Exploring the Music Genome: Lyric Clustering with Heterogeneous Features

Tamsin Maxwell

Master of Science

Cognitive Science and Natural Language Processing

School of Informatics

University of Edinburgh

2007

Abstract

This research explores the clustering of songs using lyrics features grouped into similar classes and heterogeneous combinations. Simple techniques are used to extract 140 features for analysis with Kohonen self-organising maps. These maps are evaluated using visual analysis and objective measures of validity with respect to the clustering of eight hand-selected song pairs. According to gold standard human-authored playlists, judgments of song similarity are based strongly on music, however this observation may be limited to playlists and is not necessarily extensible to music in the wider domain. In particular, since test song pairs could only be effectively matched when they were from the same genre, analysis of the correspondence between lyrics and expert human judgments of genre and style may be more fruitful than comparison with similarities observed in playlists.

Results suggest that for music in the hard-to-differentiate categories of pop, rock and related genres, a combination of features relating to language, grammar, sentiment and repetition improve on the clustering performance of Information Space with a more accurate analysis of song similarity and increased sensitivity to the nuances of song style. SOM analysis further suggests that a few well-chosen attributes may be as good as, if not better than, deep analysis using many features. Results using stress patterns are inconclusive. Although results are preliminary and need to be validated with further research on a larger data set, to the knowledge of this author this is the first time success has been reported in differentiating songs in the rock/pop genre.

Acknowledgements

I would like to thank my supervisor, Jon Oberlander, and Ben Hachey for their enthusiasm, encouragement and insight; our discussions were as enjoyable as they were productive. I am also grateful to Theresa Wilson for precise and cheerful direction regarding sentiment analysis, and provision of the sentiment lexicon constructed for her work. Finally, I remain indebted to Gregor for emotional support and structured commentary at crucial junctures, and the Stanford-Edinburgh Link that supported me during this year.

Several software packages and other resources were used in the course of this research and I would like to thank the people who made these available. Adam Berenzweig and Brian Whitman from Columbia University who collected the Art of the Mix user-authored playlists and regularised the mapping from artists to numeric codes. Esa Alhoniemi, Johan Himberg, Juha Parhankangas and Juha Vesanto and all those who contributed to the SOM Toolbox at the Laboratory of Information and Computer Science in the Helsinki University of Technology. Dimitrios Zeimpekis and Efstratios Gallopoulos at the University of Patras who created the Text to Matrix Generator (TMG). I am also grateful to Claire Grover and Ewan Klein from the University of Edinburgh for use of the `txt2onto` software used in tokenisation, chunking, lemmatisation and verb analysis, as well as Richard Tobin, also at Edinburgh, whose `LT-XML2` software supports the `txt2onto` modules. Finally, I would like to acknowledge all those at CELEX, the Dutch Centre for Lexical Information, which created the lexicon of stress patterns, and those at the SRI Speech Technology and Research Laboratory where the SRILM toolkit used for ngram counts remains under construction.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Tamsin Maxwell*)

Table of Contents

Chapter 1

Introduction

The rapid growth of digital music has opened the door to computer-based methods of music analysis, description, cataloguing, search and recommendation, and spurred the development of Music Information Retrieval (MIR) as a specialist field. Since its inception in the late 1990s, MIR has grown to encompass its own TREC-style contest in 2005 (Pienimaki, 2006), and continues to make music more accessible to information systems and commercial services, such as automated music recommendation or playlist generation services and Internet radio.

The vast majority of MIR research focuses on automatic means of extracting and applying audio features, or score notation, to classify or compute similarity between audio sequences. This is exemplified by tasks at the first two MIREX (Music Information Retrieval Evaluation eXchange) evaluation contests (Pienimaki, 2006). In addition, cultural meta-data extracted from human-authored text on the worldwide web is sometimes used as the basis of retrieval (Baumann and Hummel, 2005).

Perhaps surprisingly, lyrics have been largely neglected as a source of information about music similarity. In a recent survey of MIR systems, only one included lyric features and this ensured song matching (lyric alignment). It did not compare the affect, tone, rhythm or content of lyrics for different songs, artists or genres (Whitman, 2005). Despite this, in all likelihood lyrics are a rich source of information; they are set to a repetitive beat, are frequently emotive, and have a tendency towards non-standard use of language, thus they may be expected to express strong variety or saliency in syntactic, semantic and acoustic properties that could help separate and cluster artists or songs and help bridge the gap currently observed in this area of music retrieval (Figure 1.1)

