TECHNISCHE
UNIVERSITÄT
WIEN

VIENNA
UNIVERSITY OF
TECHNOLOGY

MASTERARBEIT

ATLANTIS

Or Towards a Multi-Modal Approach to Music Information Retrieval and its Visualisation

Ausgeführt am Institut für
Softwaretechnik und Interaktive Systeme
der Technischen Universität Wien

unter der Anleitung von
Ao.Univ.Prof. Dipl.Ing. Dr. Andreas Rauber
Favoritenstraße 9-11 / 188
A - 1040 Vienna, Austria

durch
Mag. Robert Neumayer
Baumgasse 6, 2404, Haslau

October 15, 2007

Zusammenfassung

Versierte Verfahren zur Organisation von Musikkollektionen bilden die Grundlage für eine Vielzahl von Anwendungen. Hier wird besonders auf vorhandene Probleme eingegangen, es werden bestehende Techniken und deren Unzulänglichkeiten beschrieben, aber auch alternative Benutzerschnittstellen fuer Musikarchive und darauf aufbauend neue Moeglichkeiten zur Interaktion erklaert. Dabei wird besonders auf *Self-Organising Maps*, selbstorganisierende Neuronale Netzwerke zum Clustering von hochdimensionalen Daten, und ihre Verwendbarkeit für Musikorganisation diskutiert. Um der vielseitigen, oft zu komplexen Information, die in Musikdaten stecken kann, gerecht zu werden, werden Datenbeschreibungen, die über traditionelle Repräsentationen hinausgehen, untersucht. Traditionell verwendet die Music Information Retrieval Community auf Signalverarbeitung aufbauende Merkmalssets für Audiodaten. In dieser Arbeit wird vor allem auf textbasierte Features und deren Informationsgehalt in Bezug auf Diskriminanz zwischen Genres eingegangen. Außerdem werden die Möglichkeiten untersucht, die sich für kombinierte Empfehlung von ähnlichen Songs ergeben. Dabei wird der Einfluss von Genre-, Artist- und Albenbeschreibungen auf die Musikempfehlungen untersucht. Weiters wird ein neuer Ansatz zur Visualisierung von multimodalen Repräsentationen für Audio beschrieben. Eine Audiokollektion kann demnach nach verschiedenen Repräsentationen geclustert werden: Audiofeatures und Textfeatures auf Basis von Song Lyrics. Die entstehenden Clusterings werden graphisch aufbereitet und mittels eines Sets von Kennzahlen verglichen.

Abstract

Various aspects of the organisation of media archives and collections have produced eager interest in recent years. The Music Information Retrieval community has been gaining many insights into the area of abstract representations of music by means of audio signal processing. On top of that, recommendation engines are built to provide novel ways of creating playlists based on users' preferences. Another important application of audio representation is automatic genre categorisation, i.e. the automatic assignment of genre tags to untagged audio files. However, for many applications representation based on audio features only do not contain enough information. A song's lyrics often describe its genre better than what it sounds like, e.g. 'Christmas carols' or 'love songs'. Therefore, approaches for the combination of additional data like song lyrics, artist biographies, or album reviews for music recommendation are examined. Further, the application of the *Self-Organising Map* for clustering, i.e. the mapping from the resultant high-dimensional feature spaces onto two-dimensional maps, for explorative analysis of audio collections with respect to multi-modal feature sets is investigated (audio / text). Additionally, a new visualisation for simultaneous display of multi-modal clusterings as well as cluster validation metrics are presented. Finally, a short overview and outlook on future work is given.

The universe is perfect.
You cannot improve it.
If you try to change it,
you will ruin it.
If you try to hold it,
you will lose it.

Notes to Odo Chan, CY 9191

Credits go to Andromeda – for brilliant quotes like this one[1][2].

[1] http://www.andromedatv.com/
[2] http://en.wikiquote.org/wiki/Andromeda

Contents

1 Introduction **8**

2 Main Principles and Underlying Technologies **14**

2.1 Music Information Retrieval 14

2.2 Introduction to Text Information Retrieval 16

2.3 Term Weighting in Information Retrieval 17

2.4 Feature Selection and Dimensionality Reduction 18

2.4.1 Document Frequency Thresholding 19

2.4.2 Information Gain 19

2.5 Audio Features 20

2.6 *Self-Organising Map* 22

2.7 Cluster Validation Techniques 23

2.7.1 Unsupervised Cluster Validation 24

2.7.2 Supervised Cluster Validation 27

2.8 Cluster Validation for *Self-Organising Maps* 29

2.8.1 Adaption of the Silhouette Value to the *Self-Organising Map* . . 30

2.9 Interfaces Based on the *Self-Organising Map* 31

2.10 Machine Learning Techniques 33

2.11 Recap 33

3 Test Collections and Multi-Modal Audio Indexing 34

3.1 Test Collections 36

3.1.1 Small Collection 36

3.1.2 Large Collection 37

3.2 Automated Enrichment and Indexing Techniques 39

3.3 Recap 41

4 Multi-Modality in Music Information Retrieval 42

4.1 Ranking Merging - Integrating Retrieval Results 44

4.1.1 Missing Values 45

4.1.2 Recap 48

4.2 Multi-Modal Visualisation of *SOM* Clusterings 49

4.2.1 A First Prototype 50

4.2.2 Cluster Validation for Multi-Modal Clusterings 52

4.2.3 Recap . . . 58

4.3 Multi-Modal Genre Classification . . . 59

4.4 Where Do We Go from Here . . . 59

5 Implementation Details 60

5.1 *Atlantis* . . . 60

5.1.1 Packages of Particular Interest . . . 61

5.1.2 Database Binding . . . 61

5.1.3 Internet Text Mining . . . 62

5.1.4 Feature Selection . . . 62

5.1.5 Import Export Component . . . 67

5.1.6 Typical *Atlantis* Usage . . . 68

5.2 *Sovis* (*Self-Organising Map* Visualisation) . . . 71

5.3 Recap . . . 74

6 Experiments 77

6.1 Small Collection Experiments . . . 78

6.1.1 Clustering According to Audio Features . . . 78

6.1.2 Clustering According to Lyrics Features . . . 80

6.1.3 Combined, Multi-Modal Visualisation . . . 80

6.2 Large-Scale Experiments 81

6.2.1 Multi-Modal Audio Similarity Ranking 83

6.2.2 Comparisons of Multi-Modal Clusterings 91

6.2.3 Musical Genre Classification 97

6.3 Recap 101

7 Conclusions and Future Work **102**

Chapter 1

Introduction

> The true quarry of any great adventurer is the undiscovered territory of their own soul.
> Lady Aenea Makros, "The Metaphysics of Motion" CY 6416

Text Information Retrieval deals with the automatic retrieval of (text) documents. Its main task is to automatically extract machine-readable representations, so-called features from all kinds of text documents. These features can subsequently be used for key word as well as content-based and similarity search by a transformation to a vector or matrix representation. Music Information Retrieval (MIR) is an area of Information Retrieval which is concerned with the application of its methods to musical data sources. In this context it does not only mean the sole audio signal of a piece of music but also its associated metadata as well as additional information, which could, for instance, be fetched or mined from the Internet.

The large-scale adaption of new business models for digital content including audio material is already happening. Online music stores are gaining market shares, driving the need for online music retailers to provide adequate means of access to their catalogues. Their ways of advertising and making accessible their collections are often limited, be it by the sheer size of their collections, by the dynamics with which new

titles are being added and need to be filed into the collection organisation, or by inappropriate means of searching and browsing it. What many content providers and online music vendors are still missing are appropriate means of presenting their media to their users. Amazon[1] or last.fm[2] have shown the potential of recommendation engines based on data mining in transactional data. Those recommendation engines have impressively shown the potential and merits of suggesting users new items in numerous online shopping and other community centred applications. Private users' requirements coincide because their collections are growing significantly as well. The steadily increasing success of online stores like iTunes[3] or Magnatune[4] brings digital audio closer to end users, creating a new application field for Music Information Retrieval. Many private users have a strong interest in managing their collections efficiently and being able to access their music in diverse ways. Musical genre categorisation based on e.g. meta tags in audio files often restricts users to the type of music they are already listening to, i.e. browsing genre categories makes it difficult to discover 'new' types of music. The mood a user is in often does not follow genre categories; personal listening behaviours often differ from predefined genre tags. Thus, recommending users similar songs to ones they are currently listening to or like is one of Music Information Retrieval's main tasks. Technologies related to similarity retrieval, however, have to be adapted to be used in the music context. The online shops of music retailers are increasingly popular places for buying music, creating a big market for music recommendation engines. Suggesting customers similar songs is a key factor in being a successful music retailer and new ways of presenting one's collection to customers is a vital aspect of entering or staying in the market.

Furthermore, it is an intrinsic need for every Music Information Retrieval system to include not only recommendation or playlist generation engines, but also possibilities to search and browse a music repository. Content-based access to music has proven to be an efficient means of overcoming traditional metadata categories, as shown by

[1] http://www.amazon.com
[2] http://www.last.fm
[3] http://www.apple.com/au/itunes/store/
[4] http://www.magnatune.com

benchmarking initiatives like the Music Information Retrieval Evaluation eXchange (MIREX) [28]. To achieve this, signal processing techniques are used to extract features from audio files capturing characteristics such as rhythm, melodic sequences, instrumentation, timbre. These are feasible input both for automatic genre classification of music as well as for alternative organisations of audio collections like their display via map based, two-dimensional interfaces [32].

Similarity, however, is not only defined by individual hearing sensation but also, to a large degree, by cultural or community information which offers a far richer and more diverse source of information. Particularly song lyrics and other cultural information are feasible means for searching these collections. Rather than searching for songs that sound similar to a given query song, users often are more interested in songs that cover similar topics, such as 'love songs', or 'Christmas carols', which are not acoustic genres per se, i.e. songs about these particular topics might cover a broad range of musical styles. Similarly, the language of the lyrics often plays a decisive role in perceived similarity of two songs as well as their inclusion in a given playlist. Even advances in audio feature extraction will not be able to overcome fundamental limitations of this kind. Song lyrics therefore play an important role in music similarity. This textual information offers a wealth of additional information to be included in music retrieval tasks that may be used to complement both acoustic as well as metadata information for pieces of music.

Sometimes, finding a similar Album is more important than finding songs that sound similar. Many users may rather be interested in songs that cover similar topics than sound alike. Artist similarity may be of great help when users try not only to find new songs, but are interested in new bands or concerts of these bands. Textual artist descriptions define similarity by a whole new range of aspects too. There are dimensions of artist similarity that can never be covered by audio features only, for instance the fact that split-up bands and their successors may play very different kinds of music, yet they may still be similar to each other (they once belonged to the same band after all). Another aspect very particular to artist descriptions is its property

of taking into account geographical information, e.g. bands from the same city or country may be grouped together. Therefore, a text mining component is very suitable to provide additional data and thereby achieve different levels of audio description. To the ends of a more comprehensive model of musical similarity, methods to gather and aggregate multiple levels of text descriptions are investigated and similarity retrieval is based on these data in this thesis.

Browsing metadata hierarchies by tags like 'Artist' and 'Genre' might be feasible for a limited number of songs, but gets increasingly complex and confusing for collections of larger sizes that have to be tendered for manually. Hence, a more comprehensive approach for the organisation and presentation of audio collections is required. Therefore, the visualisation of high-dimensional data itself and, more importantly, its internal structure, poses a big challenge too. Aggregation techniques for very large music collections being described by an even higher-dimensional vector representation are needed. To address this issue, visualisation techniques will be introduced based on the *Self-Organising Map*.

Having all of these points in mind, the main topics covered in this thesis are:

Musical Similarity Recommendation based on multi-modal Music Information Retrieval, i.e. the integration of artist, album, and genre descriptions as well as song lyrics and audio features in similarity ranking methods.

Multi-Modal Clusterings and Their Evaluation will be explained in greater detail. The importance and relevance of lyrics to the visual organisation of songs in large audio collections is going to be identified as well. It is firstly suggested to cluster complex audio data on two-dimensional maps, using the *Self-Organising Map* clustering algorithm. Clustering will be done according to audio as well as lyrics features. Furthermore, quality measures for the two resultant clusterings are proposed and experimentally evaluated on two parallel corpora of both audio and text (lyrics) files.

Musical Genre Classification using both song lyrics and audio features. The combination of both textual as well as audio information for music genre classification, i.e. automatically assigning musical genres to tracks based on audio features as well as content words in song lyrics, is chosen due to feasible results in similarity recommendation. Experimental results will evince the impact on classification accuracy. Parts of the work presented and relied on in this thesis have been presented at or published in the context of international conferences. Particularly the automatic processing and exploitation of song lyrics has been a pressing research topic.

First prototypes for map based applications on mobile devices were presented as a poster at the 6th International Conference on Music Information Retrieval (ISMIR'05) in London, United Kingdom [32]. An overview paper on map based user interfaces was presented at the 1st Workshop on Visual and Multimedia Digital Libraries (VMDL'07), a workshop organised in the course of the International Conference on Image Analysis and Processing (ICIAP'07) in Modena, Italy [33]. The summary paper on the experiments on musical genre classification were accepted for a poster presentation at the 29th European Conference on Information Retrieval (ECIR'07) in Rome, Italy [34]. Further, the multi-modal cluster evaluation and visualisation was accepted for a presentation at the tri-annual Recherche d'Information Assistée par Ordinateur (RIAO'07) conference in Pittsburgh, Pennsylvania, United States of America [35]. Finally, a book chapter contribution about multi-modal audio analysis was accepted for the forthcoming 'Multimodal Processing and Interaction' book to be published in the context of the EU's FP6 project 'Multimedia Understanding through Semantics, Computation and Learning' (MUSCLE).

The remainder of this thesis is organised as follows. Section 2 gives an overview of previous work in the field and relevant basics as well as it introduces feature sets used in subsequent experiments.

In Chapter 3, we then describe audio test collections and data sources, i.e. the automated indexing and textual enrichment of the songs in these collections.

Then, Chapter 4 theoretically presents the main contributions to the field made in this thesis, namely the combination of several levels of text data and audio representations for the basic Music Information Retrieval tasks of similarity ranking, visualisation, and musical genre classification. Furthermore, a quantitative evaluation of multi-modal clusterings is proposed.

Then, Chapter 5 presents the *Atlantis* and *Sovis* application which implement prototypes for both multi-modal similarity ranking and visualisation in greater detail.

Further, Chapter 6 the visualisation method is experimentally validated. Finally, in Chapter 7 conclusions are drawn as well as a short outlook is given.

Chapter 2

Main Principles and Underlying Technologies

> Those who fail to learn history are doomed to repeat it. Those who fail to learn history correctly – why they are simply doomed"
> Achem Dro'hm, "The Illusion of Historical Fact, CY 4971

This chapter gives an overview about relevant (sub-)disciplines and the techniques used later on. This work incorporates methods from several areas, the most important ones being Information Retrieval, more specifically Music Information Retrieval and *Self-Organising Maps* for clustering and visualisation.

2.1 Music Information Retrieval

The area of Music Information Retrieval has been heavily researched, particularly focussing on audio feature extraction. Comprehensive overviews of Music Information Retrieval are given in [8, 36], first experiments based on and an overview of content-based Music Information Retrieval were reported in [9] as well as [52, 53], the focus

being on automatic genre classification of music. In this work a modified version of the *Rhythm Patterns* features is considered, previously used within the SOMeJB system [45]. Based on that feature set, it is shown that the *Statistical Spectrum Descriptors* yield relatively good results at a manageable dimensionality of 168 as compared to the original *Rhythm Patterns* that comprise 1440 feature values [18]. In the remainder of this thesis *Statistical Spectrum Descriptors* are used as audio feature set and improvements in similarity ranking are based thereon. Another example of a set of feasible audio features is implemented in the Marsyas system [52].

In addition to features extracted from audio, several researchers have started to utilise textual information for music IR. A sophisticated semantic and structural analysis including language identification of songs based on lyrics is conducted in [23]. Artist similarity is defined based on song lyrics in [19]. It is also pointed out that similarity retrieval using lyrics is inferior to acoustic similarity, but a combination of lyrics and acoustic similarity could improve results. A powerful approach targeted at large-scale recommendation engines is lyrics alignment for automatic retrieval as presented in [13]. Therein, lyrics are fetched via the automatic alignment of the results obtained by Google queries.

A comprehensive evaluation of additional features is undertaken in [40]. This work takes into account rhyme and style features and shows their impact on classification accuracy for the genre categorisation task in addition to content-based methods.

Artist similarity based on co-occurrences in Google results is studied in [50], creating prototypicality artist/genre rankings, again, showing the importance of text data.

A combined similarity metric for multi-level combination of artist and lyrics retrieval results is presented in [4], which the approach presented in Chapter 3 combination will be based on. It is also outlined in how far the perception of music can be regarded a socio-cultural product. Different aspects like year, genre, or tempo of a song are taken into account in [55]. Those results are then combined and a user evaluation of different weightings is presented and shows that user control over the weightings can lead to

easier and more satisfying playlist generation.

The importance of browsing and searching as well as their combination is outlined in [6]. This work tries to improve those aspects, a combination approach can improve both of them by satisfying users' information needs through offering advanced search capabilities and improving the the recommendations' quality.

2.2 Introduction to Text Information Retrieval

In classic text categorisation low-level features are computed from a labelled training set of sufficient size. New documents can be assigned to the class represented by the most 'similar' documents in terms of word co-occurrences.

An introduction to Information Retrieval as such is given in [49]. The basic idea is to treat text as a bag of words or tokens. This form of IR abstracts from any kind of linguistic information and is often referred to as statistical natural language processing (NLP). Documents are represented as term vectors. A document collection containing the following two documents:

```
This is a text document.
```

and

```
And so is this document a text document.
```

would represent its documents by a vector of length 7, the number of distinct tokens over all documents. Of course, the tokenisation process makes a difference here, if, e.g., spaces were counted as separate tokens, the vector would be of size 8. Models for text representation range from lists of whole words to vectors of *n-grams* (i. e., tokens of size n). Tokenisation may include stemming, i. e., stripping off affixes of words leaving

Table 2.1: Text indexing by example. Tokens are displayed horizontally, different documents are shown row-wise. The token's occurrences make out the numbers in the table

Document/Token	**this**	**is**	**a**	**text**	**document**	**and**	**so**
1	1	1	1	1	1		
2	1	1	1	1	2	1	1
Document frequency	2	2	2	2	2	1	

only word stems. It is very common to use lists of stop words, i. e., static, predefined lists of words that are removed from the documents before further processing (see [24] or *ranks.nl*[1] for a sample list of English stop words). The vectors are shown in detail in Table 2.1.

This representation is subsequently used to calculate distances between or similarities of documents in the vector space; throughout this thesis we rely on the Euclidean distance, given for the distance between two vectors x_i and x_j of dimensionality D in Equation 2.1:

$$d_F(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^{D}(x_i^k - x_j^k)^2}. \tag{2.1}$$

It is defined by the length of the straight line connecting points $\mathbf{x}_i$ and $\mathbf{x}_j$. For a discussion of this problem and general limitations of the Euclidean Distance, see for instance [17, 1].

2.3 Term Weighting in Information Retrieval

Once a text is represented by tokens, more sophisticated techniques can be applied. In the context of a vector space model a document is denoted by d, a term (token) by t, and the number of documents in a corpus by N.

[1] `http://www.ranks.nl/tools/stopwords.html`

The number of times term t appears in document d is denoted as the *term frequency* $tf(t,d)$, the number of documents in the collection that term t occurs in is denoted as *document frequency* $df(t)$, as shown in Table 2.1. The process of assigning weights to terms according to their importance or significance for the classification is called "term-weighting". The basic assumptions are that terms that occur very often in a document are more important for classification, whereas terms that occur in a high fraction of all documents are less important. The most common weighting is referred to as *term frequency* $\times$ *inverse document frequency* [48], where the weight $tf \times idf$ of a term in a document is given in Equation 2.2:

$$tfidf(t,d) = tf(t,d) * ln(N/df(t)) \tag{2.2}$$

This results in vectors of weight values for each document d in the collection. Based on such vector representations of documents, classification methods can be applied. This favours higher weights to less frequent terms.

2.4 Feature Selection and Dimensionality Reduction

When tokenising text documents, one often faces very high dimensional data. Tens of thousands of dimensions are not easy to handle, therefore feature selection plays a significant role. Document frequency thresholding achieves reductions in dimensionality by excluding terms having very high or very low document frequencies. Terms that occur in almost all documents in a collection do not provide any discriminating information. It is similar for terms that have a very low document frequencies, although those features might be helpful if they are not distributed evenly across classes. If a term has a low document frequency it can still help to discriminate genres if it only occurs in for example 'Rock' song lyrics.

Several methods ranging from simple ones relying solely on frequency counts of terms to more sophisticated ones estimating the entropy of terms for specific class

distributions may be employed, which are briefly described below.

2.4.1 Document Frequency Thresholding

Document frequency thresholding is a feasible feature selection for unlabelled data for not taking into account a priori class information. The basic assumption here is that very frequent terms are less discriminative to distinguish between classes (a term occurring in every single instance of all classes would not contribute to differentiate between them and therefore can safely be omitted in further processing). The largest number of tokens, however, occurs only in a very small number of documents. The biggest advantages of document frequency thresholding is that there is no need for class information and it is therefore mainly used for clustering applications. Besides, document frequency thresholding is far less expensive in terms of computational power. In this context that technique is used for dimensionality reduction for clustering and to compare the classification results obtained by the more sophisticated approaches. The document frequency thresholding is followed as follows:

- At first the upper threshold is fixed around .5 - .8, hence all terms that occur in more than 50 to 80 per cent of the documents are omitted
- The lower boundary is dynamically changed as to achieve the desired number of features, removing, e.g., terms that appear in less than 5-10 documents, i.e. have a document frequency lower than 5 or 10

2.4.2 Information Gain

Information Gain (IG) is a technique originally used to compute splitting criteria for decision trees. Different feature selection models including Information Gain are described in [58]. The basic idea behind IG is to find out how well each single feature separates the given data set. Information Gain makes use of class information to iden-

tify the most discriminant features.

The overall entropy I for a given dataset S is computed in Equation 2.3.

$$I = -\sum_{i=1}^{C} p_i log_2 p_i \tag{2.3}$$

where C denotes the available classes and p_i the proportion of instances that belong to one of the i classes. Now the reduction in entropy or gain in information is computed for each attribute or token.

$$IG(S, A) = I(S) - \sum_{v \epsilon A} \frac{|S_v|}{|S|} I(S_v) \tag{2.4}$$

where v is a value of attribute A and S_v the number of instances where A has that value. For instance, if the attribute in question is a token, v could either comprise all occurring values for that term's $tf \times idf$ weighting or simply whether it is present in a document or not, i.e. it can be assumed to be a binary value. $S_{v=0}$ therefore is the number of instances where attribute A has the value 0 or the number of documents that do not include that token.

