):* 89

LAMERE, P. (2008). Social tagging and music information retrieval. *Journal of New Music Research*. *Cited on page(s):* 79

LEE, C.H., LEE, Y.K. & HUANG, R.Z. (2006). Automatic recognition of bird songs using cepstral coefficients. *Journal of Information Technology and Applications*, **1**, 17–23. *Cited on page(s):* 147

LEHMANN, J. & BÜHMANN, L. (2010). Ore - a tool for repairing and enriching knowledge bases. *ISWC 2010: The 9th International Semantic Web Conference*. *Cited on page(s):* 71

LEVENSHTEIN, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, **10(8)**, 707–710. *Cited on page(s):* 118

LEVY, M. & SANDLER, M. (2006). Lightweight measures for timbral similarity of musical audio. *In Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*. *Cited on page(s):* 50

LEVY, M. & SANDLER, M. (2007). Signal-based music searching and browsing. *ICCE 2007, International Conference on Consumer Electronics*. *Cited on page(s):* 50

LEVY, M. & SANDLER, M. (2009). Music information retrieval using social tags and audio. *IEEE Transactions on Multiedia.*. *Cited on page(s):* 79

LI, T. & OGIHARA, M. (2004). Content-based music similarity search and emotion detection. In *Acoustics, Speech, and Signal Processing, Proceedings. (ICASSP '04).*. *Cited on page(s):* 80

LI, T., OGIHARA, M. & LI, Q. (2003). A comparative study on content-based music genre classification. *Proceedings of the The 26th ACM/SIGIR International Symposium on Information Retrieval*. *Cited on page(s):* 74

LINDE, Y., BUZO, A. & GRAY, R.M. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, **28**, 702–710. *Cited on page(s):* 98

LOZANO-TELLO, A. & GOMEZ-PEREZ, A. (2004). Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, **15**. *Cited on page(s):* 114

LU, L., ZHANG, H.J. & LI, S.Z. (2003). Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, 482–492. *Cited on page(s):* 78

LUC, L., LIU, D. & ZHANG, H.J. (2006). Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio Speech Language Processing*, **14**. *Cited on page(s):* 74, 80

LYCAN, W.G. (2006). *The Blackwell Guide to the Philosophy of Language*, chap. 14, 255–273. Blackwell Publishing Ltd. *Cited on page(s):* 24

LYSLOFF, R.T.A. & MATSON, J. (1985). A new approach to the classification of sound-producing instruments. *Ethnomusicology*, **29**, 213–236. *Cited on page(s):* 56

MACKAY, D.J.C. (2003). *Information Theory, Inference, and Learning Algorithms.*. Cambridge University Press. *Cited on page(s):* 75

MAEDCHE, A. & STAAB, S. (2001). Ontology learning for the semantic web. In *IEEE Intelligent Systems*, 72–79. *Cited on page(s):* 84, 118

MAEDCHE, A. & STAAB, S. (2002). Measuring similarity between ontologies. *Engineering and Knowledge Management: Ontologies and the Semantic Web*, **2473/2002**, 15–21. *Cited on page(s):* 118, 121, 122

MARSLAND, S. (2009). *Machine Learning: An Algorithmic Perspective*. Chapman and Hall/CRC. *Cited on page(s):* 75

MARTIN, K.D., SCHEIRER, E.D. & VERCOE, B.L. (1998). Music content analysis through models of audition. *In Proc. 1998 ACM Multimedia Workshop on Content Processing of Music for Multimedia Applications*. *Cited on page(s):* 74, 82

MATTHEWS, B. (2005). Semantic web technologies. *JISC Technology and Standards Watch*. *Cited on page(s):* 40

MERHAV, N. & LEE, C. (1990). On the asymptotic statistical behavior of empirical cepstralcoefficients. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, **41**. *Cited on page(s):* 74

MITCHELL, T.M. (1997). *Machine Learning*. McGraw Hill series in computer science, McGraw Hill. *Cited on page(s):* 75

MONTAGU, J. & BURTON, J. (1971). A proposed new system for musical instruments. *Society for Ethnomusicology*, **15**, 49–70. *Cited on page(s):* 59

MORRIS, J. (2007). The role of ontology in modern expert systems development. *Cited on page(s):* 37

NACÉRA BENNACER, L.K.N.B.L.K.N.B.L.K.N.B.L.K.N.B., LOBNA KAROUI NACÉRA BENNACER & KAROUI, L. (2005). A framework for retrieving conceptual knowledge from web pages. *Semantic Web Applications and Perspectives (SWAP)*. *Cited on page(s):* 70

NOY, N.F., FERGERSON, R.W. & MUSEN, M.A. (2000). The knowledge model of protege-2000: Combining interoperability and flexibility. *The 12th International Conference on Knowledge Engineering and Knowledge Management*. *Cited on page(s):* 115