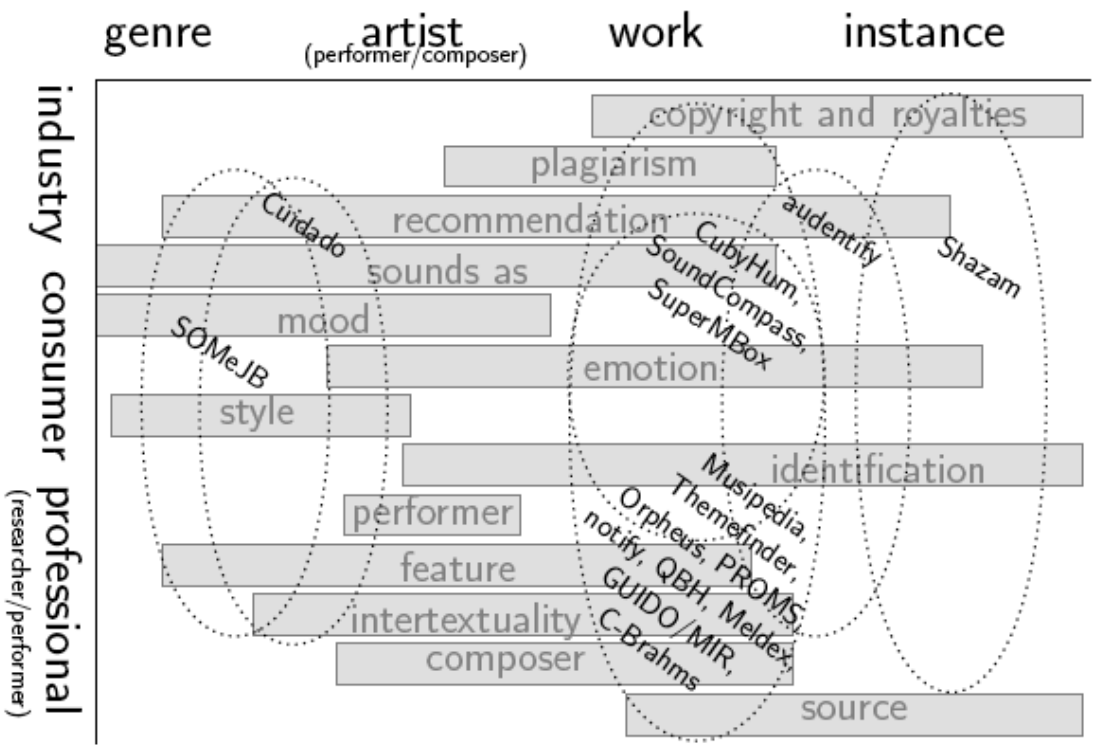


Figure 1.1: *Mapping of MIR system coverage for retrieval tasks. Figure reproduced from Typke et al. (2005).*

The paucity of research into song lyrics raises the question of what contribution, if any, lyrics might make to music classification and retrieval. It seems reasonable to assume that the level of similarity between two songs might be determined by musical attributes alone, but freeform descriptions of music point towards lyrics having an important secondary role. Take for example, reviews of Amy Winehouse's recent album *Back to Black*, which contains a wealth of musical references, a rich sound and a simple lyrical style. Table 1.1 shows that even when the album has strong musical influences, lyrics are a salient discussion point in reviews. Song topic, genre, lyrical style and sentiment, as well as audio features, vocal style and reference to other landmark artists all feature as important elements in music description.

Research on lyrics to date has addressed low-level semantic features like part of speech (POS), verb class, voice and use of function words (Li and Ogihara, 2004). A study by Logan et al. (2004) has also considered latent semantic analysis (LSA), and whilst results were not as accurate as those achieved with acoustic similarity, the authors compared the classification errors made using lyrics and music-level information and concluded that semantic and musical dimensions have different strengths and may prove to be complementary techniques. Such results are intriguing, yet the studies do

Vocal	Genre	Music audio	Sentiment	Topic	Lyric style	Landmark	Review	Source
x				x			'Back To Black' is the second album from London-based chantuese Amy Winehouse. Combining a strong, jazzy vocal style with often frank lyrical content recounting tales of love and loss	Amazon
	x		x	x			Underlying the Winehouse sound is a solid 60's soul vibe which when drizzled with a soupcon of jazz, and R & B produces an album of passion and energy...Narrative tracks drawn from life are injected with real soul and at times anger	Amazon
x	x						Amy tightens up her songwriting somewhat on 'Back To Black', swaps jazz for a more muscular 60s style R&B sound and flexes her vocal delivery to reveal a much deeper and harder style of singing	Amazon
		x				x	This time around, she's taken her inspiration from some of the classic 1960's girl groups like the Supremes and the Shangri-Las...the title track (and album highlight) is a heartbreaking musical tribute to Phil Spector, with it's echoey bass drum, rhythmic piano, chimes, saxophone and close harmonies.	Amazon - Ted Kord
			x	x			Much of the rest of this pleasingly short album - eleven three-minute or so tracks – goes on to explore the joyful misery of being young, messy and in love/lust.	BBC - Matt Harvey
x			x				Fortunately, Winehouse has been blessed by a brassy voice that can transform even mundane sentiments into powerful statements	Rolling Stone
x	x	x					[It] sounds fantastic--partly because the production nails sample-ready '60s soul right down to the drum sound' and partly because Winehouse is one hell of an impressive singer	Blender
	x	x			x		It works - even though this area of pop culture has been mined remorselessly for the past 50 years - by dint of its clever melody lines and smart lyrics	Observer Music Monthly
					x		It's precisely Winehouse's lyrics... that raise this expertly crafted set into the realm of true, of-the-minute originality.	Entertainment Weekly

Table 1.1: *Extracts of reviews for Amy Winehouse's album, Back to Black.*

not come close to exhausting the range of information that might be extracted from language.