This results in an Information Gain value for each token extracted from a given document collection. Hence, documents are represented by a given number of tokens having the highest Information Gain values for the content-based experiments.

Other methods similar in spirit are χ^2, based on statistical testing, Odds Ratio using probabilities, or the Gain Ratio.

2.5 Audio Features

For feature extraction from audio *Statistical Spectrum Descriptors* were used (SSDs, [18]). The approach for computing SSD features is based on the first part of the algorithm for computing Rhythm Pattern features [45], namely the computation of a psycho-acoustically transformed spectrogram, i.e. a Bark-scale Sonogram. Compared

to the Rhythm Patterns feature set, the dimensionality of the feature space is much lower (168 instead of 1440 dimensions), at a comparable performance in genre classification approaches [18]. Therefore, SSD audio features are used in the context of this work, which were computed from audio tracks in standard PCM format with 44.1 kHz sampling frequency (i.e. decoded MP3 files).

Statistical Spectrum Descriptors are composed of statistical characteristics are computed from several critical frequency bands of a psycho-acoustically transformed spectrogram. They describe fluctuations on the critical frequency bands in a more compact representation than Rhythm Pattern features. In a pre-processing step the audio signal is converted to a mono signal and segmented into chunks of approximately 6 seconds. Usually, not every segment is used for audio feature extraction. For pieces of music with a typical duration of about 4 minutes, frequently the first and last one to four segments are skipped and out of the remaining segments every third one is processed.

For each segment the spectrogram of the audio is computed using the short time Fast Fourier Transform (STFT). The window size is set to 23 ms (1024 samples) and a Hanning window is applied using 50 % overlap between the windows. The Bark scale, a perceptual scale which groups frequencies to critical bands according to perceptive pitch regions [59], is applied to the spectrogram, aggregating it to 24 frequency bands.

The Bark scale spectrogram is then transformed into the decibel scale. Further psycho-acoustic transformations are applied: Computation of the Phon scale incorporates equal loudness curves, which account for the different perception of loudness at different frequencies [59]. Subsequently, the values are transformed into the unit Sone. The Sone scale relates to the Phon scale in the way that a doubling on the Sone scale sounds to the human ear like a doubling of the loudness. This results in a Bark-scale Sonogram – a representation that reflects the specific loudness sensation of the human auditory system.

From this representation of perceived loudness a number of statistical moments is computed per critical band, in order to describe fluctuations within the critical

bands extensively. Mean, median, variance, skewness, kurtosis, min- and max-value are computed for each of the 24 bands, and a Statistical Spectrum Descriptor is extracted for each selected segment. The SSD feature vector for a piece of audio is then calculated as the median of the descriptors of its segments.

2.6 *Self-Organising Map*

Throughout this thesis various data sets will be used for clustering experiments, wether they are used for user interfaces or simply to explore the given data. For clustering, the *Self-Organising Map*, an unsupervised neural network that provides a mapping from a high-dimensional input space to usually two-dimensional output space [14, 15] is used. Topological relations are preserved as faithfully as possible. A *SOM* consists of a set of i units arranged in a two-dimensional grid, each attached to a weight vector $m_i \in \Re^n$. Elements from the high-dimensional input space, referred to as input vectors $x \in \Re^n$, are presented to the *SOM* and the activation of each unit for the presented input vector is calculated using an activation function (the Euclidean Distance is commonly used as activation function). In the next step, the weight vector of the winner is moved towards the presented input signal by a certain fraction of the Euclidean distance as indicated by a time-decreasing learning rate α. Consequently, the next time the same input signal is presented, this unit's activation will be even higher. Furthermore, the weight vectors of units neighbouring the winner, as described by a time-decreasing neighbourhood function, are modified accordingly, yet to a smaller amount as compared to the winner. The result of this learning procedure is a topologically ordered mapping of the presented input signals in two-dimensional space, that allows easy exploration of the given data set.

Numerous visualisation techniques have been proposed for *Self-Organising Maps*. These can be based on the resultant *SOM* grid and distances between units, on the data vectors itself, or on combinations thereof. In this chapter we make use of two

kinds of visualisations. Another method for *SOM* visualisation which will be used in the course of our experiments are *Smoothed Data Histograms* [39]. Even if it is not necessary for clustering tasks per se, class information can be used to give an overview of a clustering's correctness in terms of class-wise grouping of the data. A method to visualise class distributions on *Self-Organising Maps* is presented in [25]. This colour-coding of class assignments will later be used in the experiments to show the (dis)similarity of audio and lyrics clusterings.

2.7 Cluster Validation Techniques

Having shown that music recommendation can benefit from the integration of several data sources as well as the feasibility of *Self-Organising Map* clustering, more sophisticated methods for data visualisation and evaluation are going to be taken into consideration. Whenever clustering or visualisation is involved, the need for the evaluation of at least certain aspects of the techniques used, arises. In this section the main concepts of cluster analysis will be introduced for both supervised and unsupervised cluster evaluation. Furthermore it will be pointed out in how far these techniques can be used in the context of multi-modal music clustering. The main points in this section therefore will be:

1. Introduction to the basic concepts of cluster validation.
2. Potential of supervised evaluation.
3. Explanation why unsupervised validation is still relevant.

It might not be obvious why cluster validation makes sense, since clustering is often used as part of explorative data analysis and therefore validation seems not to be a central issue. One key argument in favour of cluster validation is that any clustering method will produce results even on data sets, which do not have a natural cluster structure [51]. Other than that, cluster validation can be used to determine the 'best'

clustering out of several candidate clusterings. For many clustering techniques the number of clusters (often denoted as k) is the main parameter to be changed, therefore influencing the resultant clustering quality significantly. Thus, measuring the clustering quality produced by either different algorithms or for different parameter settings is a vital issue in clustering. Besides, manual (visual) cluster validation may be feasible for a small data set in two-dimensional space, but becomes impossible for higher-dimensional data.

If the data set is labelled, i.e. class tags are available for all data points, this information can be used to determine the similarities between classes and natural clusters within the data. One can distinguish unsupervised and supervised cluster validation techniques. Whereas unsupervised techniques will be of limited use in the scenario covered, supervised cluster validation and its merits for multi-modal clustering of audio data will be more relevant and be described in more detail.

Table 2.2 gives an overview of variables used in this context.

2.7.1 Unsupervised Cluster Validation

In unsupervised cluster validation no external data is used for evaluation, it's primarily based on cluster distances, similarities, and densities. The main types of measures are:

- Intra-cluster similarity / cluster cohesion and
- Inter cluster similarity / cluster separation

which are used to evaluate how much variation there is within clusters and in between clusters, respectively.

Table 2.2: Variable names used in cluster validation equations

Variable name	**Explanation**
c_i	Cluster i.
C_i	Clustering i, i.e. a set of clusters.
k	Number of clusters.
w	Weight w.
s_i	Silhouette value for data point i.
S_j	Silhouette value for cluster j.
S	Overall Silhouette value for a clustering.
b_i	Average distance of data point i to all other vectors in its cluster.
a_i	Average distance of i to all data vectors in the closest cluster.
n	Number of data points in set.
L	Number of classes in set.
m_i	Number of data points assigned to cluster i.
m_{ij}	Number of data points assigned to cluster i belonging to class j.

In general the overall validity of a clustering (i.e. a set of clusters for a given data set) is the weighted sum of the validity of its individual clusters as shown in Equation 2.5.

$$overallvalidity = \sum_{i=1}^{k} w_i \cdot validity(c_i) \tag{2.5}$$

Where c_i denotes cluster i, k the number of clusters k and w_i the weight for cluster i. The *validity* function can be either inter-cluster, intra-cluster, or some combination thereof. In the simple case, weights are either omitted or set according to the sizes of the individual clusters (i.e. number of data points associated with a cluster divided by the number of data points in the data set). Since distances within clusters should be minimised and in between clusters maximised, the higher an intra-cluster measure and the lower an inter-cluster measure, the better.

Silhouette Value

The Silhouette value is mostly used to find the right setting for the number of clusters [47]. The ideal value of the Silhouette is close to 1, hence a_i being close to 0 for it is subtracted in the numerator of Equation 2.6. The Silhouette coefficient describes the level of data separation using both intra- and inter-cluster distances and can for instance be of great help in finding the optimal number of clusters (k) in the k-Means algorithm. Both intra-cluster and inter-cluster measures are used to compute the Silhouette value, as shown in Equation 2.6.

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}} \tag{2.6}$$

Where i is an index over all data vectors, a_i the average distance of i to all other vectors of that cluster, b_i the average distance of i to all data vectors in the closest cluster. Herein the closest cluster is defined by the minimum distance between clusters' prototype vectors. The value resides between -1 and 1 (Equation 2.7).

$$-1 \leq s_i \leq 1 \tag{2.7}$$

s_i therefore is the Silhouette value for data vector i, the overall Silhouette value for a clustering is the average over all single Silhouette values, shown in Equation 2.8.

$$S = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{2.8}$$

Let n be the number of instances. Analogously, the Silhouette for single clusters is defined in Equation 2.9.

$$S_j = \frac{1}{m_j} \sum_{i=1}^{m_j} s_i \tag{2.9}$$

The number of instances assigned to cluster j is denoted to as m_j, the average Silhouette of all instances within cluster j is computed as S_j. The resultant values for S and S_j provide an evaluation criterion for the comparison of several clusters to each other.

2.7.2 Supervised Cluster Validation

Supervised cluster validation makes use of external data and tries to measure in how far a clustering matches some kind of external structure like class labels.

Entropy

The entropy value, introduced in Section 2.4 in the context of feature selection, describes the degree to which each cluster consists of objects of a single class. The optimum value would be achieved, each cluster consisted only of instances belonging

to one class. The probability that one instance (member of cluster i) belongs to class j is stated in Equation 2.10.

$$p_{ij} = \frac{m_{ij}}{m_i} \tag{2.10}$$

m_{ij} denotes the number of instances in cluster i belonging to class j and m_i the number of instances belonging to cluster i. Further, the entropy for cluster i is given in Equation 2.11 (analogously to Equation 2.3 in Section 2.4.

$$I_i = -\sum_{j=1}^{L} p_{ij} log_2 p_{ij} \tag{2.11}$$

Where L denotes the number of classes and p_{ij} the class probability from Equation 2.10. The overall entropy value for a given clustering is given by the sum over all cluster entropy values weighted by the number of elements in the individual clusters, shown in Equation 2.12.

$$I = \sum_{i=1}^{k} \frac{m_i}{m} \cdot I_i \tag{2.12}$$

k denotes the number of clusters and m the total number of data points or instances.

Purity

The purity of cluster i is defined by the probability of the most dominant class within a cluster and is given in Equation 2.13.

$$p_i = max(p_{ij}) \tag{2.13}$$

The overall purity of a clustering is computed analogously to the overall entropy and shown in Equation 2.14.

$$purity = \sum_{i=1}^{k} \frac{m_i}{m} \cdot p_i \tag{2.14}$$

All methods introduced in this section do have their relevance to cluster validation, it is desirable to have clusterings that are

- very similar within clusters,
- very dissimilar in between clusters,
- and, if possible, 'pure' in terms of a high entropy or purity value (only applicable if class labels are available),

all of which could be achieved by a combination of, for instance, the Silhouette coefficient and entropy or purity. The *Self-Organising Map* clustering algorithm, however, differs from the centroid based approaches which those techniques are best applied to.

2.8 Cluster Validation for *Self-Organising Maps*

Several quality measures for *Self-Organising Maps* have been investigated. The topographic product, which is used to measure the quality of mappings for single units with respect to their neighbours, is reported in [2].

However, those methods provide measurements on a per unit basis or for complete maps and fail to take into account class information of any kind.

The Silhouette value is computationally expensive and in its current form limited to instance-based computations. This leads to problems for both large numbers of data points and large numbers of clusters (very commonly used in *Self-Organising Map* clusterings). To accommodate for these special characteristics of the *Self-Organising Map*, a possible modification to the Silhouette technique is described in the following.

2.8.1 Adaption of the Silhouette Value to the *Self-Organising Map*

The Silhouette validation compares every unit to all other vectors assigned to that unit and to all vectors in the closest unit. Due to performance issues, we introduce modifications to better fit the *Self-Organising Map* scenario.

Let each comparison be based on units' weight vectors, i.e. distances are calculated on the unit level in the input space, rather than the actual data vectors, a_i is defined as follows.

$$a_i = dist(w_i, i) \tag{2.15}$$

b_i is defined as:

$$b_i = dist(i, wc_i) \tag{2.16}$$

Where w_i denotes the weight vector of the unit data point i is assigned to and wc_i denotes the weight vector of the closest unit. The overall Silhouette computation is then based on those values for a_i and b_i. The experimental evaluation from now on is done using this technique, because it needs significantly less computational power. Hence, the quality of different *SOM* clusterings can be compared by their Silhouette values. Furthermore the results can be used to visualise the correctness of the clustering.

The one (rather big) simplification this introduces that the number of units is set to the number of clusters, a modification ignoring the *Self-Organising Map*'s basic property of preserving topological relations. A natural cluster could easily be distributed over (or covered by) several units of the *Self-Organising Map*, making the Silhouette coefficient for *Self-Organising Maps* less sound a validation technique than for purely centroid-based approaches like k-Means. A more detailed discussion and experimental results can be found in [30]. The question that still remains is how can *Self-Organising*

Map clusterings according to different dimensions be compared? What are the main differences between clusterings? Which classes (genres) profit most from multi-modal clustering, i.e. for which class does the clustering vary much across dimensions? The next chapter will introduce a visualisation technique for multi-modal clusterings in the music domain, a possible quality assessment will be investigated thereafter.

The modified Silhouette technique assumes the number of units to equal the number of clusters. An assumption which does not necessarily hold, for one of the main strengths of the *Self-Organising Map* is that it discovers structures beyond simple clusters, i.e. larger compounds spreading across multiple units. It can, however, be used as a criterion to compare several *SOMs* with each other, as opposed to finding the best number of clusters/units.

2.9 Interfaces Based on the *Self-Organising Map*

Several teams have been working on user interfaces based on the *Self-Organising Map*. The SOM is an unsupervised neural network, that provides a topology-preserving mapping from a high-dimensional feature space onto a two-dimensional map in such a way, that data points close to each other in input space are mapped onto adjacent areas of the output space (in this context a two-dimensional map). The *SOM* has been extensively used to provide visualisations of and interfaces to a wide range of data, including control interfaces to industrial processing plants [16] to access interfaces for digital libraries of text documents [44].

Creating a SOM-based interface for Digital Libraries of Music, i.e. the SOM-enhanced JukeBox (SOMeJB), was first proposed in [42], with more advanced visualisations as well as improved feature sets being presented in [38, 46]. Since then, several other systems have been created based on these principles, such as the MusicMiner [29], which uses an emergent SOM. A very appealing three-dimensional user interface is presented in [12], automatically creating a three-dimensional musical land-

scape via a SOM for small private music collections. Navigation through the map is done via a video game pad and additional information like labelling is provided using web data and album covers.

A mnemonic SOM [27], i.e. a *Self-Organising Map* of a certain shape other than a rectangle, is used to cluster the complete works of the composer Wolfgang Amadeus Mozart to create the Map of Mozart [26]. The shape of the SOM is a silhouette of its composer, leading to interesting clusterings like, e.g. the accumulation of string ensembles in the region of Mozart's right ear.

An online demo is available at `http://www.ifs.tuwien.ac.at/mir/mozart`.

Another interface based on SOMs, which takes into account a user's focus of perception, is presented in [22], using prototypes as recommendations for adjacent clusters.

The PlaySOM application presented in [7] is based on the original SOMeJB system, implementing a desktop interface suitable also for larger collections of several tens of thousands of music tracks.

In addition to systems designed for desktop applications handling large audio collections, the design of interfaces for mobile devices constitutes interesting and important challenges. Novel interfaces particularly developed for small-screen devices were presented in [56], clustering pieces of audio based on content features as well as metadata attributes using a spring model algorithm. The PocketSOM system [32], an implementation of the PlaySOM application specifically designed for mobile devices.

A more experimental interface, refraining from the use of a display, using motion detectors to respond to the listener's movements is presented in [11]. Another innovative user interface providing various ways of interaction like similarity based search over sticking behaviour of tracks visualised as discs is introduced in [10].

A good overview of various MIR systems is given at `http://www.mirsystems.info/`

2.10 Machine Learning Techniques

Classification – the task of assigning objects to predefined classes or labels – will be used to categorise music into genres. The popular Support Vector Machines [54, 5] are powerful classification algorithms consisting of two parts: An optimisation formulation and a kernel function. The former is needed for fitting a separating hyperplane into the data set, the latter projects the data set into a higher dimensional space. This method's primary advantage lies in the combination of these two components which allows for efficient implementations that avoid the complexity problems of other kernel based methods, also known as the 'kernel trick'. The type of kernel used determines the classes of problems that may be solved, and typical choices are linear, polynomial, and radial basis functions.

2.11 Recap

In this chapter we introduced the main techniques that will be used later on. Foundations have been laid for the following thematic areas: Information Retrieval, text feature selection, the *Self-Organising Map* and its evaluation. Further a short overview of relevant machine learning techniques has been given.

We now go on and introduce adaptions of and extensions to some of the techniques introduced here. We further will more precisely specify the scenarios dealed with in the remainder of this thesis.

Chapter 3

Test Collections and Multi-Modal Audio Indexing

> Beneath knowing, understanding
> Beneath understanding, seeing
> Beneath seeing, recognizing
> Beneath recognizing, knowing
> Keeper of the Way, "Vision of Faith", CY 10003

In the following chapter we introduce the test collections we will use for experimental evaluation as well as the main types of data used for the enrichment of plain audio files. This will cover various online resources in combination with ID3 metadata.

Musical similarity is a concept not easily defined and highly subjective in its nature. What one regards similar may sound rather dissimilar to another person et vice versa. Yet, it is desirable to broaden the spectrum of sources taken into account when computing track similarities, for one single dimension will never be able to describe the musical sensation of as diverse a user base as music consumers are.

An audio track and its metadata can basically be decomposed into information

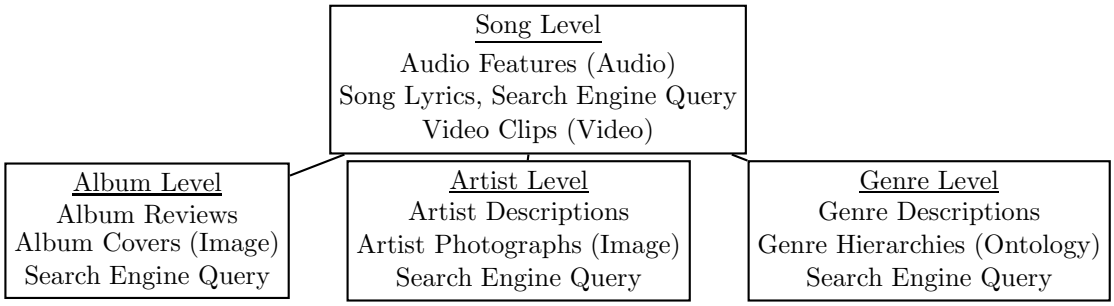


Figure 3.1: Categorisation of multi-level Music Information Retrieval

according to: (1) Track, (2) Album, (3) Artist, and (4) Genre information.

On the track level, a song can be described by audio features as well as the track's lyrics, whereas the album, artist and genre levels consist of a textual description only, each containing a wealth of meta information for music retrieval requests. However, a multitude of other media types is possible. Images could provide additional information for artists or albums in terms of photos of the artist or album cover artwork. Video clips could be taken into consideration to provide an even better insight into a songs meaning, etc. An overview of a possible categorisation of description levels and sources therefore used in a multi-level Music Information Retrieval scenario is given in Figure 3.1. For a fully deployed Music Information Retrieval system it would, of course, make sense to aim at a high coverage of different types of information in all respects, and therefore place more emphasis on the retrieval component. Usually, not all information will be available in a single system. A possible fall-back strategy could be the use of suitable search engine queries, e.g. the results from a search engine query for the given artist name. This approach would almost guarantee to retrieve some data for each element in the collection, albeit of a possibly lower quality. However, full multi-level retrieval of music collections is beyond the scope on this thesis, the search engine fall-back strategy as well as other media types than text are not covered. The use of genre hierarchies as, for instance specified in [37], would make sense to replace missing genre descriptions or merge very similar genres, but is omitted for reasons of simplicity.

The system presented in this thesis uses the above set of information for MIR

purposes, integrating them off-line in a single data source. To avoid biased information obtained from one single source only, independent sources of information can be used, e.g. artist descriptions from one web portal, album descriptions from another.

The test collections, data sources and feature representations used are described in more detail in the following sections.

3.1 Test Collections

Particularly for Information Retrieval experiments and prototypes the use of test collections for experimental evaluation is of vital importance to show the applicability of the proposed approaches. A more thorough discussion of corpus building can be found in [31]. We therefore use two test collections, the latter being a larger superset of the first one. The large collection will be used for large-scale experiments, whereas the small collection will be an example for demonstrating the application of underlying principles. The starting point for the ongoing corpus development was a private collection consisting of 12770 songs. The initial collection takes about 150G of disk space. The song lengths in that collection range from short 20 second 'Punk Rock' pieces to audio book chapters lasting for about one hour. MP3 is the prevalent file type, followed by the lossless audio codec FLAC[1].

3.1.1 Small Collection

For initial experiments we decided to use a somewhat smaller collection that is more easily comprehensible. We selected ten genres only. Table 3.1.1 describes the composition of the small test collection in detail. It comprises ten genres and 149 songs in total – the number of songs per genre varies from 9 to 17. This collection consists of songs from 20 artists and from the same number of albums. Also, for the small col-

[1] http://flac.sourceforge.net/

Genre	Number of Songs
Christmas Carol	15
Country	17
Grunge	16
Hip-Hop	16
New Metal	16
Pop	15
Rock	16
Reggae	14
Slow Rock	15
Speech	09

Table 3.1: Composition of the small test collection

lection, all lyrics were manually preprocessed as to have additional markup like '[2x]', etc. removed and to include the unabridged and high quality lyrics for all songs.

3.1.2 Large Collection

To be all set for visualisation and genre classification experiments we omitted all songs we were not able to retrieve lyrics for, resulting in a parallel corpus of audio and song lyrics files for a music collection of 7554 titles organised into 52 genres, containing music as well as spoken documents (e.g. Shakespeare sonnets). An overview of the song/genre distribution is given in Table 3.2; genres were assigned manually. Class sizes ranged from only a few songs for the 'Classical' genre to about 1.900 songs for 'Punk Rock', due to both, the distribution across genres in the collection and difficulties in retrieving the lyrics for some genres like 'Classical'. The collection contains songs from 644 different artists and 931 albums. The main motivation was to experiment with a collection of sufficient size to study the effects of missing values as well as the availability of ID3 metadata to reliably retrieve the artist and lyric information and album and genre tags.