OBRST, L., ASHPOLE, B., CEUSTERS, W., MANI, I., RAY, S. & SMITH, B. (2007). *Revolutionizing Knowledge Discovery in the Life Sciences*, chap. THE EVALUATION OF ONTOLOGIES, 139–158. 7, Springer. *Cited on page(s):* 119

OGDEN, C.K. & RICHARDS, I.A. (1923). *The Meaning of Meaning: a study of the influence of language upon thought and of the science of symbolism*. Harcourt, Brace & World, Inc. Reissue edition (1946), 8th edn. *Cited on page(s):* 9, 25

OLSON, H.F. (1967). *Music, Physics and Engineering*. Dover Publications. *Cited on page(s):* 62

PALIWAL, K.K. & ATAL, B.S. (1993). Efficient vector quantization of lpc parameters at 24 bits/frame. *IEEE Transactions on Speech and Audio Processing*. *Cited on page(s):* 76

PARK, T.H. (2004). *Towards Automatic Musical Instrument Timbre Recognition*. Ph.D. thesis, Princeton University. *Cited on page(s):* 77

PEETERS, G. (2003). Automatic classification of large musical instrument databases using hierarchical clssifiers with inertia ratio maximization. *Audio Engineering Society 115th Int. Conv.*. *Cited on page(s):* 83

PEETERS, G. & XAVIER (2003). Hierarchical gaussian tree with intertia ratio maximization for the classification of large musical instrument databases. *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFX-03)*. *Cited on page(s):* 82

PELECANOS, J., MYERS, S., SRIDHARAN, S. & CHANDRAN, V. (2000). Vector quantization based gaussian modeling for speaker verification. In *15th International Conference on Pattern Recognition*, 294–297, IEEE. *Cited on page(s):* 76

PORZEL, R. & MALAKA, R. (2004). A task-based approach for ontology evaluation. *The ECAI Workshop on Ontology Learning and Population*. *Cited on page(s):* 116

RAIMOND, Y. (2008). *A Distributed Music Information System*. Ph.D. thesis, Queen Mary University of London. *Cited on page(s):* 28, 42, 43, 57, 117

RAIMOND, Y., ABDALLAH, S., SANDLER, M. & GIASSON, F. (2007). The music ontology. *7th International Conference on Music Information Retrieval*. *Cited on page(s):* 16

RICHARD, M. (2006). *The Blackwell Guide to the Philosophy of Language*, chap. 10, 186–211. Blackwell Publishing Ltd. *Cited on page(s):* 24

RICHTER, M.M. (1992). Classification and learning of similarity measures. Tech. rep., Fachbereich Informatik (Technische Universität Darmstadt). *Cited on page(s):* 119

ROSENBERG, A.E. & SOONG., F.K. (1986). Evaluation of a vector-quantization talker recognition system in text independent and text dependent modes. In *International Conference on Acoustics, Speech and Signal Processing*. *Cited on page(s):* 76

ROSS, W.D. *et al.* (1924). *Aristotle's metaphysics*, vol. 2. Clarendon Press. *Cited on page(s):* 24

SABOU, M., WROE, C., GOBLE, C. & MISHNE, G. (2005a). Learning domain ontologies for web service descriptions: an experiment in bioinformatics. *14th International World Wide Web Conference (WWW2005)*. *Cited on page(s):* 70, 118

SABOU, M., WROE, C., GOBLE, C. & STUCKENSCHMIDT, H. (2005b). Learning domain ontologies for semantic web service descriptions. *Journal of Web Semantics*, **3**. *Cited on page(s):* 118

SANCHEZ, D. & MORENO, A. (2004). Creating ontologies from web documents. *Artificial Intelligence Research And Development*, **113**, 11–18. *Cited on page(s):* 69

SAUERMANN, L. & CYGANIAK, R. (????). Cool uris for the semantic web. *Cited on page(s):* 46

SCHUTZ, A. & BUITELAAR, P. (2005). Relext: A tool for relation extraction from text in ontology extension. *The 4th International Semantic Web Conference (ISWC)*. *Cited on page(s):* 72

SIGURDSSON, S., PETERSEN, K.B. & LEHN-SCHILER, T. (2006). Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music. *ISMIR 2006 7th International Conference on Music Information Retrieval*. *Cited on page(s):* 51

SMITH, B. (2004). *The Blackwell Guide to the Philosophy of Computing and Information*, chap. 11, 155–166. Blackwell Publishing Ltd. *Cited on page(s):* 24

SNELTING, G. (2003). Concept lattices in software analysis. In *International Conference on Formal Concept Analysis*, 272–287. *Cited on page(s):* 83

SOONG, F.K., ROSENBERG, A.E., RABINER, L.R. & JUANG, B.H. (1985). A vector quantization approach to speaker recognition. *International Conference on Acoustics, Speech and Signal Processing*, **10**. *Cited on page(s):* 76

SOWA, J.F. (1999). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Course Technology Inc. *Cited on page(s):* 24, 25

STAAB, S. & STUDER, R. (2009). *Handbook on Ontologies.*. Springer, 2nd edn. *Cited on page(s):* 37, 40

STEELANT, D.V., TANGHE, K., DEGROEVE, S., BAETS, B.D., LEMAN, M., MARTENS, J.P. & P. MARTENS, J. (2004). Classification of percussive sounds using support vector machines.