Lyrics potentially bear much information that might be used to discriminate between song styles, but it is not clear whether one or several discrete classes of information are better at this task than others, if a few well-chosen attributes could serve as proxies for more detailed and time-consuming analysis, or if extracted information enhances discrimination only when taken as a comprehensive snapshot of all that language has to offer. Further, evidence from cognitive neuroscience shows that audio signals and language are processed in parallel through separate areas of the brain (Besson et al., 1998), so for music with verbal content, any analysis that uses only the syntax and semantics of lyrics may be missing a pertinent dimension of information: the acoustics of language itself.

This research explores new territory in lyric analysis by investigating how much clustering of songs can be achieved with a simple approach using heterogeneous categories of lyric features alone and in combination. It is differentiated from previous research in several ways. First, many of the features extracted from lyrics are unique, where to the knowledge of this author lyric sentiment, repetition, combined measures of language complexity and formality, and stress patterns are addressed in this context for the first time. The study of stress patterns in the lyrical context is particularly novel as it bridges the gap between language structure and acoustic properties like beat and rhythm. Second, these features are investigated alone and in combination to ascertain whether the simultaneous use of multiple features benefits model performance. Earlier work on textual music data was limited to linguistic features, grammar, content and lyric form (e.g. Li and Ogihara (2004); Logan et al. (2004); Dhanaraj and Logan (2005)). Finally, the coverage and nature of the data is unique in that it focuses almost solely on the pop/rock genre.

Music classification using textual data has succeeded in differentiating very different genres such as classical, rap and rock (Whitman and Smaragdis, 2002) but all previous research failed, or did not attempt, to clearly differentiate the similar styles of pop, rock, indie, metal and punk (Knees et al., 2004, 2005a). The source data set and method of data selection used in this research have captured data encompassing precisely these genres, presenting a challenging task for discrimination.

In addition, the focus here is on song-level similarity, whereas previous research attempted to cluster or classify music on the basis of recording artist, genre or style. There is difficulty in assessing similarity resulting from artist, genre or style compar-

isons, most notably because judgments of genre are known to be difficult, even for humans, and because such comparisons assume bi-directional similarity. In contrast, a survey of six human-generated music similarity networks showed them all to be directed (Cano et al., 2006) so that, for example, Oasis is similar to The Beatles, but The Beatles are not similar to Oasis. There is also the challenge that songs may vary considerably within a single album and dramatically between albums on which an artist works with different collaborators and producers. Such variations have the potential to thwart sensitive language measurements. For this reason, although it made acquisition of gold standard data more time-consuming, song-level similarity was preferred.

The hypothesis of this research is that for lyrical content of music, combined features relating to language, grammar, sentiment, repetition and stress patterns will provide better clustering performance with respect to human perceptions of song similarity than LSA alone. In this sense, the research builds on earlier work by Logan et al. (2004) which is the largest analysis of song lyrics to date, applying LSA to a collection of 15,589 songs from 399 artists and comparing the results to those achieved with audio features. As will be shown, although similarity between songs in the gold standard human playlists appears to be based purely on music, comparison with expert tags for genre and style information reveals interesting results.

This research does not attempt to provide definitive answers, or conclusively show the superiority of one set of features over another. Due to the number and variety of features investigated and the need for each of these to be visually assessed, the search and selection is necessarily heuristic. The aim of this work is to provide insight into how different combinations of features might influence overall clustering results on a larger lyric data set, and useful directions for further research.

The remainder of this thesis will be organised as follows. Chapter 2 will provide background on literature relating to music classification with a strong focus on research using music-related textual data and techniques used in data mining. Kohonen self-organising maps and LSA are introduced and an explanation given for important considerations regarding their use. In Chapter 3, the data obtained for this study is discussed and brief details provided on data collection and cleaning procedures including ebonics correction. The extraction of features used to represent lyrical content is outlined in Chapter 4, including limitations of extraction techniques, and the methodology for model search and analysis is explained in Chapter 5. Chapter 6 presents results, followed by a discussion of findings and conclusion in Chapters 7 and 8.

Chapter 2

Background

2.1 Related work

Efficient and accurate automated determination of music similarity is important for successful MIR, and equally important for automated playlist generation (Knees et al., 2006b), music recommendation systems (Vembu and Baumann, 2004) and content-based querying or browsing of music collections, whether these are large databases or consumer hand-held devices (Neumayer et al., 2005). Music classification is the backbone of meta-tagging new music being added to databases, and may even support automated generation of word-of-mouth for new music releases (Shardanand and Maes, 1995). Different techniques have been applied to modeling and clustering music, including Gaussian and multi-modal mixture models, but self-organising maps (SOMs) remain popular due to the ease with which they can be visualised and explained - an especially attractive attribute in situations where the end result of classification must be communicated to consumers.

2.1.1 Clustering with self-organising maps

SOMs and a similar technique known as muti-dimensional scaling (MDS) are widely applied in music classification, and a number of studies have employed this technique to cluster audio features (Cano et al., 2002; Knees et al., 2006c; Rauber and Fruhwirth, 2001). Studies involving textual features have used SOMs to assess the performance of combined audio features and cultural metadata (Whitman and Smaragdis, 2002), web-based cultural metadata on its own (Knees et al., 2006a, 2005a), and music reviews (Vembu and Baumann, 2004).