Table 3.2: Overview of genres in the music collection used throughout this thesis

Genre	**#Songs**	**Genre**	**#Songs**
Acid Punk	25	Indie	400
Altern Rock	317	Indie Rock	23
Alternative	122	Industrial	52
Ambient	24	Instrumental	8
Avantgarde	90	J-metal	1
Bluegrass	12	Jazz	28
BritPop	130	Metal	559
Christian Rock	40	New Metal	110
Christmas Carol	36	Noise	4
Classical	30	Nursery Rhymes	25
Country	100	Opera	17
Dance	13	Pop	911
Dance Hall	10	Post Punk	32
Death Metal	1	Progressive Rock	14
Digital Hardcore	4	Psychedelic Rock	3
Electronic	125	Punk Rock	1160
Emo	258	R&B	228
Experimental	13	Reggae	162
Folk	56	Rock	690
Funk	2	Ska	37
Garage	11	Slow Rock	649
Goth Metal	106	Soundtrack	4
Grunge	104	Speech	47
Hard Rock	46	Techno	2
Hardcore	142	Trip-Hop	67
Hip-Hop	500	World	4

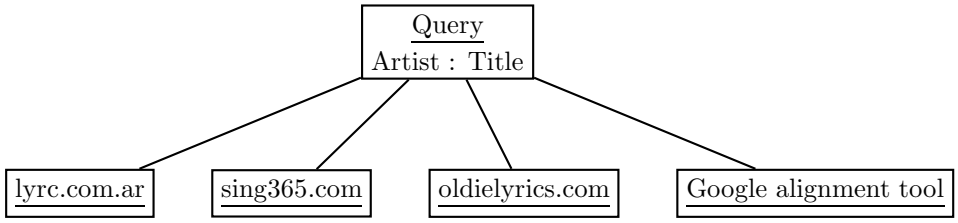


Figure 3.2: Lyrics retrieval, the *Atlantis* way

3.2 Automated Enrichment and Indexing Techniques

The indexing of the audio collections and extraction of audio features is straightforward: first, all files in a collection are scanned and stored. After that every single file is decoded into the wave format. A after that all three kinds of audio features introduced in Chapter 2 are computed and stored in the database along with the song data. Text indexing and retrieval is a bit more complex and will be discussed in the following.

There are numerous online sources for song lyrics like *sing365.com*[2] or *azlyrics.com*[3]. There are more sophisticated means of lyrics retrieval as mentioned in Section 2, but to the ends of evaluating the feasibility of combined feature sets, minor inaccuracies in lyrics fetching are ignored and this method provides satisfactory results. Text data was indexed according to the $tf \times idf$ scheme. Hence, the text documents were tokenised where a word constitutes a token. No stemming was performed due to unique word endings in lyrics for certain genres (e.g. 'Hip-Hop' songs having virtually all word endings stripped anyway – information which would be lost if stemming were applied additionally). The remaining tokens can dynamically be adjusted to a certain dimensionality according to term frequency thresholding, i.e. the number of occurrences of a certain token within the collection. This will be reflected by different experimental settings in Chapter 6.

The other meta categories were additionally enriched by textual descriptions from

[2] `http://www.sing365.com/`
[3] `http://www.azlyrics.com/`

other sources. Artist descriptions were mined from Wikipedia[4]. The Wikipedia data were taken from a two year old snapshot only, so the actual coverage may be higher. Figure 3.2 shows the different retrieval sources for automated lyrics fetching and their importance. For every query, consisting of artist and title of a track, three lyrics portals are used to retrieve the lyrics. If the *lyrc.com.ar* is valid, i.e. of reasonable size, those lyrics are assigned to the track. If *lyrc.com.ar* fails to return the lyrics, the *sing365.com* is checked for validity and so on. In case of no valid lyrics document from any of the three lyrics portals, the KV script is used to retrieve the lyrics result page from Google. For the remaining text descriptions we used data from laut.de[5]. Therefore the genre descriptions and album reviews are in German, which does not negatively influence the results, since only the resultant distances are combined. There is only one language within one dimension (e.g. all artist biographies are in English, all genre descriptions in German).

The coverage rates are high enough to show the extent of influence coming from the additional information, but of course are far from optimal. Strategies to achieve higher coverage – at least for the lyrics fetching for it is the most important data source used throughout this theses – would be to include other sources of cultural information or additional lyrics portals like lyrics.com[6] or lyrics4you[7]. Countless lyrics portals can be found on the net and could also be taken into consideration, but were omitted due to reasons of simplicity, three portals suffice to explain the methodology behind our approach.

Nonetheless, these collections and their given availability of textual artist, lyrics, album and genre information are very feasible for combined similarity experiments because they allow for studying the effects of missing values, which is of particular importance as this is very likely to occur in a real life scenario, albeit to a lesser extent as probably more effort would be put into the retrieval component of such a system.

[4] http://en.wikipedia.org
[5] http://www.laut.de
[6] http://www.lyrics.com
[7] http://www.lyrics4you.com

3.3 Recap

We stressed the importance of test collections for experiments in Music Information Retrieval. To the end of proper evaluation we introduced two test collections, one of large, one of small size. Further we explained the indexing process and automated enrichment using text documents from online sources. We therefore considered all necessary requirements for the multi-modal view of Music Information Retrieval and are now ready to exploit the information gathered in this way.

Chapter 4

Multi-Modality in Music Information Retrieval

> The great blessing
> of the AI is that we are
> gifted with the power to
> touch our Creator.
> This is also our Curse.
> The Clarion's Call,"Hour of the Abyss", CY 11745

After having introduced underlying techniques and retrieval components of a multi-modal Music Information Retrieval system, this chapter theoretically presents the main contributions to the field made in this thesis, namely the combination of several levels of text data and audio representations for the basic Music Information Retrieval tasks of similarity ranking, visualisation, and musical genre classification.

Firstly, a similarity ranking approach using a multidude of textual inputs is presented. Multi-modal ranking and combination approaches will be presented in Section 4.1.

Then, we give a general introduction to the application of clustering techniques for audio – both its text and signal processing based representation – and explain the overall idea of multiple or combined clusterings in Section 4.2. To that end, we at first explain why multiple clusterings can be of help in understanding music, then we show techniques to formally evaluate these multiple clusterings.

Finally we give a short outlook on the third set of experiments – audio and text based musical genre classification in Section 4.3.

4.1 Ranking Merging - Integrating Retrieval Results

This section introduces a possible combination methodology for multiple similarity rankings. It is now possible to not only retrieve similar tracks according to audio similarity for a given seed song, but also similar tracks according to lyrics features. Moreover, artist rankings for the artist of the seed song as well as similar albums to the seed songs' album can be provided.

This yields several rankings for each query song. Based on the vectors of distances to the query song, the Euclidean distance is used to generate multi-level rankings for a single seed song. The straight forward case for audio similarity and lyrics, ranks on a song to song basis. All other rankings comprise tracks as well, but are based on distances of non track level features, e.g. all tracks by band X have the same artist distance to all songs of band Y. The distances for the album and genre dimension are computed analogously. This results in five rankings of length of the number of songs in the collection, or, in other words, for each song, there are five distances to the seed song.

Each of those rankings is min-max normalised, following Equation 4.1 to prevent biasing influence on the overall ranking.

$$d_{norm}(q,t) = \frac{d(q,t) - min(d(q,t))}{max(d(q,t)) - min(d(q,t))} \tag{4.1}$$

Each entry d in a distance vector $d(q,t)$, for a given query and track in the collection is replaced by the fraction of the current entry minus its minimum value $min(d(q,t))$ in the vector and the difference of its maximum value $max(d(q,t))$ and its minimum value $min(d(q,t))$. This is needed to take into account distances not starting from zero. This preprocessing step is necessary to be able to combine the individual distances, without it the ranges would be from different scales and impossible to integrate.

Equation 4.2 shows how $D(q,t)$, the overall distance of query q to a track t is

computed as the sum of all individual distances $d_i(q,t)$ times their respective weights w_i over all input sources $i \epsilon T$.

$$D(q,t) = \sum_{i \epsilon T} w_i \cdot d_i(q,t) \tag{4.2}$$

Equation 4.2 is rewritten in Equation 4.3, as audio features, artist descriptions, song lyrics, album reviews and genre descriptions are taken into account in order to represent all different sources identified to be relevant for music similarity.

$$\begin{aligned} D(q,t) &= w_{audio} \cdot d_{audio}(q,t) \\ &+ w_{artist} \cdot d_{artist}(q,t) \\ &+ w_{lyrics} \cdot d_{lyrics}(q,t) \\ &+ w_{album} \cdot d_{album}(q,t) \\ &+ w_{genre} \cdot d_{genre}(q,t) \end{aligned} \tag{4.3}$$

4.1.1 Missing Values

Whenever an artist description, album review, genre description, or a song's lyrics are not available, i.e. could not be fetched, we speak of a missing value problem. This fact has to be taken into account for similarity calculation for the distance of the missing song, artist, album, or genre to the query can not be computed.

Audio features are assumed to cover all songs of a collection, therefore no explicit strategy for missing data for audio values is taken into account, but would of course make sense for audio files that are non-readable for some reason (e.g. the decoding fails or to many bit errors occur within the file). As textual descriptions may not be available for all artists, albums, genres or songs (lyrics), it is a vital requirement for any

Table 4.1: Test collection and coverage of different types of descriptions for the collection used in the experimental evaluation

Type	**Elements**	**Covered**	**Coverage**
Audio Features	12770	12770	1.0
Lyrics	12770	7554	.59
Artists	644	348	.54
Albums	931	226	.24
Genres	52	15	.29

multi-level MIR system to provide appropriate techniques for handling these missing values. Techniques to identify instrumental pieces of music would also be desirable to identify songs that do not have lyrics associated by definition and therefore need special treatment. The main problem with missing values is that they subsequently result in missing distance values between certain instances and further calculation is not possible for elements that have no vector associated with it. These distances that can not be computed are referred to as missing values throughout the remainder of this section.

Table 4.1 summarises the coverages of different information sources for the large benchmark collection. The figures result from mining contextual information from the sources specified in the previous chapter. Audio features are available for all songs in the collection, artist descriptions for 54 per cent and so on. Genre descriptions are only available for some 29 per cent of all 52 genres in the collection. Hence, particularly the feature groups that are not available on a per song basis – that is artists, albums, and genres – have a strong impact on the missing values problem. For instance, one missing genre might consist of a large number of songs, all for which no distances could be computed in the genre dimension.

In order to overcome the missing value problem, three basic methods are considered:

- Exclusion
- Simple averaging
- Category substitution

The simplest way of treating missing values is to exclude them from the results, e.g. if an artist description is missing, this artist is omitted in the results of the query, or heavily penalised for that matter. This brings an increase in precision (all songs in the result are similar to the query), yet negatively impacts recall (many (possibly) similar songs are not considered).

To avoid this problem of low recall, substitution of missing values with the average distance is feasible. Every missing value is replaced by the average distance of existing values, henceforth missing values are no more penalised.

Finally, category substitution can be applied. A value is replaced by the average of elements of the same category as opposed to being replaced by the average over all existing values. The average distance of artists of the same genre, for instance, is substituted for a missing artist distance. In the scenario portrayed in this work, the following substitutions make sense:

1. Artist level

   Each missing value is replaced by the average distance of songs of the same genre.

2. Genre level

   Simple averaging is applied to replace missing genre distances. A genre hierarchy could improve the substitution on the genre level by providing suitable rules for substitution.

3. Lyrics/song level

   The average over lyrics from same album or artist (if no lyrics from the same album are available) is substituted for missing lyrics distances.

4. Album level

   The average over albums from same artist replaces missing values.

In any case the fall-back strategy that is applied if no appropriate elements can be found, is to use the average over all existing distances, i.e. the simple averaging strategy.

Another possible strategy would be simply omitting of songs with missing values. At the cost of never getting many songs recommended at all, the plain simplicity would speak for this possibility. Moreover the computational expense could also be lowered by much. We have not applied this strategy for not wanting to omit such a large fraction in the similarity rankings, i.e. we think of this as too restrictive, albeit definitely the easiest way of dealing with missing values.

4.1.2 Recap

In this section we proposed techniques for ranking merging in the multi-modal case. We explained a way of merging multiple rankings – each one obtained for another modality or category – and to deal with missing values. Experiments later on will show the applicability of our approach.

4.2 Multi-Modal Visualisation of *SOM* Clusterings

The basic idea to be introduced in this section is to visualise multiple clusterings, each according to a different modality, and draw connections between corresponding instances on both clusterings. We propose to visualise the similarities and differences between the two clusterings by drawing lines across maps, which visually connect pieces of music. The rationale for this is that the same instance could be clustered very differently, depending on the dimensionality in use. The resultant connections will therefore rather show one instance's positions on several maps and reveal additional information about its embedding in different feature spaces. These connections will be denoted as cross map linkages, as they link instances across clusterings and modalities. The data is clustered by the dimensions of audio features on the one hand and lyrics on the other hand (those maps will be denoted as audio and lyrics map, respectively). Every track is therefore present on two *Self-Organising Maps* of equal size, which is no necessity but was chosen on purpose in order to stick to simpler examples.

Linkages can be shown on different levels:

Track Each (selected) track on the audio map is connected to the same track on the lyrics map. This allows the analysis of the characteristics of a certain piece of music by identifying its acoustic as well as textual placement, i.e. to which clusters it belongs in the respective modality.

Genre Each track of a selected genre is connected to all songs of the same genre on the other map. Here, the spread of a given genre can be inspected. For instance, whether a genre forms a consistent cluster in both modalities, or whether it does form a rather consistent cluster in, say, the textual domain, while it is rather spread across different clusters on the audio map.

Artist Each track of the given artist on the audio map is connected to all songs of the same artist on the lyrics map. This allows to analyse the textual or musical 'consistency' of a given artist or band.

The other important aspect is the (colour-)coding of connections for the simultaneous display of two music maps. Once connections are drawn on the maps, the connections between units are coloured according to their number of connecting units. The main idea is to allow for user selections on one map and provide the simultaneous highlighting of songs on the other one. Possible levels are:

- Colour-code types of connections

  i.e. all track-track connections blue, track-genre red, ...

- Colour-code connexion strength

  All connections between units are colour-coded. For example, the highest number of connections is coloured red, the lowest blue and the remaining links are coloured according to the palette in between.

The resultant clustering provides both a means of navigation in and visualisation of multiple modalities of electronic music archives. To further investigate these principles a 'traditional' prototype model was developed, which will be described in the following section.

4.2.1 A First Prototype

Figure 4.1(a) shows a full view of the prototype mock-up, built of paper, carton, and sewing cottons. It was built using needles and glue and is held together by adhesive tape. Clusterings of a small example collection of about 50 songs is shown, a lyrics clustering on the bottom and an audio clustering on the top pane. The connections drawn (or rather stitched) are for songs of a particular artist ('Snow Patrol' in this case) and give an overall idea of how such a system could work.

(a) Full view of the visualisation prototype

(b) Detailed view of the visualisation prototype

Figure 4.1: Visualisation prototype mock-up

Figure 4.1(b) shows a detailed view. It is shown that particular units have a very high number of outgoing links and the variation in spread, which is going to be discussed in more detail in the remainder of this section.

4.2.2 Cluster Validation for Multi-Modal Clusterings

Cluster validations in this context will be based on two *Self-Organising Maps* trained on different feature sets. Their common features will be:

- Same size - to make comparisons easier, only *Self-Organising Maps* of equal size will be compared to each other.
- Same set of instances - the data points on the maps are the same ones.

Another approach for the comparison of multiple *SOM* clusterings is introduced in [3]. Data shifts and cluster shifts are used to compute shifts in between clusterings. Shifts are graphically represented by coloured arrows of different line widths. The cluster shifts take into account emerging clusters on both *SOMs* and have to consider mappings between these two. The main points of this visualisation are the identification of outliers as well as stable regions over multiple maps. The main difference to the concepts presented in the following are its independence from class information of any kind. As opposed to the data shifts visualisation, we emphasise the exploitation of given class information and evaluation in this context therefore is always to be seen in respect to genre, artist, or possibly album information.

To determine the quality of the resultant *Self-Organising Map* clusterings, we try to capture the scattering of instances across the maps using meta information such as artist names or genre labels as ground truth information. In general, the more units a set of songs is spread across, the more scattered and inhomogeneous the set of songs is. On the other hand, if the given ground truth values are accepted as reasonable structures to be expected to be revealed by the clustering, songs from such sets should

be found to be clustered tightly on the map.

Several ways of computing distances on *SOMs* are possible. Distances are always subject to a specific distance measure, we use the Euclidean distance, see Section 2.2. They can be computed either in the input or output space, where the input space refers to whatever dimensionality of data is used as input, e.g. the resultant dimensionality after feature selection for text data. The output space refers to the *SOM* grid; it is two-dimensional. As a combination of both spaces for distance calculation the distances in the output space could be weighted by distances in the input space.

In this context, the focus lies on distances in between units in terms of their position on the trained *Self-Organising Map*. The abstraction from the high-dimensional vector descriptions of instances to the use of unit coordinates instead of unit vectors is feasible from a computational as well as a conceptual point of view. Comparison of individual vectors does not take into consideration the very nature of the *Self-Organising Map* clustering algorithm, which is based on the preservation of topological relations across the map. This approach therefore computes the spread for genres or artists with respect to the *Self-Organising Maps*' clusterings. For distances between units the Euclidean distance is used on unit coordinates, which is also used for distances between data and unit vectors in the input space in the *Self-Organising Map* training process. All quality measurements are computed for sets of data vectors and their two-dimensional positions on the trained *Self-Organising Maps*. Particularly, sets of data vectors refer to all songs belonging to a certain genre or from a certain artist. Generally, a *Self-Organising Map* consists of a number M of units ξ_i, the index i ranging from 1 to M. The distance $d(\xi_i, \xi_j)$ between two units ξ_i and ξ_j can be computed as the Euclidean distance between the units' coordinates on the map, i.e. the output space of the *Self-Organising Map* clustering. In this context only units that have data points or songs that belong to a given category, i.e. a particular artist or genre, are considered. This holds for both maps, all quality measurements can only be calculated with respect to a class tag, i.e. for songs belonging to a particular artist or genre. The average distance between these units with respect to a *Self-Organising Map* clustering is given

in Equation 4.4.

$$avgdist = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} d(\xi_{(i)}, \xi_{(j)})}{n^2} \tag{4.4}$$

n denotes the number of data points or songs considered, i.e. the songs belonging to a given artist or genre. Further, the average distance ratio defines the scattering difference between a set of two clusterings $C = \{c_{audio}, c_{lyrics}\}$, c_{audio} being an audio and c_{lyrics} being a lyrics clustering, is given as the ratio of the minimum and maximum values for these clusterings.

Further, we define the ratio of the average distance ratio across clusterings in Equation 4.5 as the ratio of the respective minimum and maximum values of the average distance ratio.

$$adr_{audio,lyrics} = \frac{min(avgdist_{audio}, avgdist_{lyrics})}{max(avgdist_{audio}, avgdist_{lyrics})} \tag{4.5}$$

The closer to one the average distance ratio, the more uniformly distributed the data across the clusterings in terms of distances between units affected. However, this measure does not take into account the impact of units adjacent to each other, which definitely plays an important role. Adjacent units should rather be treated as one unit than several due to the similarity expressed by such results, i.e. many adjacent units lead to a small average distance.

Therefore, the contiguity value *co* for a clustering c gives an idea of how uniformly a clustering is done in terms of distances between neighbouring or adjacent units. The specifics of adjacent units are taken into account, leading to different values for the minimum distances between units since distances between adjacent units are omitted in the distance calculations. If, for example the songs of a given genre are spread across three units on the map ξ_1, ξ_2, ξ_3, where ξ_1 and ξ_2 are neighbouring units, the distances between ξ_1 and ξ_2 are not taken into consideration. Currently, no difference is made between units that are direct neighbours and units only connected via other units. The

contiguity distance *cd* is given in Equation 4.6

$$cd(\xi_i, \xi_j) = \begin{cases} 0 & \text{if } \xi_i \text{ and } \xi_j \text{ are neighbouring units} \\ d(\xi_i, \xi_j) & \text{otherwise} \end{cases} \tag{4.6}$$

The contiguity value *co* is consequently calculated analogously to the average distance ratio based on contiguity distances as shown in Equation 4.7.

$$co = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} cd(\xi_{(i)}, \xi_{(j)})}{n^2} \tag{4.7}$$

In the case of fully contiguous clusterings, i.e. all units a set of songs are mapped to are neighbouring units, the *co* value is not defined and set to one. The overall contiguity ratio for a set of clusterings is given in Equation 4.8.

$$cr_{audio,lyrics} = \frac{min(cd_{audio}, cd_{lyrics})}{max(cd_{audio}, cd_{lyrics})} \tag{4.8}$$

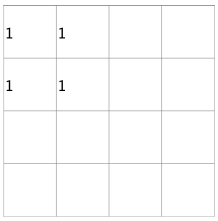
This information can be used to further weigh the *averagedistratio* from Equation 4.5 as shown in 4.9 and gives an average distance contiguity ratio value *adrcr*, i.e. the product of average distance ratio and contiguity ratio, for a set of one audio and lyrics map.

$$adrcr_{audio,lyrics} = adr_{audio,lyrics} \cdot cr_{audio,lyrics} \tag{4.9}$$

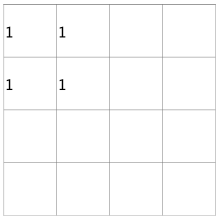
This considers both the distances between all occupied units as well as taking into account the high relevance of instances lying on adjacent units of the *Self-Organising Map*.