*In Proceedings of the Annual Machine Learning Conference of Belgium and The Netherlands*, 146–152. *Cited on page(s):* 78

STOWELL, D. & PLUMBLEY, M.D. (2011). Birdsong and c4dm: A survey of uk birdsong and machine recognition for music researchers. Tech. rep., Centre for Digital Music, Queen Mary, University of London. *Cited on page(s):* 148

STUMME, G., WILLE, R. & WILLE, U. (1998). Conceptual knowledge discovery in databases using formal concept analysis methods. *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*. *Cited on page(s):* 84

SURE, Y., ANGELE, J. & STAAB, S. (2003). Ontoedit: Multifaceted inferencing for ontology engineering. *Journal on Data Semantics*. *Cited on page(s):* 115

TIDHAR, D., FAZEKAS, G. & KOLOZALI, S. (2009). Publishing music similarity features on the semantic web. *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*. *Cited on page(s):* 49, 50

TROHIDIS, K., TSOUMAKAS, G., KALLIRIS, G. & VLAHAVAS, I. (2008a). Multi-label classification of music into emotions. *ISMIR 2008 9th International Conference on Music Information Retrieval*. *Cited on page(s):* 80

TROHIDIS, K., TSOUMAKAS, G., KALLIRIS, G. & VLAHAVAS, I. (2008b). Multi-label classification of music into emotions multi-label classification of music into emotions multi-label classification of music into emotions multi-label classification of music into emotions. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*. *Cited on page(s):* 80

TURNBULL, D., BARRINGTON, L., TORRES, D. & LANCKRIET, G. (2008). Semantic annotation and retrieval semantic annotation and retrieval of music and sound effects. In *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 16. *Cited on page(s):* 80

TZANETAKIS, G. (2002). *Manipulation, Analysis and Retrieval Systems for Audio Signals*. Ph.D. thesis. *Cited on page(s):* 74

VAN RIJSBERGEN, C.J. (1979). *INFORMATION RETRIEVAL*. Butterworth-Heinemann, 2nd edn. *Cited on page(s):* 119

VASUMATHI, D. & GOVARDHAN, D.A. (2009). Efficient web usage mining based on formal concept analysis. *Journal of Theoretical and Applied Information Technology*. *Cited on page(s):* 84

VINCENT, E. & RODET, X. (2004). Instrument identification in solo and ensemble music using independent subspace analysis. *International Conference on Music Information Retrieval*. *Cited on page(s):* 81

VOLKER, J., VRANDECIC, D. & SURE, Y. (2005). Automatic evaluation of ontologies (aeon). *International Semantic Web Conference*, 716–731. *Cited on page(s):* 115

VON HORNBOSTEL, E.M. & SACHS, C. (1914). Systematik der musikinstrumente: Einversuch. zeitschriftffir ethnologie, translated by a. baines and k. wachsmann as a classification of musical instruments. *Galpin Society Journal.*. *Cited on page(s):* 56

VRANDECIC, D. (2010). *Ontology Evaluation*. Ph.D. thesis, The Karlsruhe Institute of Technology (KIT). *Cited on page(s):* 119

WIECZORKOWSKA, A. (1999). Classification of musical instrument sounds using decision trees. In *International Symposium on Sound Engineering and Mastering*, 225–230. *Cited on page(s):* 83

WILLE, R. (1982). Restructuring lattice theory: an approach based on hierarchies of concepts. . *I. Rival (Ed.), Ordered sets. Reidel, Dordrecht-Boston.*, 445–470. *Cited on page(s):* 83

WONG, W., LIU, W. & BENNAMOU, M. (2009). Acquiring semantic relations using the web for constructing lightweight ontologies. In *The 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. *Cited on page(s):* 71

YAHIA, A., LAKHAL, L., BORDAT, J.P. & CICCHETTI, R. (1996). An algorithmic method for building inheritance graphs in object database design. In *15th International Conference on Conceptual Modelings*, 422–437, Springer. *Cited on page(s):* 83

YOUNG, S., WOODLAND, P. & BYRNE, W. (1993). Htk: Hidden markov toolkit. *J. Audio Engineering Society*, **1**. *Cited on page(s):* 74