Still more applications take advantage of the ease with which the resulting maps lend themselves to visualisation. Several music browsing interfaces are based on SOMs, including an Islands of Music application that produces a island landscape as a metaphor for the geography of music genres (Pampalk et al., 2003; Pampalk, 2001). Other examples include the PlaySOM and PocketSOMPlayer interfaces that navigate music collections by clusters of tracks (Neumayer et al., 2005), and the Traveller's Sound Player, which represents attributes such as genre or tempo as a continuous bar graph around a central dial, very much like the dial on a compass (Schedl et al., 2006).

Finally, SOMs have been used to organise as well as visualise large music data archives. Kurimo (1999) describes the application of SOMs to present documents as smoothed histograms of word categories. The authors also note the use of SOMs to identify new index terms following extraction of potential terms with LSA.

2.1.2 Web-based textual data and lyrics

Scott and Matwin (1998) were perhaps the first to use linguistic features for music classification in 1998, when they explored the effectiveness of representing 6500 folk songs by a vector of hypernym density values. Their approach was similar to LSA in that it abstracted over word counts by taking their hypernyms as LSA abstracts over co-occurences of words and content-bearing words using singular value decomposition (SVD). The authors used WordNet hypernym categories to build the density vectors and compared their technique with a bag-of-words approach using familiar term-document frequencies. A dramatic 47% reduction in classification error was found on a two way classification task between songs about murder or marriage, and those about politics or religion.

A new line of research was started by Whitman and Smaragdis (2002) following their success using a novel technique that simultaneously considered music audio and web-based textual data. The authors classified a small, established set of 25 artists representing five artists from each of five genres (heavy metal, contemporary country, hard-core rap, intelligent dance music and R&B) using acoustic features combined with meta-data drawn from the internet. Although neither text-based data nor music audio alone was successful across all genres, combining the feature sets achieved the correct classification for all songs and the approach could be used to classify diverse styles of music that do not contain lyrics.

This modest success spurred an interest in the use of textual data for music clas-

sification. Knees et al. (2004) achieved up to 87% accuracy using support vector machines (SVMs) to determine genre, working with textual data extracted from web pages. Artist vectors consisted of co-occurrence counts between the artist's name and various indexing words. More recently, the co-occurence of artist and genre names were investigated by Schedl et al. (2006) who predict genre using a probabilistic approach based on page counts of various queries submitted to Google. In addition, Knees et al. (2006) developed their research, using artist similarities calculated from SOM clustering of web-based textual data to reduce the number of pairwise similarity calculations required for the generation of automatic playlists. This cut computation time and subjectively improved musical quality.

Along the similar lines, Vembu and Baumann (2004) explored different weighting schemes for word co-occurrence counts obtained from Amazon music reviews in an attempt to improve music recommendations. SOMs SOMs were once again employed in this work, as they were in further study by Knees et al. (2005). Whilst undertaking to cluster artists according to pre-defined genres, Knees et al. found that techniques using web-based information do not work well in differentiating related genres such as punk, alternative, metal, rock and rock 'n' roll. Further, pop music was intertwined with many genre clusters, indicating how difficult it was to cluster this group. This was in contrast with classical, rap, reggae and blues artists that were clearly distinguishable (Knees et al., 2004, 2005a).

Approaches using web-based data can be successful in combination with musical features, but have several drawbacks. They rely on the ability of search engines to retrieve highly relevant pages, and assume that the content retrieved is accurately described. For example, they are easily misled by band names such as "Daft Punk", referring to a dance music artist (Knees et al., 2005a). They also mimic many of the errors observed with classification using music features alone, wherein they struggle to delimit closely related genres.

Lyrics themselves have been the subject of only a few studies relating to song similarity. Semi-supervised learning was used by Li and Ogihara (2004) to classify the style of singer-songwriters using similarity information obtained from the *All Music Guide*. Separate, unspecified classifiers were trained on music and lyric features, adding samples for which there was minimal cross-classifier disagreement to a seed set at each iteration. Four types of textual features were utilised: bag of words, POS statistics, lexical and orthographic. Bag of words features were the top 200 unstemmed words as measured by TF-IDF (Baeza-Yates and Ribeiro-Neto, 1999) and POS features were

counts for each of 36 possible POS tags normalised for lyric length. Lexical features were a list of 303 function words that represented choices which were syntactic (e.g. preposition phrase modifiers vs. adjectives and adverbs), semantic (e.g. passive vs. active voice) or pragmatic (e.g. first or second person), whilst orthographic features were attributes such as capitalisation and word length distribution. Using a data set of 43 artists and 56 albums, the accuracy of a classifier built using lyrics data alone was almost equal to the accuracy of a classifier using only music-level features. Combining the two feature sets improved performance, achieving precision of 0.7414 and recall 0.7867.

Lyrics were also the focus for Logan et al. (2004), who used probabilistic LSA to retrieve the top ten most similar artists for a series of artist queries. These results were directly compared with the results of a survey in which users were presented with an artist and asked to select the artist who was most similar from a list of ten possibilities. Evaluation compared the number of times LSA agreed with average rank of the artists in the survey list (ARR), and the number of times it agreed with respondents about the most similar artist (FPA). ARR and FPA scores for LSA were compared with the human survey results and scores achieved using music features alone. LSA was better than music features at identifying Latin music, which has foreign words, but noticeably worse at Electronica and Country. In keeping with the results of (Knees et al., 2004, 2005a), the authors reported that the large and diverse category of Rock was not a strength for either feature class.