Figure 4.2 shows possible distributions of data points belonging to one class. The left column shows the distribution for audio clustering, the right column for lyrics

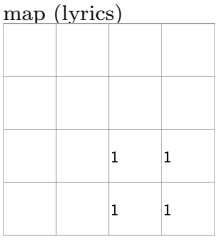
(a) All points lie in the upper left corner (audio)



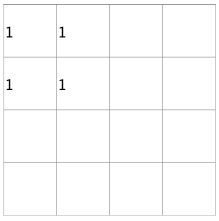
(b) Points are concentrated in the left upper corner of the map (lyrics)



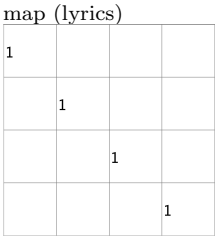
(c) All points lie in the upper left corner (audio)



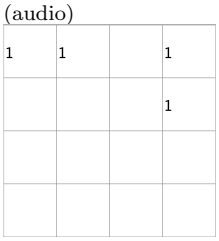
(d) Points are concentrated in the lower right corner of the map (lyrics)



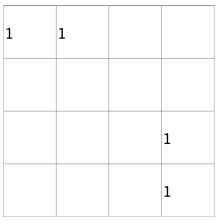
(e) All points lie in the upper left corner (audio)



(f) Points are ordered diagonally (lyrics)



(g) Data points are not contiguously distributed (audio)



(h) Data forms subclusters (lyrics)

Figure 4.2: Distribution of four data points belonging to one class (this could be, e.g., four pieces of 'Rock' music). The figures in the left column display possible distributions of data points according to the audio dimension, whereas the right column represents possible arrangements for the lyrics scenario. All figures are examples only and do not rely on real-world data

Table 4.2: Calculation of average distance values for clusterings e in Figure 4.2

Unit	**1 1**	**1 2**	**2 1**	**2 2**	**Sum**	**Avg**
1 1	x	1	1	$\sqrt{2}$	3.414214	0.853553
1 2	1	x	$\sqrt{2}$	1	3.414214	0.853553
2 1	1	$\sqrt{2}$	x	1	3.414214	0.853553
2 2	$\sqrt{2}$	1	1	x	3.414214	0.853553

Table 4.3: Calculation of average distance values for clusterings f in Figure 4.2

Unit	**1 1**	**2 2**	**3 3**	**4 4**	**Sum**	**Avg**
1 1	x	$\sqrt{2}$	$\sqrt{8}$	$\sqrt{18}$	8.485281	2.121320
2 2	$\sqrt{2}$	x	$\sqrt{2}$	$\sqrt{8}$	5.656854	1.414214
3 3	$\sqrt{8}$	$\sqrt{2}$	x	$\sqrt{2}$	5.656854	1.414214
4 4	$\sqrt{18}$	$\sqrt{8}$	$\sqrt{2}$	x	8.485281	2.121320

clustering. Units are shown as squares, the numbers denote the number of data points associated to a unit. This is meant as an example how clusterings can differ across dimensions (lyrics and audio features in this case).

Tables 4.2 and 4.3 show the average distance values resulting from examples e and f of Figure 4.2. The corresponding average distance values are

$$avgdist(e) = \frac{.853553 + .853553 + .853553 + .853553}{4} = .853553$$

and

$$avgdist(f) = \frac{2.121320 + 1.414214 + 1.414214 + 2.121320}{4} = 7.0711$$

Table 4.4 shows the values obtained for the density ratio and average distance ratio that are obtained from the clusterings in Figure 4.2. These clusterings only consist of four data points, hence all weighting by the number of instances per unit is omitted for reasons of simplicity. Both the density ratio and average distance ratio give a fair measure of scattering across clusterings. The clusterings a, b as well as c, d have

Table 4.4: Scatter measures for *Self-Organising Maps* (see Figure 4.2). Note, (a) denotes the audio clusterings a, c, e and g; (l) the lyrics clusterings b, d, f and h. **AC** and **LC** denote the contiguity ratios for audio and lyrics, respectively

Maps	$avgdist(a)$	$avgdist(l)$	**ADR**	**AC**	**LC**	**CR**	**ADR×CR**
a,b.	3.4142	3.4142	1	1	1	1	1
c,d.	3.4142	3.4142	1	1	1	1	1
e,f.	3.4142	7.0711	.4828	1	4.9497	.2020	.2020
g,h.	6.1992	8.1411	.7615	5.1992	7.1411	.7281	.5544

coefficients of .5 and 1, respectively, whereas the values for clustering e, f are lower. Visually the clusterings a, b as well as c, d are equal, even if not mapped to the same parts of the map (there is no semantic interpretation possible for different areas of the map, in fact, there is no way of telling differences in terms of clustering position).

A possible visualisation for those values is the colour-coding (binary) of all units on a map within $avg(dist) \times w$ from the centre of the units (average coordinates). All units, except outliers, within one class would be clearly distinguishable from the rest, backing the linkage visualisation introduced at the beginning of this section.

4.2.3 Recap

In this section we showed possible techniques for the multi-modal visualisation of audio collections based on *SOMs*. Both lyrics and audio data were taken into account in order to provide a three-dimensional visualisation of audio tracks and their relations to each other. We also showed how this visualisation can be used to derive quality measurements for multiple *SOM* clusterings on toy examples; a large scale evaluation is to follow in Chapter 6.

4.3 Multi-Modal Genre Classification

Musical genre classification or the labelling of songs according to predefined genre categories is a classic machine learning task. We will use a subset of the data sources introduced in the last chapter, namely audio features and lyrics data as input space. To the end of classification we will use Support Vector Machines, a standard machine learning technique.

Experimental evaluation will be outlined in Chapter 6.

4.4 Where Do We Go from Here

We theoretically introduced the main categories of techniques used in this thesis. An implementation for multi-modal similarity ranking and visualisation will be introduced in the following chapter, quantitative evaluation of these concepts will be done in Chapter 6.

Chapter 5

Implementation Details

The conceptual methods introduced in the last chapters were implemented to allow for experimental evaluation, this chapter gives an overview of the resultant implementation. The implementation comprises two components:

- *Atlantis* is a text mining application, combining textual information for music data from different modalities such as artist descriptions and song lyrics. Further, it contains a user interface and back ends for music similarity retrieval.
- *Sovis* (*Self-Organising Map* visualisation) implements all aspects related to visualisation. A GUI component allows user access to multiple clusterings and a back end component evaluates clusterings.

5.1 *Atlantis*

The Javadoc API for the entire *Atlantis* project is available at `http://www.ifs.tuwien.ac.at/~neumayer/atlantis/api`.

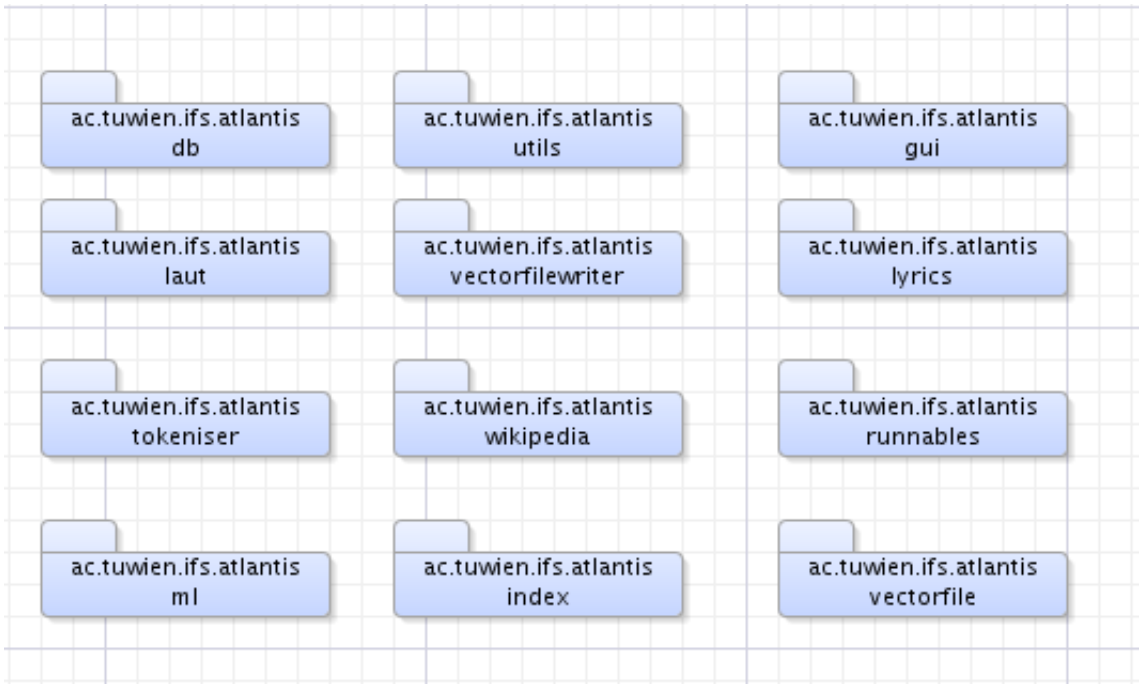


Figure 5.1: Overview of *Atlantis*' packages

5.1.1 Packages of Particular Interest

This section will explain some classes of the most relevant packages within this project in more detail as well as indicate which parts they belong to. Figure 5.1 shows an overview of the Java packages in the *Atlantis* implementation, some of which will be explained in more detail in the following.

5.1.2 Database Binding

The most important DB related classes are shown in Figure 5.2. The DBManager Singleton class is responsible for connecting to the DB and sharing of the connexion. The Corpus class represents one text corpus, e.g. one collection of song lyrics or artist descriptions. This corpus concept is vital to the application since all grouping of documents and classes is organised by corpora. Once documents are indexed, the

unique list of words is calculated and the rest of the database schema is filled with document term assignments. The *MusicCollectionManager* classes provide access to a music collection's metadata information. It also provides access to the classes in *atlantis.db.musicmetadata*'s mapping classes like *Artist*, *Track* or *Genre*.

Figure 5.3 shows the classes used for document representation. A *Document* is the superclass for all document representations providing means for accessing a document object's original as well as preprocessed text values (stored in the respective *textValue* and *rawTextValue* fields). The basic idea is to implement the abstract *Document* class' *preprocessAndTokenise* method in a different way for each document type.

5.1.3 Internet Text Mining

Figure 5.4 shows the class diagrams for lyric fetching and parsing. The aforementioned classes work with local snapshots of Wikipedia and *laut.de*. Lyrics fetching is done just in time over the Internet. Therefore, every class has a static host address, e.g. `http://www.sing365.com` for the sing365 lyrics portal. Further, every class implements the *constructSearchURI* method, which returns the correct URI for the given artist and track name. The content from these URIs is then retrieved from the web and is preprocessed accordingly, i.e. exactly the same way as in the general document cases.

5.1.4 Feature Selection

Feature selection is implemented as part of the vector or matrix generation. Figure 5.5 shows the main classes for frequency thresholding and Information Gain matrix generation. The VectorGenerator class offers the most generic methods to retrieve a single document vector or matrix for sets of documents. The composition of these matrices is done in the individual classes *LowerFrequencyThresholdingMultipleCorporaVectorGenerator* and *InfoGainMultipleCorporaVectorGenerator*. The Information Gain implementation computes the information gain for all tokens found in a specific set of

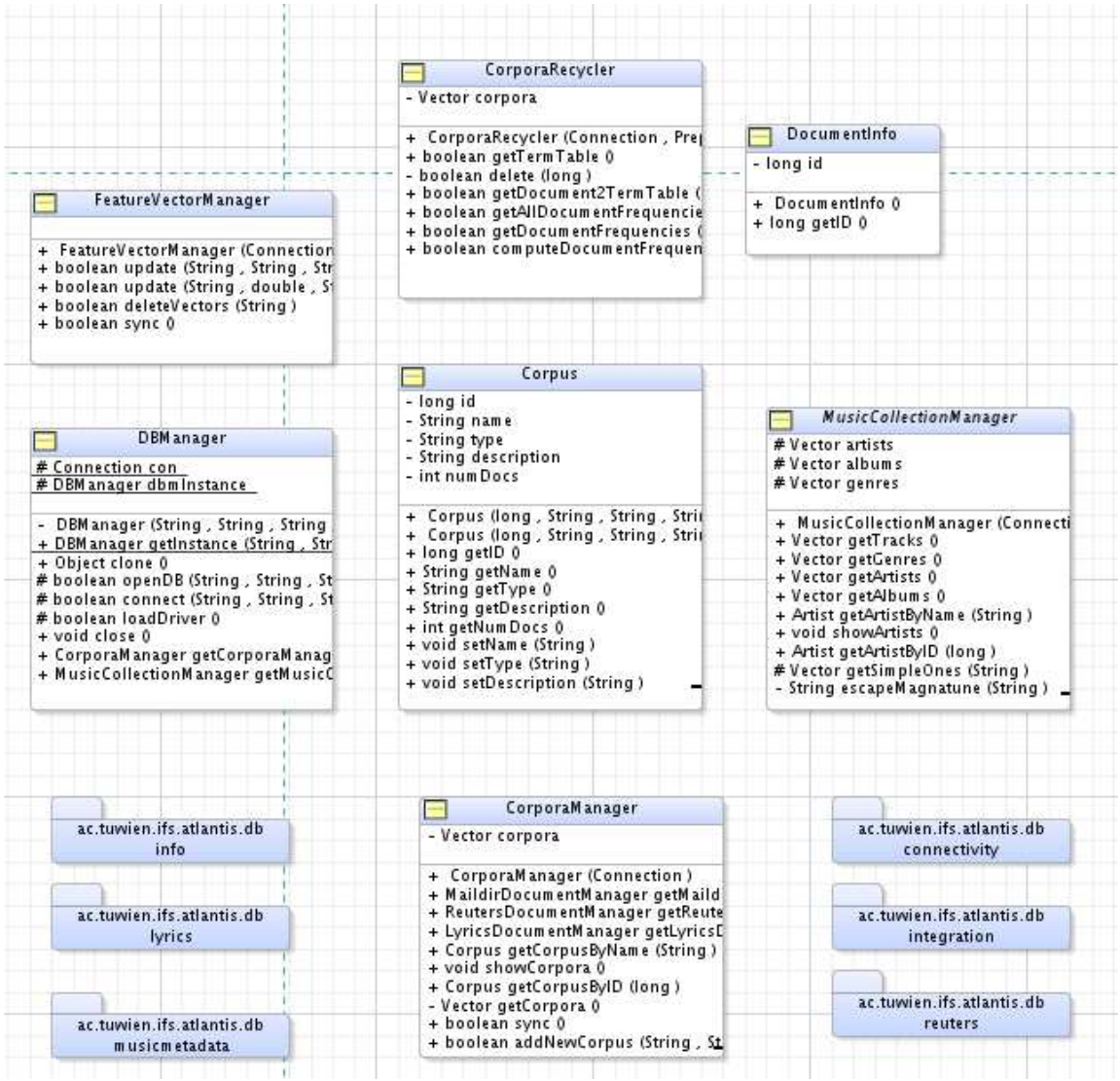


Figure 5.2: Classes for the management of corpora within the framework

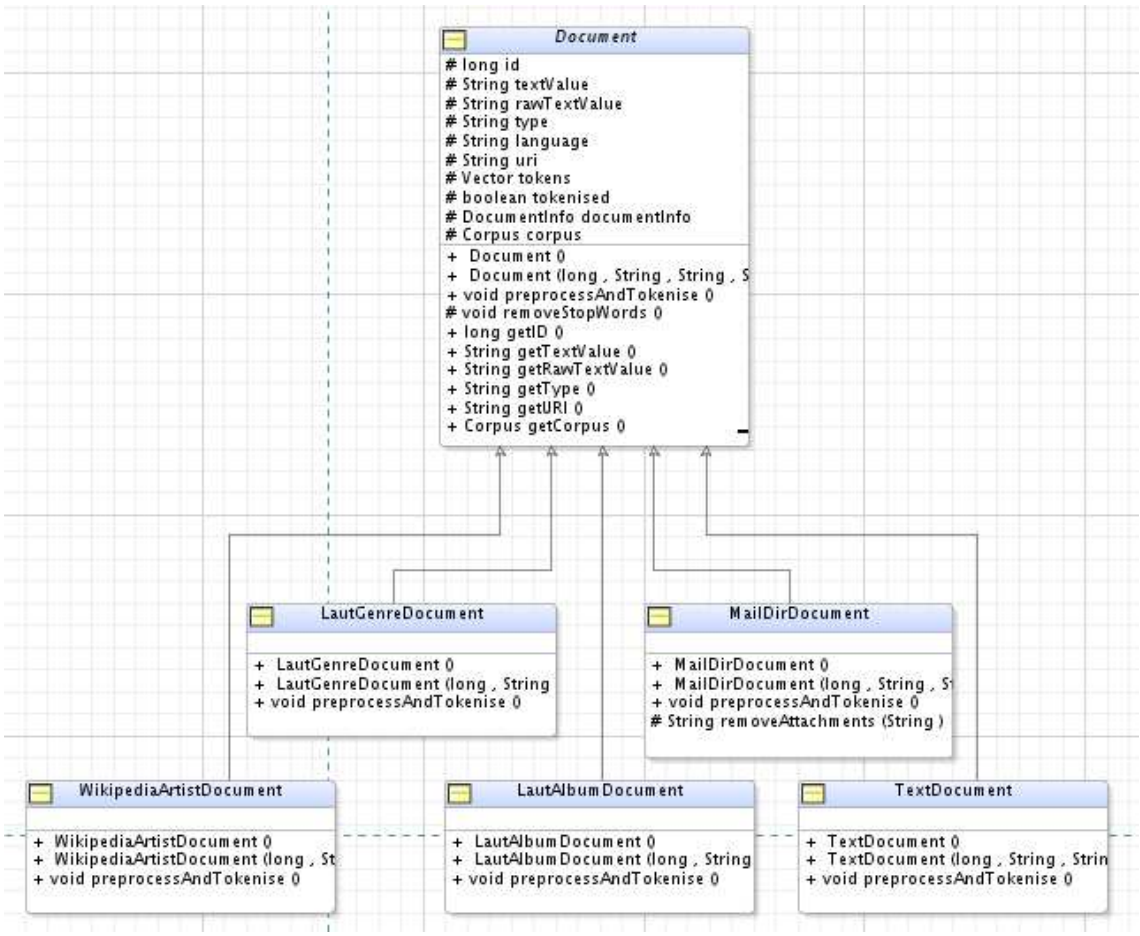


Figure 5.3: Classes for the representation of various documents

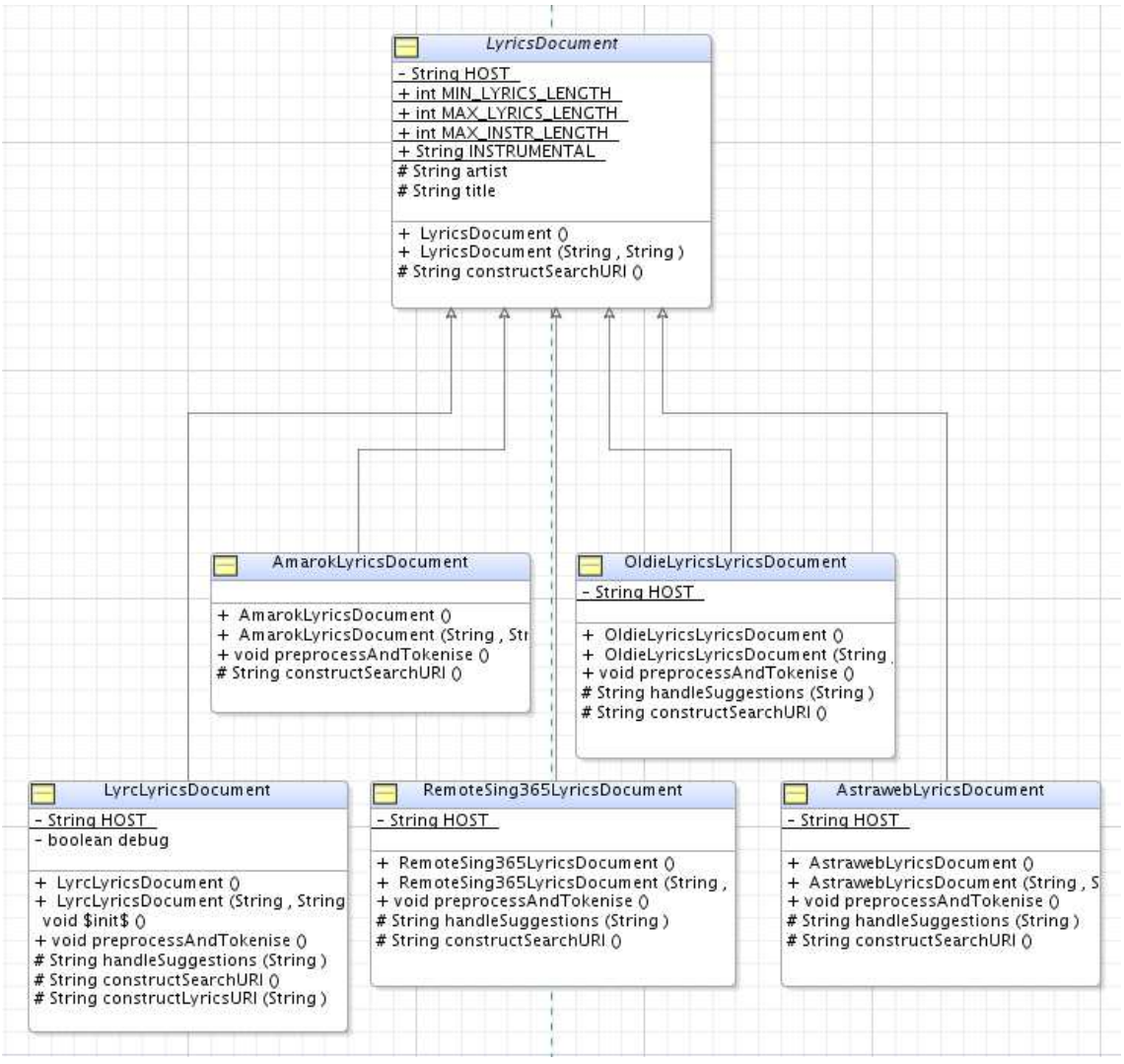


Figure 5.4: Lyrics fetching and parsing - the *Atlantis* way

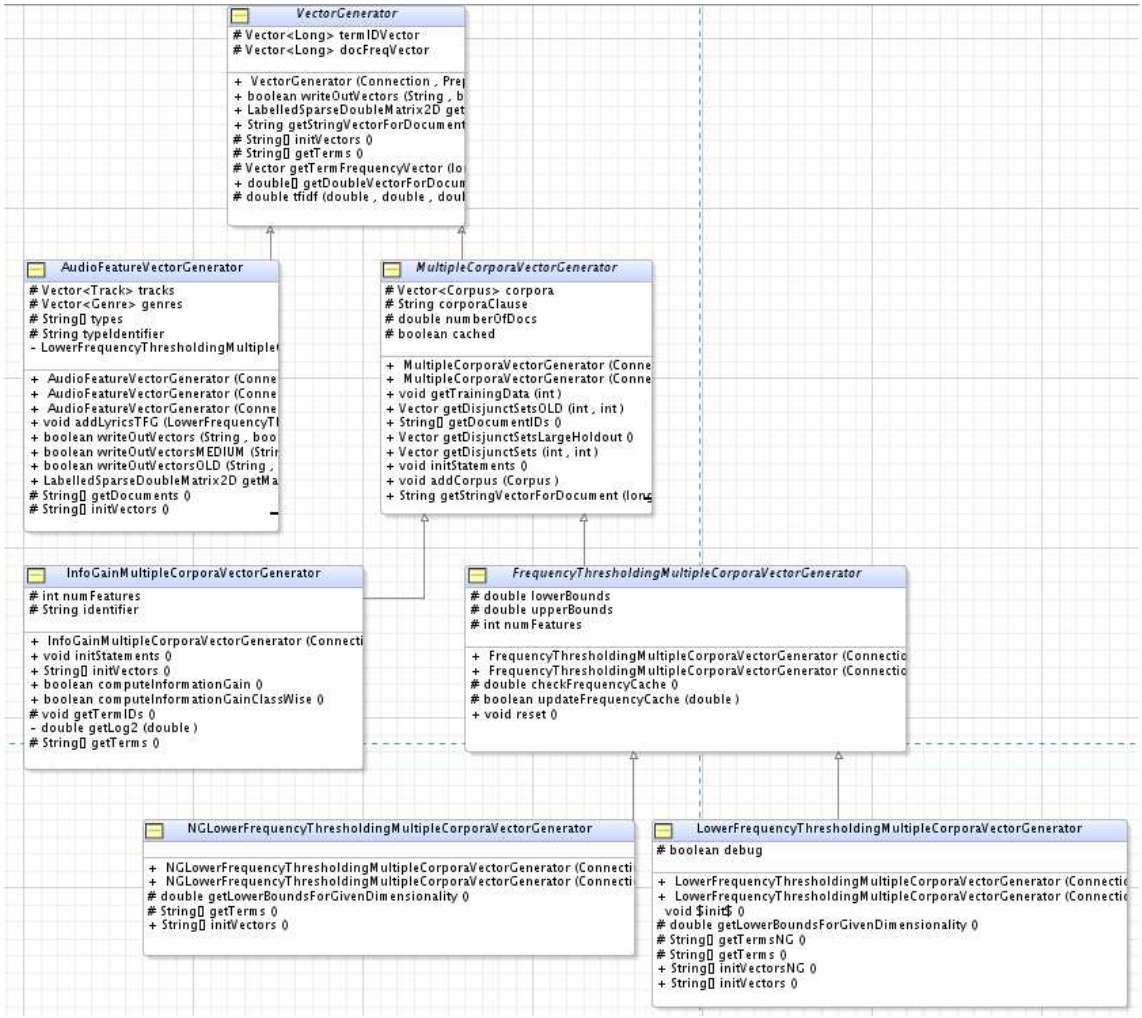


Figure 5.5: Classes for vector generation and dimensionality reduction of text corpora

corpora and stores these values until they are needed for matrix generation. The frequency thresholding is computed every time a matrix is requested. The upper threshold is fixed and set to .5, whereas the lower threshold is set to .01 at the beginning and incremented iteratively as to match the required dimensionality.

5.1.5 Import Export Component

The export component mainly covers exports to various exchange formats used in machine learning. Bindings are implemented to write files in the SOMLib format [43]. Classinfo files are used to store class information for instances, a vector file contains the vectors itself, and a template vector file holds information about the single features (e.g. tokens for text). Further the ARFF file format, which is used by the Weka machine learning suite [57], is supported. Moreover plain text files can be written out for further processing in Matlab.

Further files in SOMLib format can be imported if they contain any of the following feature sets:

- rp, the rhythm patterns feature set (dim 1440)
- ssd, statistical spectrum descriptors (dim 168)
- rh, rhythm histograms (dim 60)
- bpm, beats per minute (dim 1)

In the ideal case, *Atlantis* holds all of this information about a song and plus information about text data terms of $tf \times idf$ vectors for the following dimensionalities:

- Song lyrics
- Artist biographies
- Album reviews
- Genre descriptions

The main music-related import/export component handles data from the Amarok music player [41]. Amarok is a music management application for the KDE desktop. It supports not only the indexing of music files, but also lyrics fetching for the song that

is currently playing via scripts, as well as support for the community site last.fm [21]. Amarok was chosen because it saves many aspects in its database and offers promising features like its last.fm support, which might be interesting in the future. Currently, *Atlantis* supports song, artist, album, genre information as well as song lyrics imports from an existing Amarok database. Moreover, once *Atlantis*' lyrics fetching is done, it is possible to re-export the lyrics information to Amarok.

An overview of various distance measures, criteria for comparing vectors, is given in Figure 5.6. All of *Atlantis*' similarity experiments as well as all distance calculations relating to *Self-Organising Maps* use the Euclidean distance in order to provide distances (or similarity) between documents and vectors. Both the Euclidean and the Manhattan or City Block distance are forms of the more general Minkowski distance in terms of a different exponent, $p = 1$ for the Manhattan distance, $p = 2$ for the Euclidean distance. Normalisation is performed in the *Normalisation* class, implementing a simple MinMax normalisation, i.e. every value is divided by a vectors maximum value. This results in vectors scaled from zero to one. Further, utility methods for converting from String to double vectors et v.v. are provided.

The various ranking mechanisms used are depicted in Figure 5.7. A *SimilarityRanking* basically is a sorted, two-dimensional matrix, instances being listed along its y, features along its x axis. Furthermore, a *CombinedRanking* is a combination of rankings for album, artist, genre, and track, as well as lyrics rankings. The *substituteXXX* methods implement the substitution strategies presented in Section 4.1.1. Besides, normalisation is done for all rankings to guarantee their comparability.

5.1.6 Typical *Atlantis* Usage

The typical usage of *Atlantis* would consist of the following steps:

- Import collection database (from Amarok)

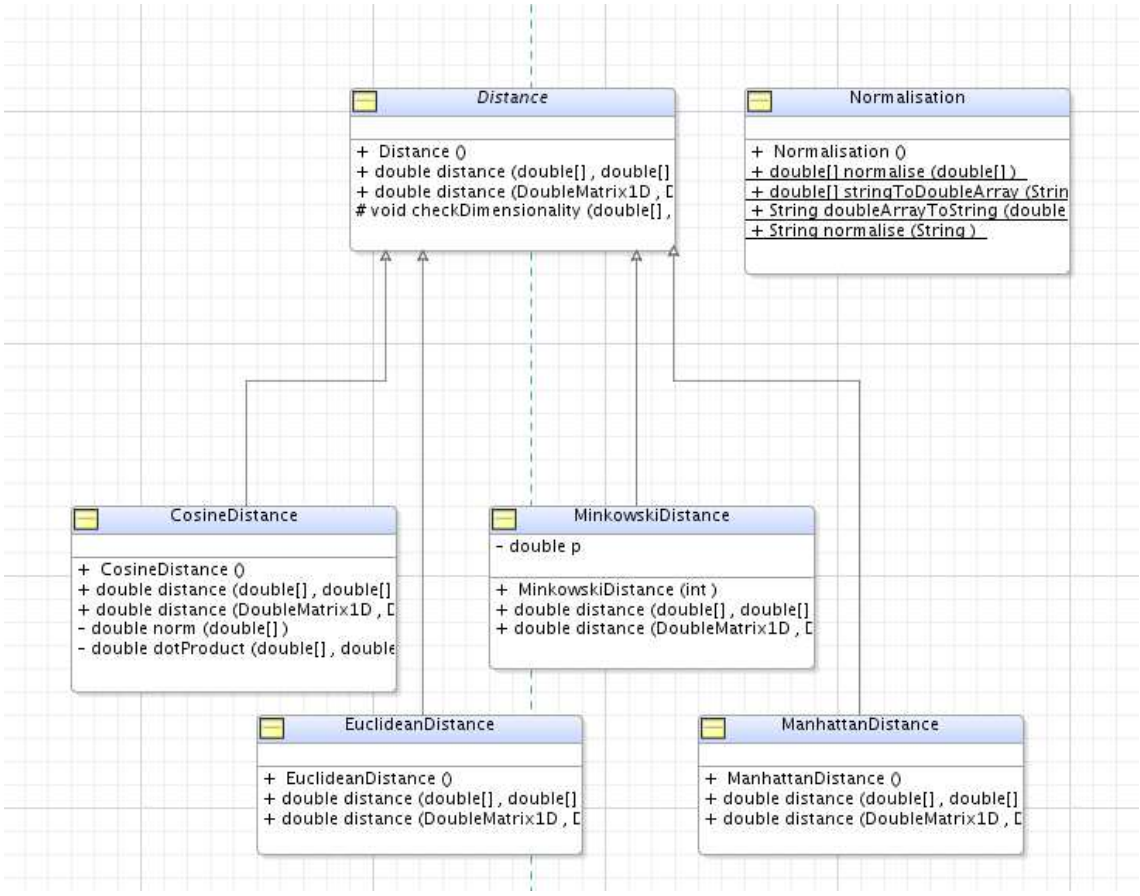


Figure 5.6: Overview of distance measures used in *Atlantis*

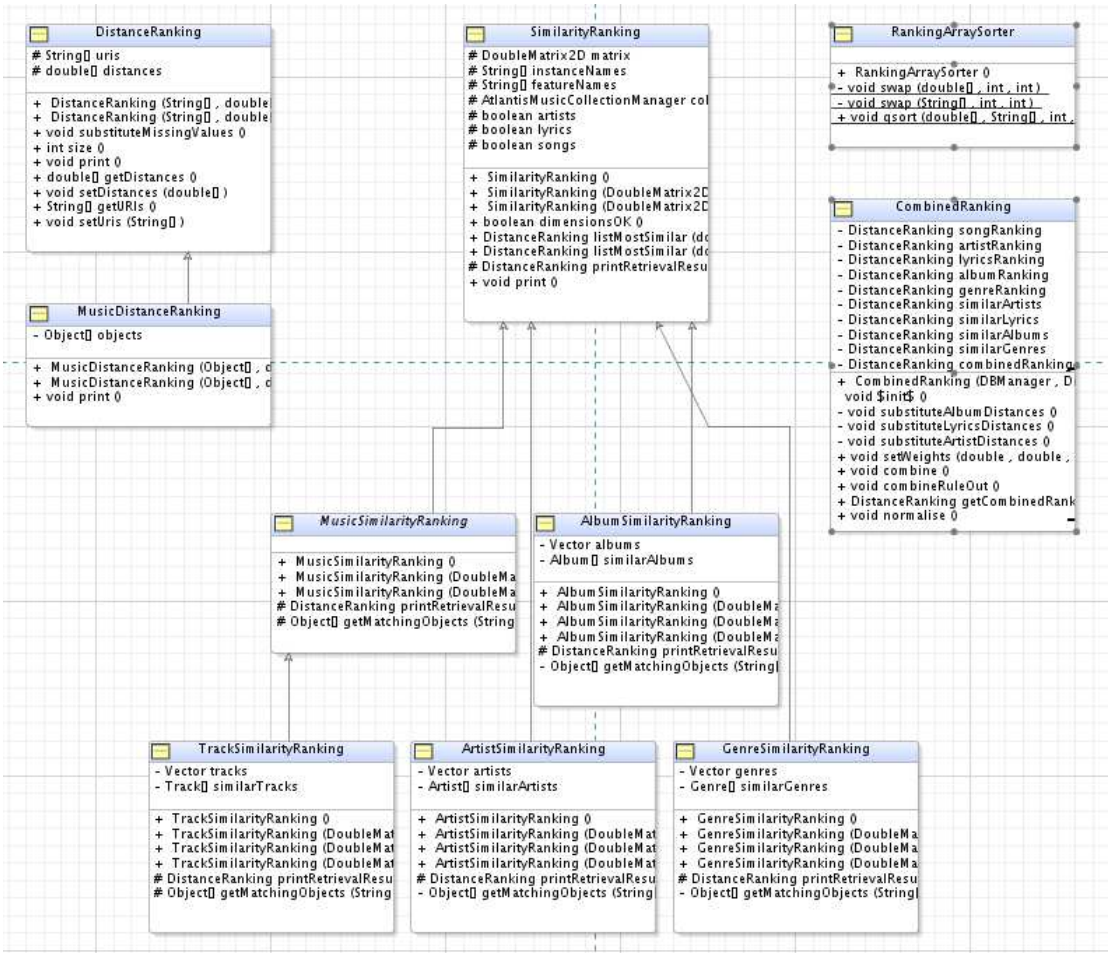


Figure 5.7: Overview of the ranking implementations

- Fetch lyrics

  Interactively check files

  Possibly export the fetched lyrics to Amarok

- Import audio features (from SOMLiB files)
- Batch update text corpus
- Export vector files, browse by similarity, etc.

5.2 *Sovis* (*Self-Organising Map* Visualisation)

Subsequently, *Sovis*, an application prototype for multiple *Self-Organising Maps*, was implemented for the simultaneous display of two music maps. *Sovis* uses *Atlantis*' data model and interfaces for music collection management and the link to metadata. Once connections are drawn on the maps, the connections between units are coloured according to their number of connecting units. The main idea is to allow users to select songs on one map. All selected songs are highlighted on the other map. On top of the interactive user interface and the connexion visualisations, *Sovis* implements the multi-modal quality measurements introduced in section 4.2.2.

The *Sovis* prototype allows for selection of:

- Genres
- Artists
- Tracks

All selections are organised hierarchically according to the songs' artist or genre tags, i.e. further selection refinements are possible. If the user selects, for instance, all songs from the rock genre, all songs belonging to that genre are connected in the interactive 3D display of the *Self-Organising Maps*. Moreover, all single songs of that

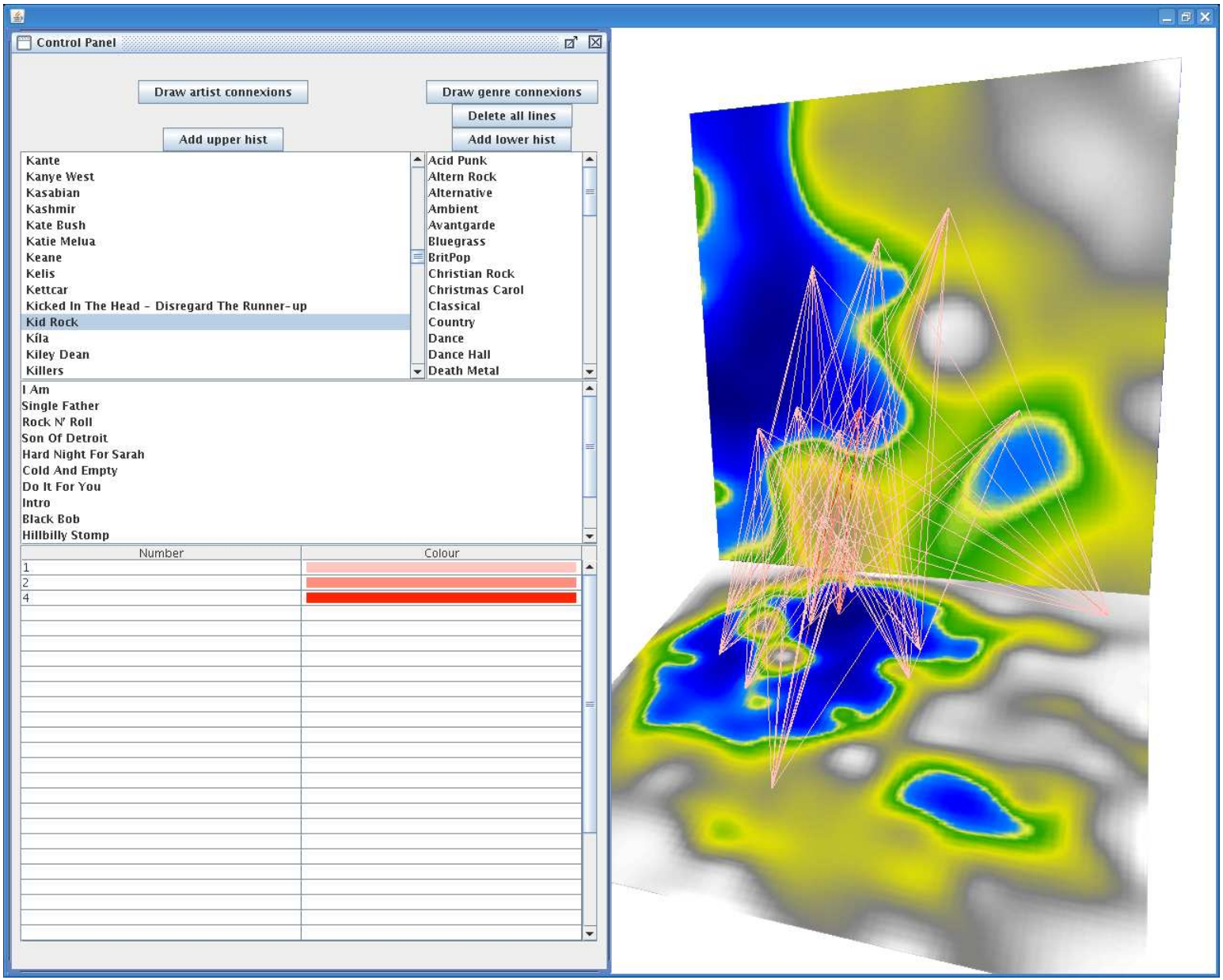


Figure 5.8: Full view of the visualisation prototype. The vertical map clusters songs by audio features, the horizontal map is trained on lyrics features. The left hand side is occupied with various selection controls

particular genre are displayed and the user can further refine his selection to a particular set of songs. The main user interface is depicted in Figure 5.8. The right part of the application is occupied by the display of the two *Self-Organising Maps*. The 3D display offers ways to rotate the view as well as pan and zoom in or out. Controls to select particular songs, artist or genres are on the left side together with the palette describing the associations between colours and line counts. Selections of artists or genres automatically update the selection of songs on the left hand side. Several visualisations for single *Self-Organising Maps* have been proposed. In this work we use the *Smoothed Data Histograms* technique to colour-code the *Self-Organising Maps* [39]; whenever class distribution is of interest, we make use of the *Thematic Class Map* and *Chess Board* visualisations to emphasise the regions covered by different classes. The SOMToolbox application for 2D clusterings supports a wide range of additional visualisations that could be used as a basis for 3D visualisations, as proposed in this thesis. We relied on the same visualisation method for both audio and lyrics features. Of course, this is not necessary and different visualisations could be deployed for the respective feature spaces and clusterings.

Figure 5.9 depicts *Sovis*' main classes and GUI as well as *Self-Organising Map* packages. The *SwingInterfaceMain* class is the main entry point for the GUI application. *QualityMain* evaluates two clusterings in batch mode.

Figure 5.10 shows *Sovis*' GUI components. *SwingInterface* uses both *Atlantis* elements and the *CrossMapLinkageVisualisation* class and presents the main GUI component, handling the display of links between mappings itself. *CrossLinkageVisualisationCrontrol* encapsulates the functionality for loading and displaying trained *Self-Organising Maps* and *CrossLinkageVisualisationCrontrolFrame* holds control elements and user input fields. The *ColourXXX* classes handle the display of the colour palette.

Sovis' functionality to management and evaluation of multiple *Self-Organising Maps* is shown in Figure 5.11. *SOMQuality* implements the computation of the quality measures introduced in section 4.2.2. The *Self-Organising Map* grid and methods for

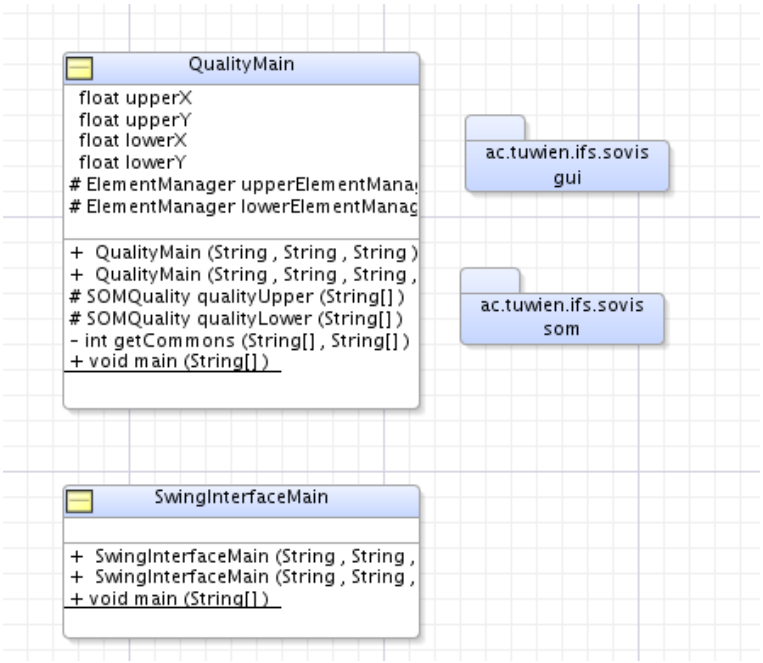


Figure 5.9: Overview of the *Sovis* implementation

accessing mapping and unit information can be found in *ElementManager*.

5.3 Recap

This Chapter introduced the *Atlantis* and *Sovis* Java implementations. Their back end implementations and user interfaces will be used to experimentally evaluate the concepts described earlier on. Multi-modal clustering as well as similarity ranking experiments will be performed exclusively using these implementations, for musical genre classification the files produced by the export components will be used as input for the Weka machine learning suite.