Two further minor studies employed lyrical features. Dhanaraj and Logan (2005) used SVMs and boosting to automatically predict hit songs, and whilst they found lyrics to be slightly more useful than audio features, the study struggled with several issues. Only 91 identified hit were eligible for analysis and supplementing the data set with more than 1600 non-hit songs caused data imbalance. In addition, the lyrics not cleaned prior to feature extraction resulting in poor results for some features. The second study by Mahedero et al. (2005) conducted basic experiments with lyrics including language classification, identification of lyric structure and five-way topic classification. Unfortunately, the tasks were not particularly challenging. Lyrics were pre-divided into paragraphs representing a clearly recognizable structure before attempting identification of verse and chorus, and the classification task was to identify easily separable topics named “love, violent, protest, Christian and drugs”.

2.1.3 Acoustic modeling of language

Statistical methods have been used to model stress patterns of language in several studies related to speech recognition and synthesis. Arnfield (1996) calculated bigram frequencies for words with co-occurring POS tags and prosodic annotation, and used these to estimate likelihoods for a prosodic tagger taking POS tagged words as input. Training and testing was performed on the prosody and POS tagged Lancaster/IBM Spoken English Corpus (SEC) and achieved up to 91% accuracy. Problems occurred when words were taken out of context, however, for example the phrase "John isn't here" has several different correct sequences of prosodic tags for different intended meanings; prosody might suggest that "John isn't here, he's over there" or that "John isn't here, but Mary is".

The tagger built by Arnfield (1996) identified the stress of whole words, whereas Taylor (2005) modeled the stress patterns within words by combining the output of three Hidden Markov Model (HMM) taggers trained using only vowels and the consonant *y*. One HMM was trained to model each of the three levels of stress taken from a lexicon: primary, secondary and reduced. The combined model achieved 76% accuracy on syllable tags, where the main constraining factor was the phonemic model that underlay stress modeling, which was only capable of handling ngrams up to four phonemes in length. This was compared to actual words in English that could have 15 or more phonemes. Lower accuracy than reported by Arnfield (1996) may be due to the increased difficulty of syllable stress tagging or the fact that the tagger relied on a lexicon. The superior results reported by Arnfield (1996) were achieved with a tagger trained directly on spoken language.

Finally, statistical models of stress and prosodic information have been applied in bigram language models to constrain the possible interpretations of speech acts (Taylor et al., 1996) and to channel recognition of dialogue acts (Shriberg et al., 1998). Further, Chen and Hasegawa-Johnson (2003) investigated the ability of prosody-syntax dependence to reduce data sparseness introduced by prosody dependent language modeling. Experiments on the Radio News Corpus showed that use of their technique in a prosody dependent language model reduced word perplexity by up to 84% compared with a standard maximum likelihood prosody independent language model.

2.1.4 Defining music similarity

Music similarity is difficult to specify as there are so many dimensions along which it may be determined. Songs may share musical features such as timbre, tempo or instrumentation, they may have comparable lyrics, be from the same genre, time period, location, or be similar according to some other specific knowledge such as the career trajectory of a particular musician. Similarity judgments by the same individual may also be inconsistent depending on their state of mind. Development of automatic techniques to extract music similarity presumes a metric against which they may be judged, consequently studies have attempted to define a 'ground truth' for music similarity or analyse the nature of subjective truths for particular modes of comparison.

Cano et al. (2005) completed an insightful study of the topology of music recommendation networks and found a difference between networks based on collaborative filtering and those built by human experts. After reviewing the *All Music Guide, Amazon, Launch-Yahoo!, MSN-Entertainment* and *MusicSeer* networks, and *Art of the Mix* (*aotm*) playlists, it was observed that *aotm* playlists have a similar topology to Amazon "people who bought x also bought y" networks. Both revealed more connections to certain 'hub' artists, indicating that they were substantially affected by artist popularity. In contrast, human experts filter links to hub artists, resulting in some artists with fewer connections. At a global level, all networks were directed, so that artist A being similar to artist B did not presume that artist B was similar to artist A. This could be due to factors such as the variety of an artist's work. For example, Oasis may be similar to The Beatles because some music by The Beatles is similar to all work by Oasis, but the reverse is not true; the music performed by Oasis is fairly consistent and in no way corresponds to all music by The Beatles. Whilst there may be commercial reasons why the music networks studied are directional (e.g. Amazon would not want to repeatedly recommend the same album to one individual), the directional nature of artist similarity is not often acknowledged and may represent as an intrinsic flaw in studies of artist similarity.

Ellis et al. (2002) addressed 'ground truth' in music similarity by exploring the comparative accuracy of three subjective sources of data for 400 artists. These similarity measures were further compared with a web-based survey in which users were presented with an artist and asked to select the artist who was most similar from a list of ten possibilities. The three subjective sources were: co-occurrence in personal music collections (OpenNap); community meta-data, which was converted into similarity

vectors determined by co-occurrence of an artist's name and various predictive terms on a web page; and an 'Erdös' measure that represented the number of similarity links between two artists. Links were taken from the 'similar artists' section of the *All Music Guide* such that if A is similar to B, and B is similar to C, then the Erdös distance between A and C is 2. An alternative version of the Erdös distance was also considered that took into account similarities where A was similar to B because they were both independently related to C, but this was found to be substantially inferior to the standard Erdös distance. The best result with subjective data was found to be only 52.6% using combination of plain Erdös and OpenNap scores - barely above random agreement at 50%. Overall, subjective similarity was found to be highly variable, suggesting that there is no 'ground truth' in music similarity (Ellis et al., 2002).