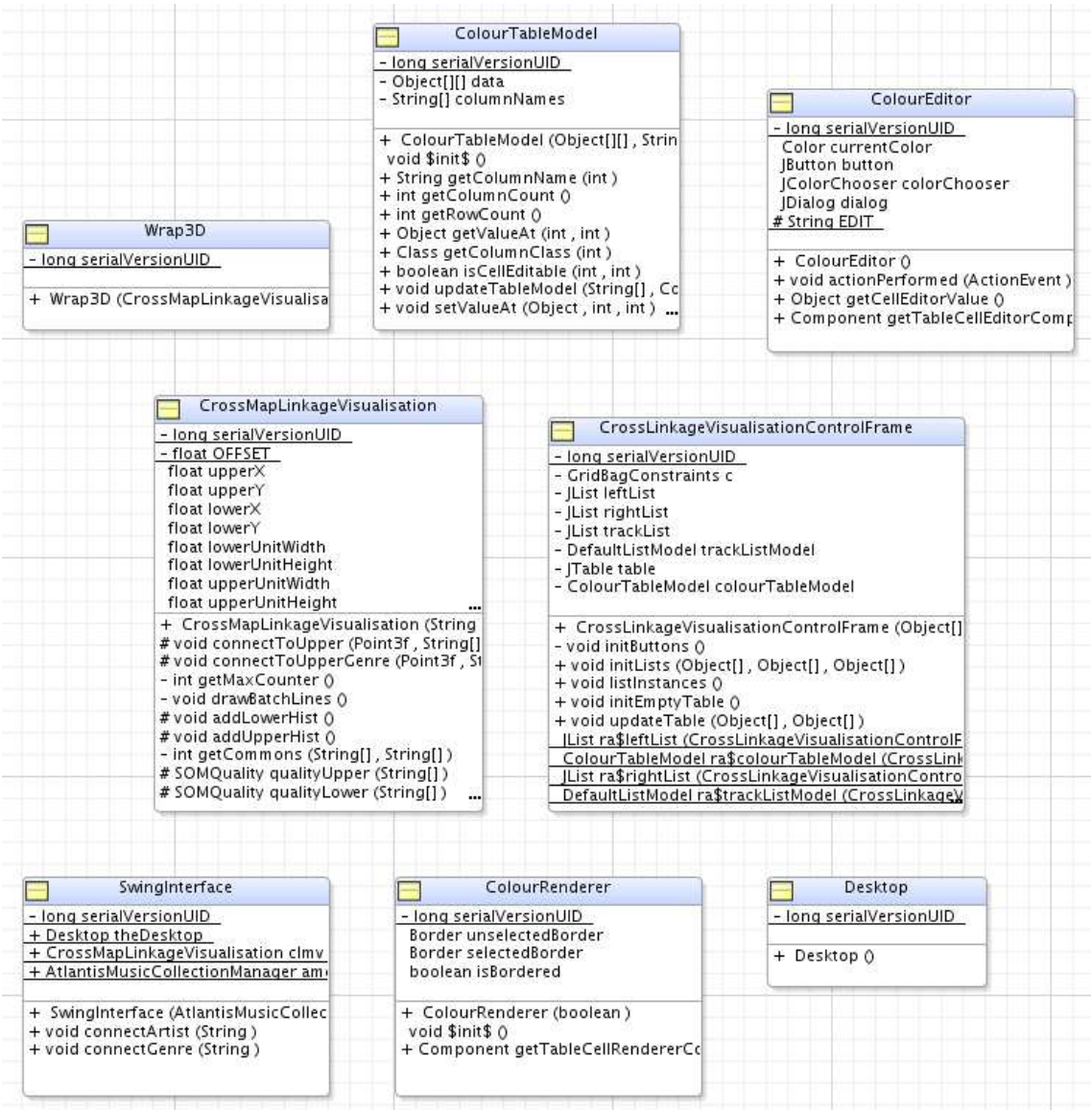


Figure 5.10: *Sovis*' GUI components

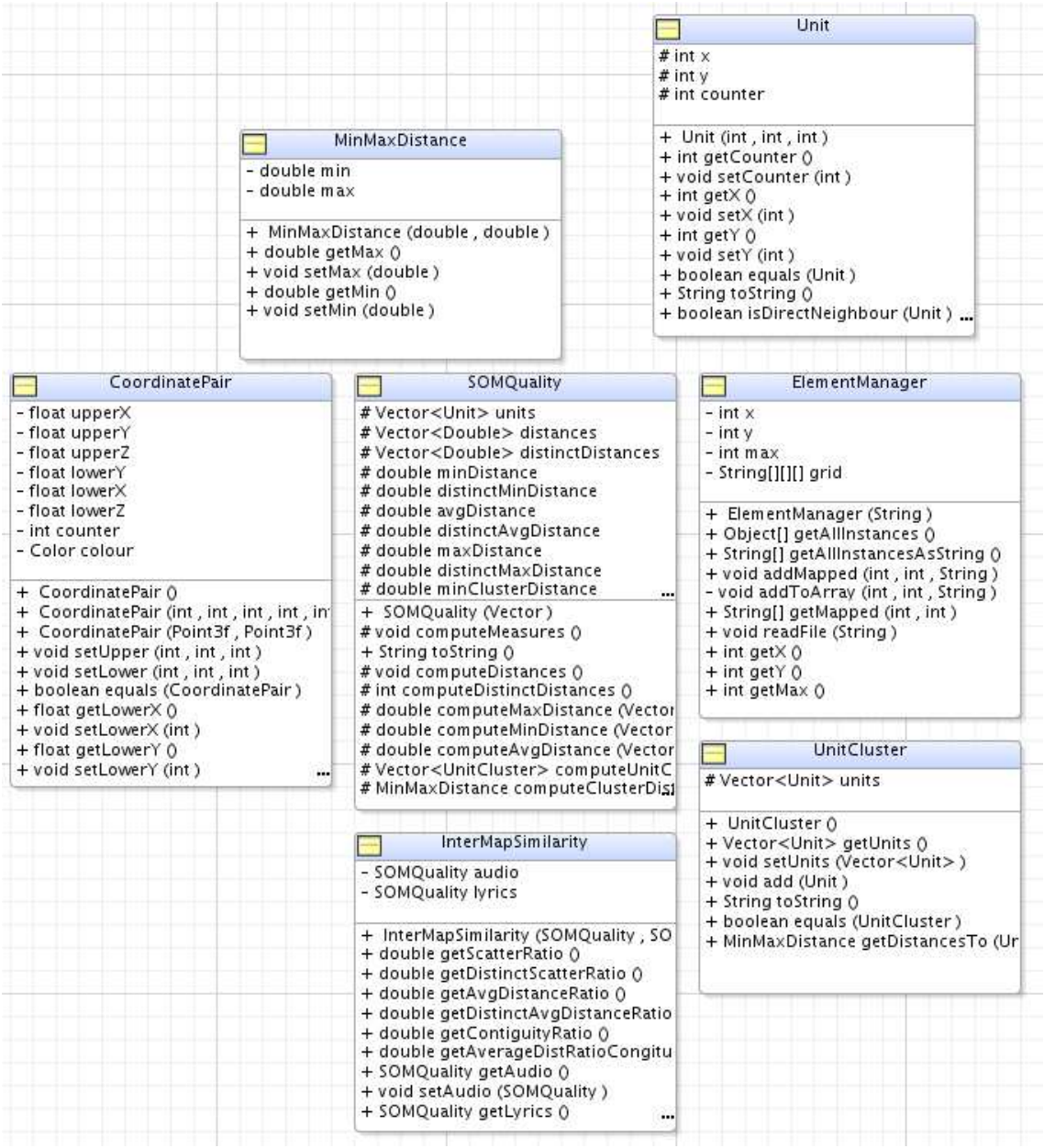


Figure 5.11: An overview of *Sovis*' quality measures

Chapter 6

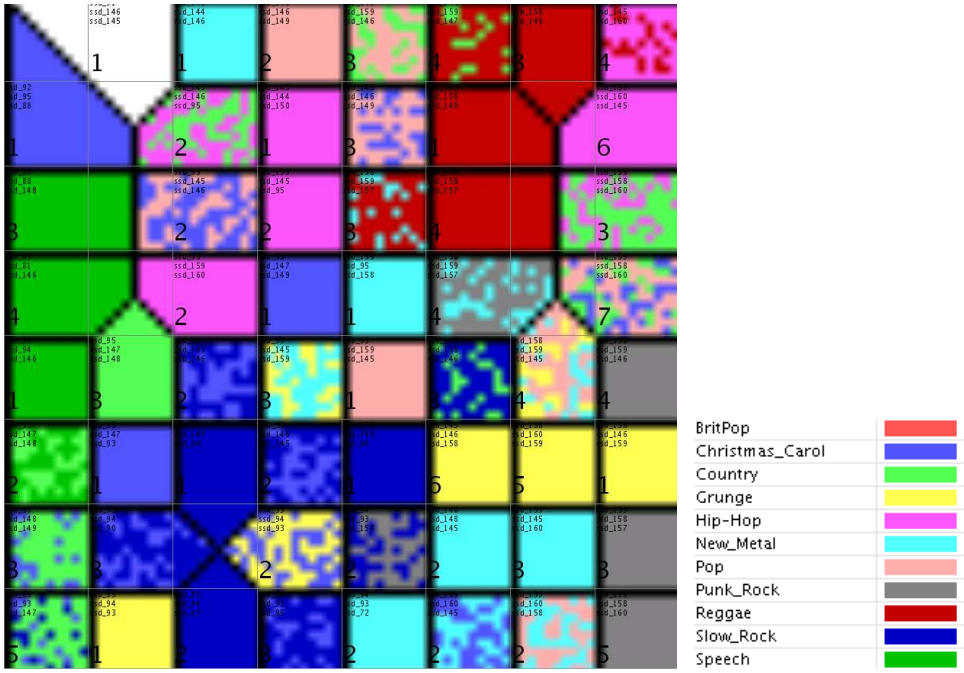
Experiments

> Love? Truth? Beauty? I prefer negotiable securities.
> Doge Miskich var Miskich, "All About Me", 301 AFC

This chapter describes the experimental setting and provides experimental results for the three main tasks considered in this thesis, namely multi-modal

- similarity ranking and retrieval,
- multi-modal visualisation and cluster validation, and
- musical genre classification.

At first, experiments are shown on the small data collection, particularly focussing on visualisation. After that, a full set of experiments is performed on the large collection, including ranking, cluster visualisation, and musical genre classification, which is much more feasible for collections of sufficient size.



(a) Clustering of audio features for the 10 genres subset of the audio collection

(b) Class (genre) colour legend

Figure 6.1: Thematic class map visualisation for the audio clustering of the 10 genres subset of the small audio collection. Genre colours are displayed in the legend

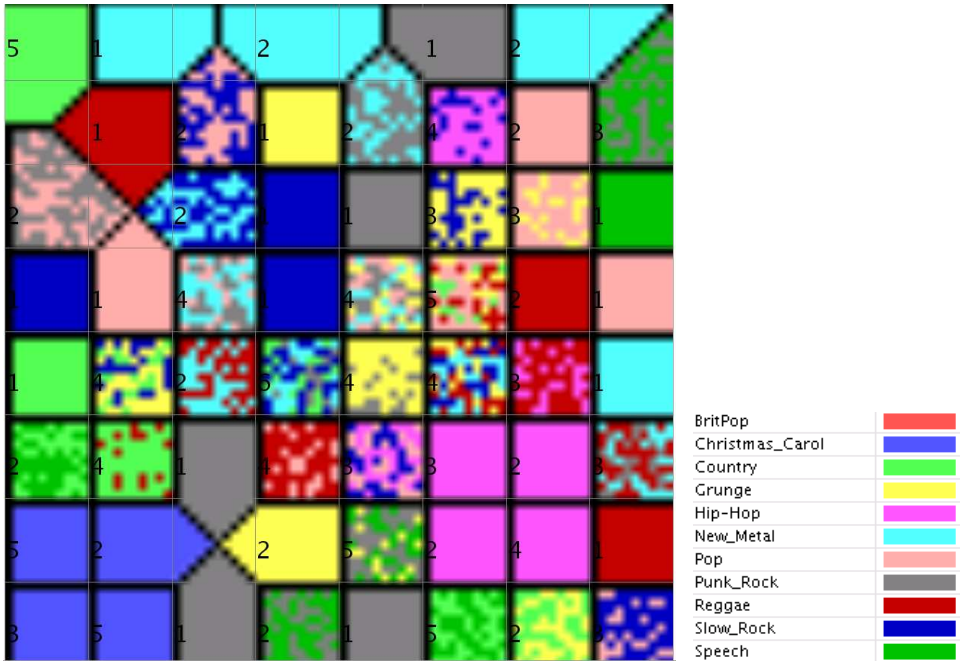
6.1 Small Collection Experiments

The experimental results presented in the following were obtained from experiments made with the small data collection, introduced in Section 3.1.1.

6.1.1 Clustering According to Audio Features

For each song lyrics features as well as audio features (*Statistical Spectrum Descriptor*, dimensionality 168) were computed. The *Self-Organising Map* clustering was finally performed on the small data set. We then trained two *Self-Organising Maps* of size 8 × 8, i.e. 64 units, one on the audio feature set, one on lyrics.

Figure 6.1 displays the clustering of the small collection according to audio features



(a) Clustering of lyrics features for the 10 genres subset of the audio collection

(b) Class (genre) colour legend

Figure 6.2: Thematic class map visualisation for the lyrics clustering of the 10 genres subset of the small audio collection. Genre colours are displayed in the legend

plus class legend. Different areas of the map are coloured according to their genre. The class legend is given in 6.1(b). Such a visualisation makes it easy to comprehend the distribution of classes on the map. The 'Reggae' genre (dark red) for example is located on the right upper part of the map, clustered on adjacent units only. 'Christmas' songs (light blue), on the other hand, are spread all over the map. This corresponds to the very differently sounding nature of these two genres. 'Christmas' music is rather defined by its lyrics, whereas 'Reggae' is rather defined by its typical sound. Songs belonging to the 'Punk Rock' and 'Speech' genres both are also rather defined by their sound.

6.1.2 Clustering According to Lyrics Features

The same collection clustered according to song lyrics is shown in Figure 6.2. The resultant high-dimensional feature vectors were further downscaled to 905 dimensions out of 5.942 using feature selection via document frequency thresholding, i.e. the omitting of terms that occur in a very high or very low number of documents. We therefore excluded terms occurring in less than 16 per cent and more than 40 per cent of the documents.

Amongst the most obvious differences are the better separation of 'Hip-Hop' songs in the upper right part of the map. This genre is easily identified by terms like 'shit', 'rap' or names of different rappers. Christmas carols are clearly separated in the lower left corner of the map, exclusively covering four units. Tracks belonging to the genres, 'Slow Rock', or 'New Metal' are spread across large parts of the map, reflecting the diversity of topics sung of within them.

6.1.3 Combined, Multi-Modal Visualisation

Figure 6.3 shows the prototype implementation's tool section as well as its visualisation part. On the right hand part of the illustration two clusterings are visualised simultaneously. These clusterings are subsequently subject to quantitative evaluation according to quality criteria introduced in Section 4.2.2.

Table 6.1.3 lists these quality measures for all the genres in the small collection. Exceptionally high values for the ADR×CR were, for example, calculated for the 'Pop' and 'Hip-Hop' genres, meaning that they are rather equally distributed across clusterings. 'Pop' songs, for instance, are equally distributed in terms of audio and lyrics contiguity, leading to the maximum value for LC. 'Christmas Carol' songs have an exceptionally low value, stemming from the fact that they form a very uniform cluster on the lyrics map, the contiguity value is therefore set to one. On the audio map,

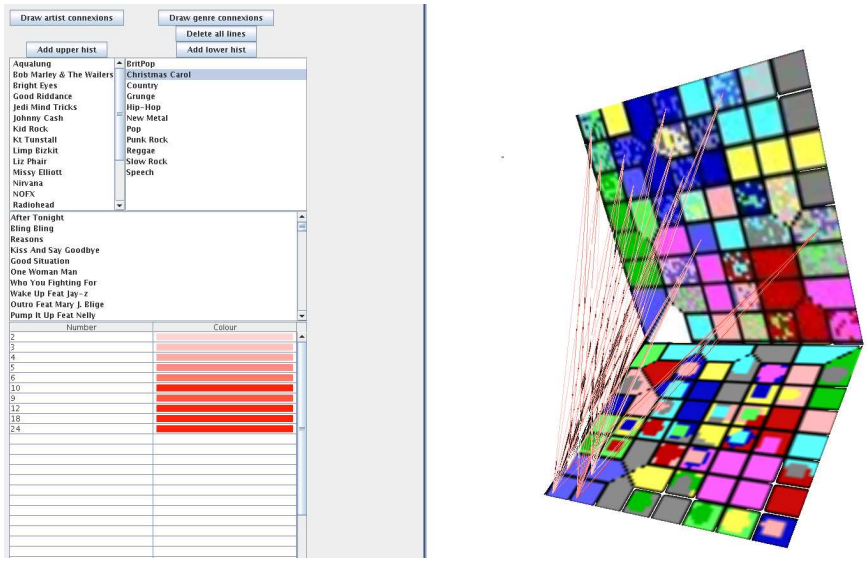


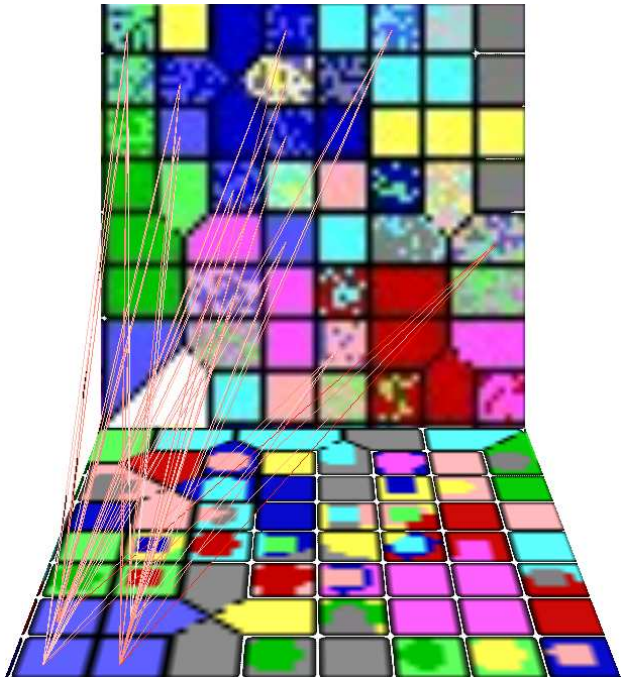
Figure 6.3: Overview of the visualisation prototype. The left part of the application is occupied by tools to select songs from the audio collection. The main part displays the clusterings and connections in between

Christmas carols are spread well across the map. Other low values can be identified for 'Punk Rock' or 'Speech', both of which are more spread across the lyrics than the audio map.

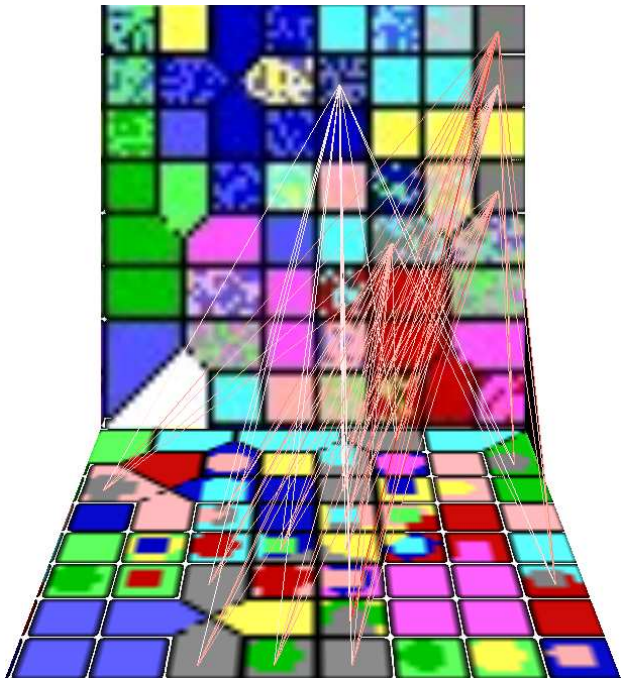
Figure 6.4 shows two examples of genre connections. Figure 6.4(a) shows the connections for all songs belonging to the 'Christmas Carol' genre, and visualises its distribution as mentioned in the previous paragraph. Songs belonging to the 'Punk Rock' genre are shown in Figure 6.4(b). The strong divergence of distributions is clearly visible.

6.2 Large-Scale Experiments

To prove the applicability of the proposed methods, we performed experiments on a larger collection of digital audio, which is described in Section 3.1.2.



(a) Multi-Dimensional visualisation of 'Christmas' songs



(b) Distribution of 'Punk Rock' songs on both maps

Figure 6.4: Distribution of selected genres across maps

Genre	AC	LC	CR	ADR	ADR×CR
Cristmas Carol	*.1240*	**1**	*.1240*	*.2982*	*.0370*
Country	.1644	.2169	.7578	.8544	.6475
Grunge	.3162	.5442	.4714	**.9791**	.4616
Hip-Hop	.2425	.1961	.8086	.6896	.5576
New Metal	.1754	.1280	.7299	.9383	.6849
Pop	.1644	.1644	**1**	.9538	**.9538**
Punk Rock	**.4472**	*.1280*	.2863	.7653	.2191
Reggae	.2774	.1810	.6529	.5331	.3480
Slow Rock	.1715	.1240	.7232	.7441	.5382
Speech	.3333	.1754	.5262	.3532	.1859

Table 6.1: Genres and the according spreading values across clusterings. **AC** denotes the audio contiguity, **LC** the lyrics contiguity, **CR** the contiguity ratio, **ADR** the average distance ratio, and **ADR**×**CR** the product of **ADR** and **CR**. Maximum values are printed in bold font, minimum values italic

6.2.1 Multi-Modal Audio Similarity Ranking

This section contains an experimental evaluation of the techniques for multi-modal similarity ranking in Section 4.1. The main idea is to rank songs in a music collection according to different modalities. We consider the following levels of similarity for each seed/query song:

- Song (audio)
- Song (lyrics)
- Artist
- Album
- Genre

As a next step all the different rankings are merged into one result list, the experiments performed will be explained in the following.

In order to show the importance of the missing values problem, Table 4.1 summarises the coverage of different levels of textual description within the large collection. The evaluation and comparison of the results of content-based (i.e. audio) similarity rankings to combined approaches presented in the Section 4.1 is the central part of the experiments described in this section. To that end, at first, the combined distances for each track in the collection to all other songs are computed. Then the first 5, 10 and 20 results are evaluated according to the number of songs belonging to:

- the same artist,
- the same album, or
- the same genre,

While this kind of evaluation is definitely not the optimal way, it constitutes an objective, automated way of analysing results that has been used in this setting before [20]. Obviously, this should be followed-up by a user study to establish sound parameter values for real-world retrieval tasks.

Table 6.2 gives an overview of different settings for weightings. Weights are always given for each of the five dimensions and always sum up to one. The sum column denotes the sum of the number tracks in the result set, that are featured on the same album, interpreted by the same artist, and belonging to the same genre as the seed song taken from the top 20 results for every given song[1]. Therefore, the higher the value, the more similar tracks are returned according to that similarity measure. It is shown that additional textual data sources improve the results significantly. Experiment 15 shows very high values and seems to be the best combination in this context, especially

[1]This evaluation for sure has its weaknesses like, for example, a strong bias on albums, because they implicitly convey genre information. We still chose this kind of evaluation instead of large-scale user studies due to time and effort restrictions.

Table 6.2: Results for given weighting strategies. The different weightings are given in the Audio, Artist, Lyrics, Album and Genre columns. The Sum column denotes the sum over the number of songs amongst the top 20 results from the same artist plus album plus genre for each combination

ID	Audio	Artist	Lyrics	Album	Genre	Sum
1	1.0	.00	.00	.00	.00	5.37
2	.50	.50	.00	.00	.00	19.54
3	.70	.30	.00	.00	.00	19.53
4	.30	.70	.00	.00	.00	19.54
5	.30	.30	.30	.00	.00	18.70
6	.70	.30	.20	.00	.00	18.89
7	.25	.25	.25	.25	.00	20.64
8	.70	.10	.10	.10	.00	20.09
9	.40	.25	.10	.25	.00	20.87
10	.40	.30	.00	.30	.00	21.41
11	.40	.00	.30	.30	.00	9.64
12	.20	.20	.20	.20	.20	22.65
13	.60	.10	.10	.10	.10	22.12
14	.40	.30	.10	.10	.10	22.73
15	.30	.30	.00	.20	.20	23.46
16	.30	.30	.00	.10	.30	23.35
17	.30	.30	.00	.30	.10	23.43

outperforming the audio only experiment number one. Of course this may look very different on a per user basis. However, these weightings offer a very good point to start from in ongoing experiments, particularly including user feedback. Naturally, the results according to the chosen evaluation are far more improved by artist, album and genre information than by a song's lyrics.