As a result of these findings, the same research group moved away from 'ground truth' towards 'consensus truth', proposing a way to compute similarity using what they termed 'anchor space' (Berenzweig et al., 2003). Anchor space is a multidimensional space in which each dimension represents the output of a classifier trained on a particular feature or set of classification features. These outputs are posterior probabilities for membership in the anchor classes, with distributions modeled by Gaussian mixture models. Results using low-level music audio features achieved 62% accuracy on a 25-artist test set and 38% accuracy on a 404-artist set (random guessing achieves 25% accuracy). A similar probabilistic model using multi-modal mixture models to jointly model music and textual features was proposed by Brochu and de Freitas (2003).

The group's later research compared similarity determined by songs that co-occurred in *aotm* user-authored playlists, those that co-occurred in OpenNap personal music collections, similarity extracted from information in the *All Music Guide*, human survey data as outlined earlier, and music-level acoustic similarity (Berenzweig et al., 2004). Co-occurrence in music collections was found to be the most reliable indicator although different measures appeared to be providing different information. Song classification using co-occurrence counts in radio playlists and compilation CD track lists was also investigated by Pachet et al. (2001), where song similarity was computed as distance between vectors of co-occurrence counts, or covariance between the same vectors. Expert human judges found that direct co-occurrence counts provided good clustering 70% to 76% of the time, and that clustering appeared to be genre-based. Covariance measures were used to represent song similarity through mutual association with third-party songs but were found to be substantially inferior to clustering with direct counts for pairwise similarity measures.

2.2 Text representation

2.2.1 Latent semantic analysis and variants

Latent semantic analysis (LSA), also known as latent semantic indexing (LSI), shares the mathematical tradition of vector-based semantic space models used in information retrieval (Padó and Lapata, 2007). Originally, vector-based models extracted the meaning of words from a term-document matrix that held a row for every word type and a column for every document in a corpus. Any particular word was represented by a series of numbers, where each number was a count of how many times that word appeared in each document. The resulting vector, or number signature, was projected into a lower dimensional space that no longer held information about the distribution of specific words with specific documents. Rather, the lower dimensional representation extracted information about the similar context shared by groups of words, so that these contexts replaced actual documents as points of reference (Infomap, 2007a).

The problem with term-document matrices is that many words are related but not used in the same document, as might be the case for ‘car’ and ‘automobile’, which are synonyms but applied to different levels of text formality. Instead, LSA replaces term-document occurrences with counts for the number of times a word occurs near to each of what are called ‘content bearing words’ or ‘target words’. Content-bearing words are normally unambiguous, so they form landmarks in relation to which other words may be defined. The example provided in Figure 2.1 shows how content-bearing words help discriminate word meaning

The general idea is that the sum of all contexts in which a word appears, or does not appear, constitutes a set of constraints that largely determine word meaning. Even though it treats language as a bag or words disregarding all syntactic and logical relations, the approach is able to mimic certain aspects of human behaviour relating to language. For this reason it has been proposed as a computational representation of the acquisition and representation of knowledge (Landauer et al., 1998). This may be somewhat overselling its strengths, however, since word order and separation distance clearly help determine the contribution of one word to the meaning of another (Padó and Lapata, 2007).

The word vectors created by counting co-occurrences with content-bearing words are typically very sparse, but this is overcome by projecting the vectors into a lower dimensional space. In addition, dimensionality reduction replaces the reference points provided by actual words with an abstract context. Figure 2.2 shows words that might

Document 1

HOT-FROM-THE-OVEN MEALS: Keep hot food HOT; warm isn't good enough. Set the oven temperature at 140 degrees or hotter. Use a meat thermometer. And cover with foil to keep food moist. Eat within two hours.

Document 2

"Change is always happening," said the ebullient trumpeter, whose words tumble out almost as fast as notes from his trumpet. "That's one of the wonderful things about jazz music." For many jazz fans, Ferguson is one of the wonderful things about jazz music.

words	content-bearing words				
	eat	hot	jazz	meat	trumpet
music			3		1
food	1	2		1	

Figure 2.1: *Frequency counts with content-bearing words (Infomap, 2007a).*

occur in the text window of content-bearing words 'car' and 'driving', and thus share a similar car-related context. Several dimensionality reduction techniques can be applied, most frequently singular value decomposition (SVD) which has been used to reduce term-document matrices (Infomap, 2007a). SVD starts by decomposing the original matrix N into three other matrices: two orthogonal matrices $U'U = V'V = I$ representing the left and right singular vectors (analogous to eigenvectors), and the diagonal matrix Σ of singular values (analogous to eigenvalues). These matrices are related such that:

$$N = U\Sigma V'$$

By equating all but the largest singular values of Σ with zero, the approximation $N = U\Sigma V' \approx U\Sigma V' = N$ is achieved (Hofmann, 1999). Other means of reducing dimensionality include probabilistic latent semantic analysis (PLSA), local linear embedding, and principle component analysis (PCA), although PCA implies a different technique called Information Space (IS). Although very similar, LSA and IS differ in that LSA results in a vector space in which all terms are mutually orthogonal, so reducing the dimensionality produces an approximation of an orthogonal term space. In contrast, IS results in a vector space in which term relations are identically scaled, resulting in an approximation of relations between terms as they actually appear in the raw distribution matrix (Newby, 2000). (Infomap, 2007a).

PLSA is also similar to LSA but takes a more principled, statistical approach by defining a generative model of the data. PLSA uses an aspect model for co-occurrence data that associates an unobserved class variable z with with each word observation so that documents and words are independently conditioned on z. PLSA has been shown

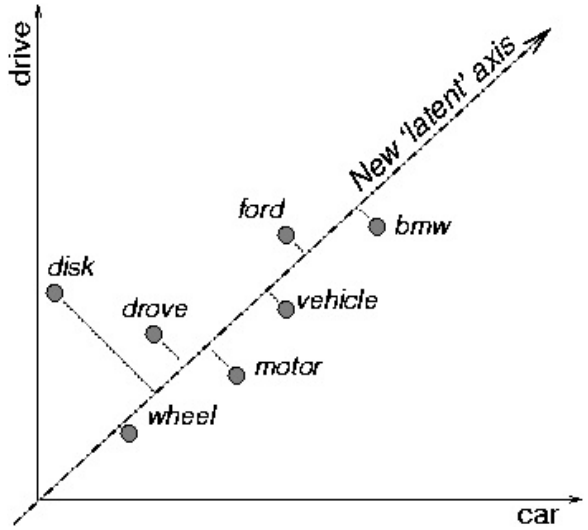


Figure 2.2: *Extraction of abstract concepts with LSA. Figure reproduced from Infomap algorithm description (Infomap, 2007a).*

to demonstrate substantial improvement over LSA (Hofmann, 1999), and is similar to state-of-the-art Latent Dirichlet Allocation (LDA). LDA is a three-level hierarchical Bayesian model in which each word in a document is modeled as a finite mixture over an underlying set of topics and each topic is modeled as an infinite mixture over an underlying set of topic probabilities (Blei et al., 2003). Blei et al. point out that PLSA provides no probabilistic model for the vector representation of documents that indicate probability of membership in each topic, leading to problems with model scaling and overfitting, and uncertainty regarding vector assignment to documents outside of the training set.

Before computing LSA, a term weighting scheme composed of three types of weighting (local, global and normalisation) is usually applied to raw counts. Local weighting is a function of a term's within document frequency, global weighting is a function of the number of times the term appears in a document collection as a whole, and normalisation compensates for differences in document length. Weighting is usually given by

$$L_{ij}G_iN_j$$

where L_{ij} is the local weight of term i in document j, G_j is the global weight for term i and N_j is the normalization factor for document j (Chisholm and Kolda, 1999).

The original matrix may be reconstructed but not perfectly. This is intentional and points to one of the benefits of LSA and its variants; they help overcome an author's random choice of words and the unreliability of data by assuming that the underlying, latent semantic structure of documents is partially obscured by real occurrence counts,

and that it can be estimated by statistical means (Deerwester et al., 1990). In other words, LSA helps differentiate what was actually said, which could be misleading, from what was intended or referenced (Padó and Lapata, 2007).

Once reduced, word vectors can be compared to find words with related meanings, often using the cosine similarity measure. They may also be used to compute meaningful similarity between two documents that share no words in common, since the context of words they do contain point to a certain area of the semantic space. Sentences or whole documents are compared by adding together the vectors that represent their component words. The resulting vectors, sometimes called 'context vectors', can be clustered and used for word disambiguation (Schutze, 1998) or compared with pseudo-document vectors representing queries and used for information retrieval.

In real-world applications of LSA, there are many adjustable parameters that make a substantial difference to performance. The size of the text window that determines proximity to a content-bearing word is one such parameter. This may be a few words, a paragraph, or a whole document (Landauer and Dumais, 1997). The final number of dimensions is pivotal as complex word relationships require more dimensions but fewer dimensions achieve vital context abstraction. Constraints on the selection of content-bearing words is another parameter, including the number of such words and exclusion of high frequency words, and the decision regarding whether to build word vectors for lemmas or all word types is yet another. Further, the resulting co-occurrence matrix may be supplemented with additional information such as part-of-speech tags or multi-word expressions (Infomap, 2007a).

Ultimately, the task of providing the final LSA model with sufficient power to make accurate predictions whilst minimising unwanted distortion of the semantic space is an empirical exercise. Model settings are task-specific, and require a degree fine-tuning to achieve an optimal result (Deerwester et al., 1990).

2.2.2 Sentiment analysis

Research into sentiment analysis can be roughly divided into methods that label words, ngrams or lemmas along various dimensions using manually constructed lexicons (the symbolic approach), and methods that use machine learning to label words or documents with indicators of sentiment. Encompassing both these approaches there is a trend towards a bag-of-words approach that uses a lexicon of words, or individual word features in the case of automated classifiers, as well as a higher-level distinction

between classification according to sentiment categories or along axes of emotion.