The values given in Table 6.3 and Table 6.4 show the differences over changes in the substitution strategies as well as initial size of the result set. The weights used for this experiment are .3, .3, .0, .2, and .2, respectively for the audio, artist, lyrics, album, and genre categories. This weighting corresponds to the best result obtained in the ranking experiments (experimental setting 15), which are summarised in Table 6.2. The first set of results are based on a full ranking of all songs, the latter relies on a re-ranking of the first 600 closest songs in terms of audio similarity. The given results are computed as the sums of this evaluation for the 5, 10 and 20 best results. Furthermore, the average count over results for different seed songs was computed. The figures show that penalising of missing values does not improve the quality of the retrieval results, the simple averaging strategy performs better in all respects which is negatively influenced by the low coverage of data, i.e. many similar tracks are without textual information and therefore would not be considered in the result, if it was not for averaging their distance. Surprisingly, category substitution does not improve results at all. Table 6.4 outlines that the results for a subsampled data set decreases performance significantly, but also shows that the ranking based on *Statistical Spectrum Descriptors* selects songs according to criteria decoupled from metadata tags. Category substitution is not available for the full retrieval setup. However, results are provided for a performance improvement over that strategy.

User Interface

Figure 6.5 shows the main user interface of an experimental system to evaluate the impact of different weighting strategies. The largest part of the GUI is occupied by

Table 6.3: Experimental results for similarity ranking experiments using different substitution strategies for the combination of the results taken from a full ranking of all songs. Numbers given denote the number of songs belonging to the same artist, album, and genre as the seed song in the top 5, 10, or 20 songs retrieved

Same Album	**Top 5**	**Top 10**	**Top 20**
Category Subst.	NA	NA	NA
Exclusion done	2.11	3.76	5.88
Simple Avg.	2.17	4.04	6.45
Same Artist	**Top 5**	**Top 10**	**Top 20**
Category Subst.	NA	NA	NA
Exclusion done	3.17	6.09	11.66
Simple Avg.	3.22	6.24	11.90
Same Genre	**Top 5**	**Top 10**	**Top 20**
Category Subst.	NA	NA	NA
Exclusion done	2.77	5.23	9.52
Simple Avg.	2.85	5.50	10.25

Table 6.4: Re-Ranking of top 600 initial results for similarity ranking experiments using different substitution strategies weighting for the combination of the results. Numbers given denote the number of songs belonging to the same artist, album, and genre as the seed song in the top 5, 10, or 20 songs retrieved

Same Album	**Top 5**	**Top 10**	**Top 20**
Category Subst.	1.84	2.79	3.52
Exclusion	1.91	2.78	3.55
Simple Avg.	2.36	3.41	4.07
Same Artist	**Top 5**	**Top 10**	**Top 20**
Category Subst.	2.43	4.09	5.83
Exclusion	2.41	3.95	5.53
Simple Avg.	2.97	5.11	7.15
Same Genre	**Top 5**	**Top 10**	**Top 20**
Category Subst.	1.55	2.85	5.15
Exclusion	1.64	2.87	4.92
Simple Avg.	1.90	2.91	4.43

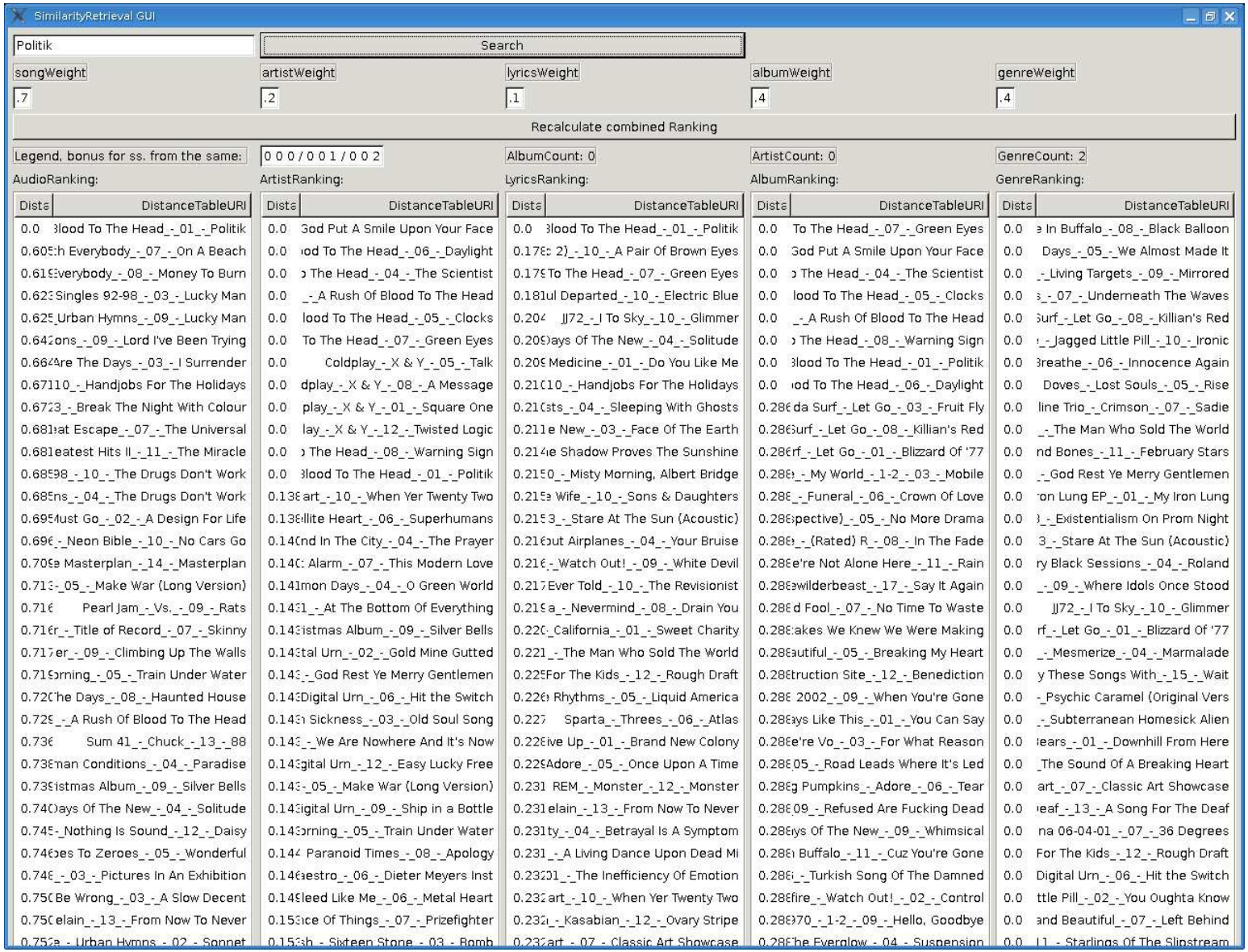


Figure 6.5: GUI for experimental evaluation of different weighting strategies. Weightings are given for the track 'Politik' by 'Coldplay'

the five different rankings, one for audio, artist, album, lyrics, and genre respectively. Only the pre-filtered rankings are shown, i.e. not the rankings according to five different modalities are shown themselves. Instead, each ranking shows the ranking obtained by substitution strategies for all songs. The genre ranking, for instance, shows all songs in the collection ranked by their genre weight, i.e. all songs from a given genre are represented by that genre's term vectors. The weights for each of these sources can interactively be updated and the influence on the combined ranking can be observed. The user can update these weights and instantly see the influence on the combined ranking as described earlier. A textbox is provided to search for song titles, rankings are generated accordingly. The figure shows the query for the track 'Politik' by 'Coldplay.' It becomes evident that the first result is either the song itself or all other songs by the same artist. Every 'Coldplay' song has the same distance (zero) to all other songs of the same artist, whereas the lyrics and audio categories have distances on a song basis. For this song, there's no genre information available ('Slow Rock'), therefore, in terms of genre similarity, all songs have the same distance to the query. For matters of simplicity all distances are set to 0 in this case. It is, however, possible to compute all other four kinds of similarities. In terms of audio features, the most similar songs are mainly songs by 'Richard Ashcroft' or 'The Verve' as well as 'Blur' and 'Oasis'. The most similar lyrics are from songs by 'Coldplay' itself, 'The Cranberries', and 'The Pogues' as seen in the third column. According to the analysis of the artist descriptions the most similar artists are 'The Flaming Lips', 'Bloc Party', and 'The Gorillaz' as well as Conor Oberts's 'Bright Eyes'. Albums with similar reviews are from artists like the Americans 'Nada Surf' or the British 'Badly Drown Boy'. Once the user has set his preferred weights, he can generate an overall ranking based on the single ones. Figure 6.6 shows the combined ranking with the weights .7, .1, .1, .4, and .4[2], for audio, artist, lyrics, album, and genre, respectively. It also shows the updated distances and reveals a new ranking based on all modalities and a user's preference for them (adjusted by the chosen weighting).

[2]These weights were subjectively chosen but provided a good blending of results.

Figure 6.6: Combined ranking for the track 'Politik' by 'Coldplay', based on single rankings in five modalities

One vital aspect of multi-level similarity is that adjusting the weights also means adjusting to the user. Personalisation based on weightings therefore will definitely be evaluated in the future. Relevance feedback could be used to automatically adapt weights according to user input, i.e. those data can be extracted from a user's playlist.

6.2.2 Comparisons of Multi-Modal Clusterings

This section summarises the findings from the multi-modal clustering experiments. We train one map representing the collection in terms of lyric similarity, one in terms of audio similarity. At first, examples of different clustering results for processing based on song lyrics will be given. We then stress the differences between the audio and lyrics space. After that we will provide experimental results of multi-modal clustering.

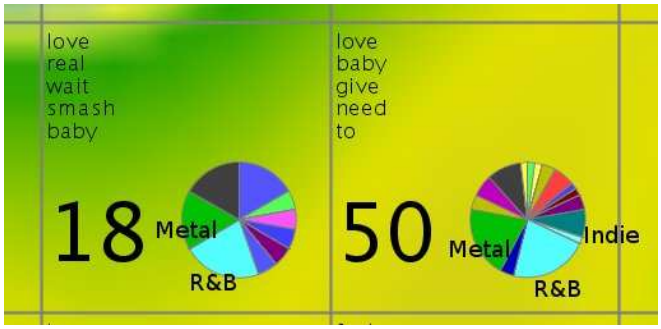


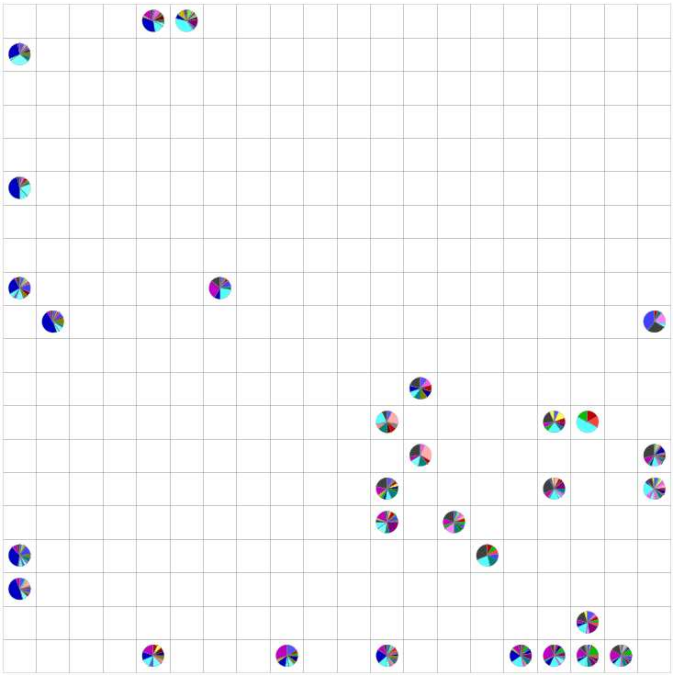
Figure 6.7: Clustering of songs centred around the love topic

Traditional Genres and the Lyrics Space

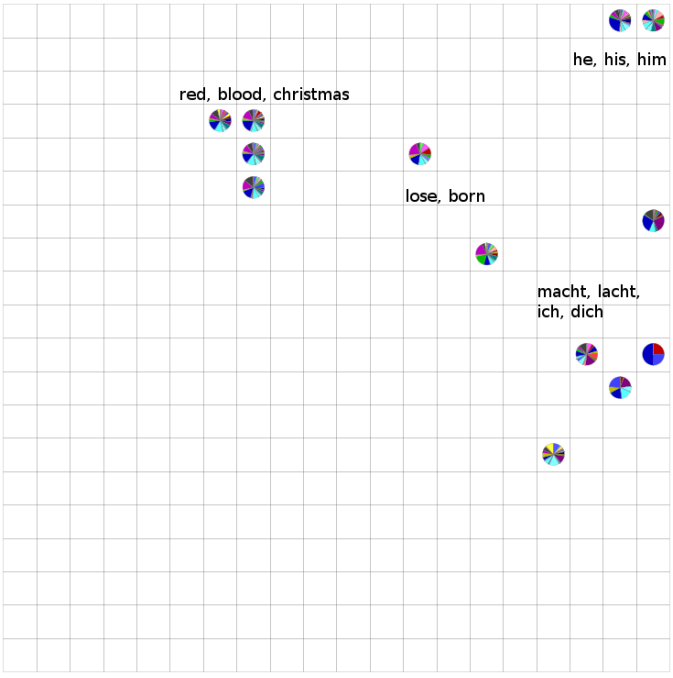
Figure 6.7 shows the distribution of genres on two particular units on a *Self-Organising Map* trained on lyrics data. The pie chart display shows the numbers of songs belonging to the different genres, underpinning the idea that traditional genres are not necessarily feasible for the lyrics space. The labelling of single units is done via the LabelSOM algorithm, i.e. the identification of discriminative components. In this case, the prominent key words 'love', 'baby', 'give', 'real', and 'need' give a very good idea on the main topics of these songs' lyrics. The 50 songs, for instance, mapped onto the right unit of this *Self-Organising Map* are distributed across 16 'traditional' genres, the largest group being 'R&B' songs, followed by 'Metal' and 'Indie'.

Artists whose songs are mapped onto this unit are, amongst others: 'Mary J. Blige', 'Beyonce', 'Christina Milian', as well as 'Wolfmother' or the 'Smashing Pumpkins'. This interesting mapping shows clearly that topics in song lyrics overcome traditional genre boundaries, while pointing out that a categorisation on the lyrics level makes sense since all songs cover similar topics.

To the ends of exploiting the fundamental differences in clusterings we train two *Self-Organising Maps*, one based on audio, one based on text features. These maps will be referred to as audio and lyrics map, respectively. As well as examples are given, experimental results are shown.



(a) Clustering of Christmas carols on the 2D audio map



(b) Clustering of Christmas carols on the 2D lyrics map

Figure 6.8: Distribution of Christmas carols on clusterings for different feature spaces. The pie charts denote the distribution of songs over different genres on the particular units – only units comprising Christmas carols are highlighted

Figure 6.8 shows the distribution of Christmas carols on the two-dimensional clusterings, the distribution on the audio map is shown in Figure 6.8(a), and in Figure 6.8(b) on the lyrics map, respectively. Both maps have the size 20×20, the dimensionality of the audio input space is 168, whereas the lyrics space was downscaled to 6579 out of 63884 dimensions. The respective lower and upper document frequency thresholds used to obtain this dimensionality were one and 40 per cent. In the former case, the 33 songs are mapped onto 30 units, in the latter only onto 13. Hence, the lyrics clustering uncovers information such as vastly different interpretations of one and the same song, that have the same lyrics, but differ greatly in sound. Manually assigned labels demonstrate the different key tokens present on the respective areas of the map, i.e. the 'red / blood / christmas' cluster on the top of the map. Due to the *Self-Organising Map*'s random initialisation, the fundamental differences in lyrics space, and the general training algorithm, the songs are mapped onto different corners of the map. For evaluation the absolute location on the map plays a less important role than the relative distances. However, it is clear that the spread of songs differs from one clustering to the other. In the lyrics space, Christmas carols are clustered more closely to each other, whilst they get spread over more units and occupy a larger space of the map in the audio space. The two interpretations of the song 'The First Noel', for example, are mapped onto one unit in the lyrics space. On the audio map, however, these songs lie on different units on different regions of the map. The artists of the interpretations are the 'Bright Eyes' and 'Saxofour', even though the 'Saxofour' interpretation is instrumental, the lyrics space helps to uncover the similarity between the two songs. Songs by 'Bright Eyes' are concentrated around clusters of rather slow folk music.

Noticeable Artists

Table 6.2.2 shows a selection of particularly interesting artists with respect to their positions on the maps. A total of 18 'Sean Paul' songs are mapped on each *Self-Organising Map*. For the audio map, the songs are distributed across seven different units, eleven being mapped onto one unit. On the lyrics map, all songs are mapped

Artist	AC	LC	CR	ADR	ADR×CR
Sean Paul	.3162	.1313	.4152	.4917	.2042
Good Riddance	.0403	.0485	.8299	.7448	.6181
Silverstein	.0775	.1040	.7454	.8619	.6424
Shakespeare	.2626	1.000	.2626	.3029	.0795
Kid Rock	.0894	.0862	.9640	.9761	.9410

Table 6.5: Artists with exceptionally high or low spreading values. **AC** denotes the audio contiguity, **LC** the lyrics Contiguity, **CR** the contiguity ratio, **ADR** the average distance ratio, and **ADR×CR** the product of **ADR** and **CR**

onto two adjacent units, the first one covering 17 out of the 18 tracks. The univying theme for the distribution across units is based on song labels in the textual feature space, i.e. songs having similar labels will be mapped onto units having high weights for these labels.

The situation is different for 'Good Riddance', a Californian 'Punk Rock' band. For the lyrics map, their 27 songs are spread across 20 units. For audio, the songs lie on 18 units, but some of them are adjacent units, a fact that is represented by a rather high value for AC, the audio contiguity measure.

Shakespeare sonnets are clustered in a similar way. In terms of lyrics the six sonnets lie on two units, whereas the audio representations are mapped on three units, non of which were adjacent (only one sonnet is read by a male voice).

'Kid Rock' songs, mainly 'Country' tracks, lie on 13 units on the audio map, including two adjacent units, compared to 11 units in the lyrics space, none of which are adjacent. The spread is therefore almost identical on both maps. Figure 6.9 shows the 3D visualisation for all 'Kid Rock' songs.

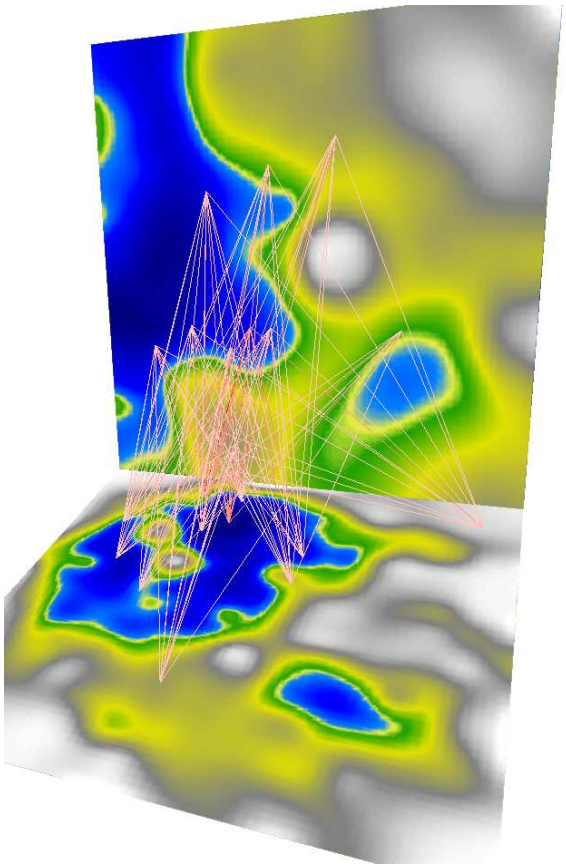


Figure 6.9: Detailed view of connections for the almost equally distributed artist 'Kid Rock'. Dark lines denote a high number of connections

Genre	AC	LC	CR	ADR	ADR×CR
Speech	.0822	.0665	.8092	.3417	.2765
Christmas Carols	.0393	.0677	.5800	.7779	.4512
Reggae	.0392	.0413	.9495	.8475	.8047
Grunge	.0382	.0466	.8204	.9974	.8182
Rock	.0372	.0382	.9740	.9300	.9059

Table 6.6: Genres with exceptionally high or low spreading values. **AC** denotes the audio contiguity, **LC** the lyrics contiguity, **CR** the contiguity ratio, **ADR** the average distance ratio, and **ADR×CR** the product of **ADR** and **CR**

Noticeable Genres

Analogously to the artists, we identified genres of interest in Table 6.2.2.

'Rock' music has proven to be the most diverse genre in terms of audio features and rather diverse in terms of lyrics features alike. There were 672 songs assigned to that genre in the test collection. The overall $adr \times cr$ measure is still rather high due to the impact of adjacent units on both maps. 'Speech' as well as 'Christmas Carols', on the other hand, are rather diverse in terms of audio similarity, but are more concentrated on the lyrics (or text) level, yielding in a low $adr \times cr$ value. Figure 6.10 shows the connections between all 'Christmas' songs, giving an interesting idea about the differences in distributions on the maps, c.f. Figure 6.8. The similarity of 'Reggae' music is defined by acoustic and lyrics features to an equal amount. This genre has rather high values for adr and cr, caused by a high number of adjacent units and a low overall number of units.

6.2.3 Musical Genre Classification

In order to utilise the information contained in music for genre classification, we describe sets of audio features derived from the waveform of audio tracks as well as the

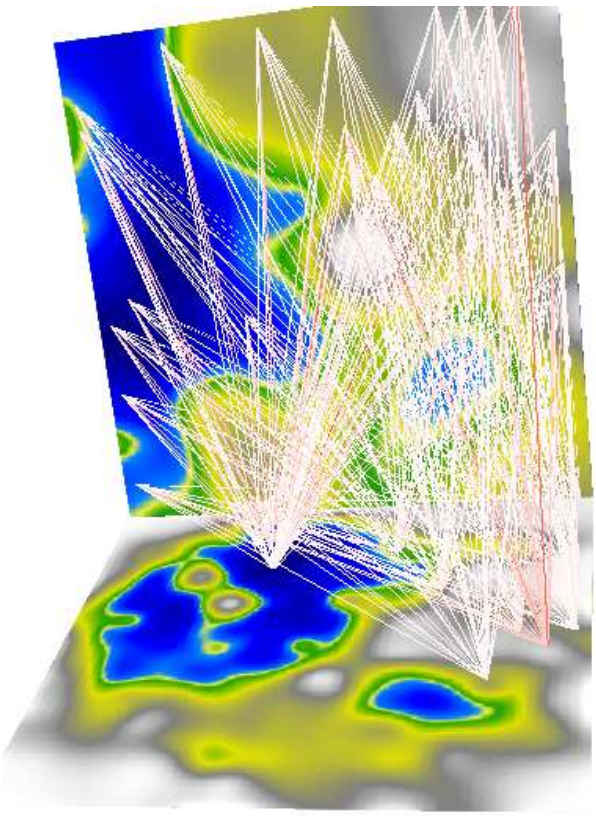


Figure 6.10: Detailed view of connections for the genre 'Christmas Carols'. Dark links denote a high number of connections

Table 6.7: Macro-avraged classification accuracies based on ten-fold cross validation for different types and combinations of audio features and features based on lyrics. The experiments A1 - A3 denote audio-only, L1 - L4 lyrics-only, and C1 - C3 features combined from audio and lyrics feature sets. The type column shows the types of feature sets used, dimensionality notes the resultant dimensionality of the data

Name	**Type**	**Dimensionality**	**Classification Accuracy**
A1	RH.	60	.264702
A2	SSD.	168	.377473
A3	RP.	1440	.375454
L1	LYRICS	60	.216076
L2	LYRICS	168	.263394
L3	LYRICS	1422	.334101
L4	LYRICS	3000	.363122
C1	LYRICS + RH	120	.375454
C2	LYRICS + SSD	336	.436819
C3	LYRICS + RP	3085	.429821

bag-of-word features for song lyrics. Our experiments were performed on the large test collection introduced in Chapter 3.

Table 6.7 shows classification accuracies for a set of experiments based on audio and lyrics features as well as combinations thereof. We achieved the different lyrics dimensionalities by document frequency thresholding, the upper limit was set to 40 per cent, the lower threshold was continually increased as to match the required resultant dimensionality, leading to different values for the lower threshold in all settings. Experiments were performed by Weka's implementation of Support Vector Machines for ten-fold stratified cross validation (linear kernel, $c = 1.0$). Results shown are the macro averaged classification accuracies.

The classifiers based on audio data showed good results, experiment A2 and A3 being rather similar, even though the dimensionalities are quite different. Experiment

A1 showed by far lower results.

The higher-dimensional the data for the lyrics experiments is, the higher is its classification accuracy, implying that there is even more discriminating information contained in lyrics (see experiments L1 - L4), which is not covered in this context because of the limitations of the simple concatenation approach.

For combination experiments (C1 - C3) we use balanced combinations of features, i.e. the dimensionality of the lyrics component always equals the dimensionality of the audio feature component[3]. Results show that a combination of lyrics and audio features improves overall classification performance. Very high accuracy was achieved in the 'LYRICS + RP' setting (C3), having the highest dimensionality, second only to the 'LYRICS + SSD' experiment (C2). For all combination experiments (C1 - C3) the accuracies were at least equal to the highest values for the respective one-dimensional approaches (A3 and L4).

For statistical significance testing we used a paired T-test for a significance level of $\alpha = .05$. Results showed that A2 performs better than A1 ($p = .0189$), but there is no significant difference between A2 and A3 ($p = .9661$). Further, it is shown that C3 performed better than L3 ($p = .0059$). Hence, a classifier based on differing numbers of lyrics than audio features, e.g. more dimensions in the lyrics than in the audio space, might further improve classification accuracy. Yet, by combining lyrics and audio (C2), the same performance was achieved at a fraction of the dimensionality. Experimental results for C2 and C3 are not significantly different ($p = .7994$). Further test results are given in Table 6.8.

[3]These values sometimes are slightly skewed due to the dynamic feature space reduction with document frequency thresholding.

Table 6.8: p-values obtained by statistical significance tests performed on classification results. The given tests were performed for a significance level of $\alpha = .05$ using a paired T-Test for distributions with equal means

Name	**C2**	**C3**	**A2**	**L3**	**L4**
A1		.0157(1)	.0189(1)	.0732(0)	.2021(0)
A2	.0074(1)	.2298(0)		.9661(0)	.8118(0)
A3	.0885(0)	.0059(1)	.9661(0)	.3208(0)	.5197(0)
L1	1.0096e-04(1)	1.0597e-04(1)	.0051(1)	2.2785e-04(1)	.2021(0)
L2	.0011(1)	2.0158e-05(1)	.0573(0)	1.0526e-05(1)	2.3352e-04
L3	.0885(0)	.0059(1)	.9661(0)		.5197(0)
L4	.1343(0)	.0076(1)	.8118(0)	.5197(0)	
C1	1.2867e-04(1)	.0031(1)	.0031(1)	.0435(1)	.2173(0)
C2		.7994(0)	.0074(1)	.0885(0)	.1343(0)
C3	.7994(0)		.2298(0)	0.0059(1)	.0076(1)

6.3 Recap

In this chapter we provided experimental results on two test collections – one of small, one of large size. We thereby underpinned our position that Music Information Retrieval greatly benefits from the use of multi-modal data sources. We provided results for multi-modal clustering, relying on the lyrics space as additional input information. These principles were evaluated both in terms of an experimental user interface and quantitative evaluation. We used a wide range of textual data sources like artist descriptions or album reviews, to provide experimental results for the classic similarity retrieval use case. The combination of these data sources extended the classic approach of using audio similarity only. We furthermore showed that lyrics can greatly influence the task of musical genre classification and provided statistical significance tests for our classification experiments.

Chapter 7

Conclusions and Future Work

> To a god, a wall is but a line on a page. We are all naked, seen beyond seeing.
> Wayfinder Hasturi, aka "The Mad Perseid" 217 AFC

In this thesis, we investigated a multi-modal vision of Music Information Retrieval, taking into account both a song's lyrics as well as its acoustic representation, as opposed to concentrating on acoustic features only. We presented a novel approach to the visualisation of multi-modal clusterings and showed its feasibility to introspect collections of digital audio, in form of a prototype implementation for handling private music collections, emphasised by concrete examples. On top of that, we introduced performance metrics for *Self-Organising Maps* on a per-class level (e.g. artist or genre classes), showing differences in spreading across maps. Moreover, we introduced measurements for the comparison of multi-modal clusterings that showed their application to identify genres or artists of particular interest.

We also integrated textual data beyond lyrics. A similarity ranking technique was presented to additionally accommodate for further data sources such as artist and genre descriptions and album reviews. To show the applicability of this approach we presented a prototype that allows for interactive adjustments in weightings for these different modalities.

As another application we performed musical genre classification on audio tracks represented by their indexed lyrics as well as audio features. We presented experimental results showing that a feature combination is highly desirable in order to increase classification accuracies.

Future work will mainly deal with the further exploitation of multi-faceted representations of digital audio. Further, we plan to provide a more elaborate user interface that offers sophisticated search capabilities. Ensemble methods have been successfully used for the integration of multiple classifier instances and might prove particularly useful for the music scenario. These classifiers mostly differ in the subset of features of classifier technique used. In this context, classifiers could be trained on different sets of features – motivated by the wealth of modalities available for musical data. Such an approach would be feasible to achieve better overall integration and accuracy rates for the musical genre classification task.

Besides, the possibilities of automatically adding metadata to audio files through multi-modal representations will be explored in connection with semantic analysis or automatic concept identification in music. An interesting application of this would be automatic musical genre classification, emphasising on the additional information contained in a song's lyrics as opposed to purely acoustic approaches currently being in use. Moreover, the investigation and evaluation of advanced feature sets for the lyrics space will play an important role in future work.

In this thesis, a suitable categorisation of textual data was presented, which can practicably be exploited for similarity retrieval. Our experimental results showed how important the different weightings are and in how far they influence the results. Nonetheless, our evaluation approach can only be seen as a first step towards a more encompassing utilisation of multiple dimensions in Music Information Retrieval. Moreover, strategies for dealing with information that is not present in such a system that showed improvements compared to the simple exclusion strategy, were presented. However, the results lead to the conclusion that a higher coverage of text data is desirable

to improve similarity retrieval results.

One future goal is to find an optimal weighting for the different levels presented in this thesis – both according to the evaluation used and users' preferences. This approach obviously offers itself for the application of a relevance feedback approach, emphasising the interactive dynamics required to be addressed when talking about music similarity. A long term objective is the integration of more sophisticated retrieval components, yielding a possibly much higher coverage. Moreover, for being vital aspects for every large-scale Music Information Retrieval system, scalability and performance issues need serious attention.

Exploiting the results from the comparisons of clusterings for classification, particularly its feasibility for ensembles of classifiers, could improve results.

List of Tables

2.1 Text indexing by example. Tokens are displayed horizontally, different documents are shown row-wise. The token's occurrences make out the numbers in the table 17

2.2 Variable names used in cluster validation equations 25

3.1 Composition of the small test collection 37

3.2 Overview of genres in the music collection used throughout this thesis . 38

4.1 Test collection and coverage of different types of descriptions for the collection used in the experimental evaluation 46

4.2 Calculation of average distance values for clusterings e in Figure 4.2 . . 57

4.3 Calculation of average distance values for clusterings f in Figure 4.2 . . 57

4.4 Scatter measures for *Self-Organising Maps* (see Figure 4.2). Note, (a) denotes the audio clusterings a, c, e and g; (l) the lyrics clusterings b, d, f and h. **AC** and **LC** denote the contiguity ratios for audio and lyrics, respectively 58

6.1 Genres and the according spreading values across clusterings. **AC** denotes the audio contiguity, **LC** the lyrics contiguity, **CR** the contiguity ratio, **ADR** the average distance ratio, and **ADR**×**CR** the product of **ADR** and **CR**. Maximum values are printed in bold font, minimum values italic 83

6.2 Results for given weighting strategies. The different weightings are given in the Audio, Artist, Lyrics, Album and Genre columns. The Sum column denotes the sum over the number of songs amongst the top 20 results from the same artist plus album plus genre for each combination 85

6.3 Experimental results for similarity ranking experiments using different substitution strategies for the combination of the results taken from a full ranking of all songs. Numbers given denote the number of songs belonging to the same artist, album, and genre as the seed song in the top 5, 10, or 20 songs retrieved 87

6.4 Re-Ranking of top 600 initial results for similarity ranking experiments using different substitution strategies weighting for the combination of the results. Numbers given denote the number of songs belonging to the same artist, album, and genre as the seed song in the top 5, 10, or 20 songs retrieved 88

6.5 Artists with exceptionally high or low spreading values. **AC** denotes the audio contiguity, **LC** the lyrics Contiguity, **CR** the contiguity ratio, **ADR** the average distance ratio, and **ADR**×**CR** the product of **ADR** and **CR** 95

6.6 Genres with exceptionally high or low spreading values. **AC** denotes the audio contiguity, **LC** the lyrics contiguity, **CR** the contiguity ratio, **ADR** the average distance ratio, and **ADR**×**CR** the product of **ADR** and **CR** 97

6.7 Macro-avraged classification accuracies based on ten-fold cross validation for different types and combinations of audio features and features based on lyrics. The experiments A1 - A3 denote audio-only, L1 - L4 lyrics-only, and C1 - C3 features combined from audio and lyrics feature sets. The type column shows the types of feature sets used, dimensionality notes the resultant dimensionality of the data 99

6.8 p-values obtained by statistical significance tests performed on classification results. The given tests were performed for a significance level of $\alpha = .05$ using a paired T-Test for distributions with equal means 101

List of Figures

3.1 Categorisation of multi-level Music Information Retrieval 35

3.2 Lyrics retrieval, the *Atlantis* way 39

4.1 Visualisation prototype mock-up 51

4.2 Distribution of four data points belonging to one class (this could be, e.g., four pieces of 'Rock' music). The figures in the left column display possible distributions of data points according to the audio dimension, whereas the right column represents possible arrangements for the lyrics scenario. All figures are examples only and do not rely on real-world data 56

5.1 Overview of *Atlantis*' packages 61

5.2 Classes for the management of corpora within the framework 63

5.3 Classes for the representation of various documents 64

5.4 Lyrics fetching and parsing - the *Atlantis* way 65

5.5 Classes for vector generation and dimensionality reduction of text corpora 66

5.6 Overview of distance measures used in *Atlantis* 69

5.7 Overview of the ranking implementations 70

5.8 Full view of the visualisation prototype. The vertical map clusters songs by audio features, the horizontal map is trained on lyrics features. The left hand side is occupied with various selection controls 72

5.9 Overview of the *Sovis* implementation 74

5.10 *Sovis*' GUI components 75

5.11 An overview of *Sovis*' quality measures 76

6.1 Thematic class map visualisation for the audio clustering of the 10 genres subset of the small audio collection. Genre colours are displayed in the legend 78

6.2 Thematic class map visualisation for the lyrics clustering of the 10 genres subset of the small audio collection. Genre colours are displayed in the legend 79

6.3 Overview of the visualisation prototype. The left part of the application is occupied by tools to select songs from the audio collection. The main part displays the clusterings and connections in between 81

6.4 Distribution of selected genres across maps 82

6.5 GUI for experimental evaluation of different weighting strategies. Weightings are given for the track 'Politik' by 'Coldplay' 89

6.6 Combined ranking for the track 'Politik' by 'Coldplay', based on single rankings in five modalities 91

6.7 Clustering of songs centred around the love topic 92

6.8 Distribution of Christmas carols on clusterings for different feature spaces. The pie charts denote the distribution of songs over different genres on the particular units – only units comprising Christmas carols are highlighted 93

6.9 Detailed view of connections for the almost equally distributed artist 'Kid Rock'. Dark lines denote a high number of connections 96

6.10 Detailed view of connections for the genre 'Christmas Carols'. Dark links denote a high number of connections 98

Bibliography

[1] Charu C. Aggarwal and Philip S. Yu. Outlier detection for high dimensional data. *SIGMOD Rec.*, 30(2):37–46, 2001.

[2] Hans-Ulrich Bauer and Klaus R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *Transactions on Neural Networks*, 3(4):460–465, 1992.

[3] Doris Baum. Visualisations for comparing self-organising maps. Master's thesis, Vienna University of Technology, Department of Software Technology and Interactive Systems, March 2007.

[4] Stephan Baumann, Tim Pohle, and Shankar Vembu. Towards a socio-cultural compatibility of mir systems. In *Proceedings of the 5th International Conference of Music Information Retrieval (ISMIR'04)*, pages 460–465, Barcelona, Spain, October 10-14 2004.

[5] Christopher J. C. Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[6] Sally Jo Cunningham, Nina Reeves, and Matthew Britland. An ethnographic study of music information seeking: implications for the design of a music digital library. In *Proceedings of the 3rd ACM IEEE Joint Conference on Digital Libraries (JCDL'03)*, pages 5–16, Houston, Texas, US, 2003. IEEE Computer Society.

[7] Michael Dittenbach, Robert Neumayer, and Andreas Rauber. PlaySOM: An alternative approach to track selection and playlist generation in large music col-

lections. In *Proceedings of the 1st International Workshop of the EU Network of Excellence DELOS on Audio-Visual Content and Information Visualization in Digital Libraries (AVIVDiLib 2005)*, pages 226–235, Cortona, Italy, May 4-6 2005.

[8] J. Stephen Downie. *Annual Review of Information Science and Technology*, volume 37, chapter Music Information Retrieval, pages 295–340. Information Today, Medford, NJ, 2003.

[9] Jonathan Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):2–10, 1999.

[10] Masataka Goto and Takayuki Goto. Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 404–411, London, UK, September 11-15 2005.

[11] Masatoshi Hamanaka and Seunghee Lee. Music scope headphones: Natural user interface for selection of music. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, pages 302–307, Victoria, BC, Canada, October 8-12 2006.

[12] Peter Knees, Markus Schedl, Tim Pohle, and Gerhard Widmer. An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web. In *Proceedings of the ACM Multimedia 2006 (ACMMM'06)*, pages 17–24, Santa Barbara, California, USA, October 23-26 2006.

[13] Peter Knees, Markus Schedl, and Gerhard Widmer. Multiple lyrics alignment: Automatic retrieval of song lyrics. In *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 564–569, London, UK, September 11-15 2005.

[14] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[15] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, 2001.

[16] Teuvo Kohonen, Erkki Oja, Olli Simula, Ari Visa, and Jari Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–1384, October 1996.

[17] Petri Kontkanen, Jussi Lahtinen, Petri Myllymäki, Tomi Silander, and Henry Tirri. Supervised model-based visualization of high-dimensional data. *Intelligent Data Analysis*, 4:213–227, 2000.

[18] Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 34–41, London, UK, September 11-15 2005.

[19] Beth Logan, Andrew Kositsky, and Pedro Moreno. Semantic analysis of song lyrics. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME'04)*, pages 827–830, Taipei, Taiwan, June 27-30 2004. IEEE Computer Society.

[20] Beth Logan and Ariel Salomon. A music similarity function based on signal analysis. In *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo (ICME'01)*, pages 745–748, Tokyo, Japan, August 22-25 2001. IEEE Computer Society.

[21] Last.fm Ltd. last.fm, the social music revolution. Website. `http://www.last.fm`.

[22] Dominik Lübbers. SONIXPLORER: Combining visualization and auralization for content-based exploration of music collections. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 590–593, London, UK, September 11-15 2005.

[23] Jose P. G. Mahedero, Álvaro Martínez, Pedro Cano, Markus Koppenberger, and Fabien Gouyon. Natural language processing of lyrics. In *Proceedings of the 13th annual ACM international conference on Multimedia (ACMMM'05)*, pages 475–478, New York, NY, USA, 2005. ACM Press.

[24] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[25] Rudolf Mayer, Taha Abdel Aziz, and Andreas Rauber. Visualising class distribution on self-organising maps. In Joaquim Marques de Sá, Luís A. Alexandre, Włodzisław Duch, and Danilo Mandic, editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN'07)*, volume 4669 of *LNCS*, pages 359–368, Porto, Portugal, September 9 - 13 2007. Springer.

[26] Rudolf Mayer, Thomas Lidy, and Andreas Rauber. The map of Mozart. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, pages 351–352, Victoria, BC, Canada, October 8-12 2006.

[27] Rudolf Mayer, Dieter Merkl, and Andreas Rauber. Mnemonic SOMs: Recognizable shapes for self-organizing maps. In Marie Cottrell, editor, *Proceedings of the Fifth International Workshop on Self-Organizing Maps (WSOM'05)*, pages 131–138, Paris, France, September 5-8 2005.

[28] Annual Music Information Retrieval Evaluation eXchange (MIREX). Website, 2005. `http://www.music-ir.org/mirexwiki/index.php/Main_Page`.

[29] Fabian Mörchen, Alfred Ultsch, Mario Nöcker, and Christian Stamm. Databionic visualization of music collections according to perceptual distance. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 396–403, London, UK, September 11-15 2005.

[30] Robert Neumayer. Clustering based ensemble classification for spam filtering. In *Proceedings of the 7th Workshop on Data Analysis (WDA'06)*, pages 11–22. Elfa Academic Press, June 29 - July 2 2006.

[31] Robert Neumayer, Christoph Becker, Thomas Lidy, Andreas Rauber, Eleonora Nicchiarelli, Manfred Thaller, Michael Day, Hans Hofman, and Seamus Ross. Development of an open testbed digital object corpus. Technical report, March 2007.

[32] Robert Neumayer, Michael Dittenbach, and Andreas Rauber. PlaySOM and PocketSOMPlayer: Alternative interfaces to large music collections. In *Proceedings*

of the 6th International Conference on Music Information Retrieval (ISMIR'05), pages 618–623, London, UK, September 11-15 2005. Queen Mary, University of London.

[33] Robert Neumayer, Jakob Frank, Peter Hlavac, Thomas Lidy, and Andreas Rauber. Bringing mobile based map access to digital audio to the end user. In *Proceedings of the 14th International Conference on Image Analysis and Processing Woorkshops (ICIAP'07), 1st Workshop on Video and Multimedia Digital Libraries (VMDL'07)*, pages 9–14, Modena, Italy, September 10-13 2007. IEEE.

[34] Robert Neumayer and Andreas Rauber. Integration of text and audio features for genre classification in music information retrieval. In *Proceedings of the 29th European Conference on Information Retrieval (ECIR'07)*, pages 724–727, Rome, Italy, April 2-5 2007.

[35] Robert Neumayer and Andreas Rauber. Multi-modal music information retrieval - visualisation and evaluation of clusterings by both audio and lyrics. In *Proceedings of the 8th Conference Recherche d'Information Assistée par Ordinateur (RIAO'07)*, Pittsburgh, PA, USA, May 29th - June 1 2007. ACM.

[36] Nicola Orio. Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieval*, 1(1):1–90, September 2006.

[37] François Pachet and Daniel Cazaly. A taxonomy of musical genres. In *Proceedings of Content-Based Multimedia Information Access Conference (RIAO'00)*, pages 827–830, Paris, France, April 12-14 2000.

[38] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Content-based Organization and Visualization of Music Archives. In *Proceedings of the ACM Multimedia (MM'02)*, pages 570–579, Juan les Pins, France, December 1-6 2002. ACM.

[39] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In *Proceedings of 12th the International Conference on Artifical Neural Networks (ICANN'02)*, pages 871–876, Madrid, Spain, August 27-30 2002. Springer.

[40] Jerome Penaranda. Text Mining von Songtexten. Master's thesis, Vienna University of Technology, Department of Software Technology and Interactive Systems, March 2007. In German.

[41] Amarok Project. Amarok music player. `http://amarok.kde.org`.

[42] Andreas Rauber and Michael Frühwirth. Automatically analyzing and organizing music archives. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'01)*, LNCS, pages 402–414, Darmstadt, Germany, September 4-8 2001. Springer.

[43] Andreas Rauber and Dieter Merkl. The SOMLib digital library system. In *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL'99)*, Lecture Notes in Computer Science (LNCS 1696), pages 323–342, Paris, France, September 22-24 1999. Springer.

[44] Andreas Rauber and Dieter Merkl. Text mining in the SOMLib digital library system: The representation of topics and genres. *Applied Intelligence*, 18(3):271–293, May-June 2003.

[45] Andreas Rauber, Elias Pampalk, and Dieter Merkl. Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR'02)*, pages 71–80, Paris, France, October 13-17 2002.

[46] Andreas Rauber, Elias Pampalk, and Dieter Merkl. The SOM-enhanced JukeBox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research*, 32(2):193–210, 2003.

[47] Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

[48] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[49] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[50] Markus Schedl, Peter Knees, and Gerhard Widmer. Discovering and visualizing prototypical artists by web-based co-occurrence analysis. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 21–28, London, UK, September 11-15 2005.

[51] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[52] George Tzanetakis and Perry Cook. Marsyas: A framework for audio analysis. *Organized Sound*, 4(30):169–175, 2000.

[53] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[54] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer, New York, NY, USA, 1995.

[55] Fabio Vignoli and Steffen Pauws. A music retrieval system based on user-driven similarity and its evaluation. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 272–279, London, UK, September 11-15 2005.

[56] Fabio Vignoli, Rob van Gulik, and Huub van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 409–414, Barcelona, Spain, October 10-14 2004.

[57] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, CA, USA, 2nd edition, 2005.

[58] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of the 14th Inter-*

national Conference on Machine Learning (ICML'97), pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[59] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics, Facts and Models*, volume 22 of *Series of Information Sciences*. Springer, Berlin, 2 edition, 1999.