Emotional categories are often identified with the 'big six' from psychology, based on anthropologist Paul Ekman's (Ekman, 1972) study of facial expressions. These emotions are *fear, anger, happiness, sadness, surprise* and *disgust*, however it is questionable whether transient emotions such as surprise could have a measurable impact on written expression. Charles Osgood's Theory of Semantic Differentiation (Osgood et al. (1957) cited by Kamps and Marx (2002)) underpins the dimensional approach with his finding that most of the variance in subject ratings of affective words, phrases and texts can be explained by three factors: '*Evaluation*', meaning positive or negative orientation; '*Potency*' meaning strong or weak orientation; and '*Activity*' suggesting active or passive orientation.

Possibly the simplest approach to sentiment analysis was taken by Turney (2002) who used the sum of the orientations of all words in a document and achieved between 65.83% and 84.0% accuracy on different review types. Turney took context into account by using tuples of adjective-noun or adverb-verb combinations and calculated positive/negative orientation by submitting phrases to the Altavista search engine along with an orientation word ('excellent' or 'poor') and counting the number of times each phrase occurred within a ten word window of the orientation word. Those words that were more frequently found in close proximity to 'excellent' were positive, and those that more often occurred near 'poor' were negative.

Early work based on Osgood's theory included Anderson and McMaster's 1982 paper on modeling affective tone in text. This research used a list of the one thousand most frequent words in English annotated with semantic information corresponding to the *Evaluation, Potency* and *Activity* dimensions. Growing interest was spurred by the work of Hatzivassiloglou and McKeown (1997) on predicting the semantic orientation of adjectives by automatically grouping documents into clusters with the same orientation and manually labeling the clusters as positive or negative. Kamps et al. (2004) also use Osgood's theory as the basis for determining word orientation. They calculate the minimum path length (MPL) between a word and two bipolar terms for each of Osgood's three dimensions: good/bad for *Evaluation*, strong/weak for *Potency*, and active/passive for *Activity*. MPL is defined as the minimum distance between synset nodes in WordNet, where a unit is a 'hop' from one synset to another (Figure 2.3). Orientation of a word in a given dimension is determined by the relative distance to each pole in the respective pair. Interestingly, the same number of words have connections to each of the bipolar pairs: exactly 5410. Using this technique, accuracy of 76.72% is

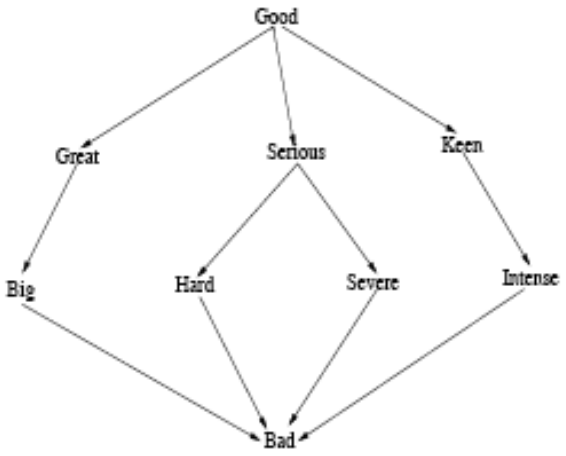


Figure 2.3: *Four ways to get from good to bad in three hops using WordNet synset categories. Figure reproduced from Godbole et al. (2007).*

achieved when compared against the manually constructed word list provided by the *General Inquirer* (Inquirer, 2007).

Much research has focused on the *Evaluative* and *Potency* dimensions of Osgood's theory, whilst the *Activity* dimension has been somewhat neglected. This may be symptomatic of research in sentiment analysis focusing on the evaluative aspect of opinion towards an object, rather than the emotion of the writer per se. Martin and White's (2005) Appraisal Theory was developed specifically for the analysis of evaluation in text as observed in reviews (e.g. for music, books). Appraisal Theory uses three main dimensions: '*Attitude*' includes things such as personal affective state, appreciation, and social judgement as well as attitude orientation (positive/negative); '*Amplification*' indicates whether an attitude is low, median, high or maximal; and '*Engagement*' reflects the degree of commitment to an opinion (Martin and White (2005) cited by Argamon et al. (2007)). Appraisal Theory has been applied, for example, by Taboada and Grieve (2004) who found that different types of reviews contained varying levels of attitude types.

Other work has focused on how context affects word orientation (Wilson et al., 2007), since this may be changed or intensified by negation (e.g. 'not good' vs. 'not only good but amazing'), change with domain topic (e.g. 'bad' in a rap song can mean 'great'), or be determined by the perspective of a speaker, in the way that a comment about an 'expensive dress' might be a compliment or a criticism. Determining the orientation of ngrams instead of individual words is one way to circumvent issues with negation, or context may be predicted using syntactic or alternative semantic clues, such as verb tense, voice, or sentence structure.

2.2.3 Text processing

Tokenisation, part-of-speech (POS) tagging and lemmatisation are forms of text processing that are stand-alone forms of analysis or preliminary steps for partial parsing. Tokenisation is simple in concept and largely considered to be a mature technology, but can be a messy task, especially when handling freeform text such as email and web postings, or highly technical domains such as biotechnology. More challenging genres are those that present unique and complex 'words' incorporating unexpected capitalisation and punctuation.

Part-of-speech tagging is a highly developed field and most modern taggers are reliable, robust and efficient with minimal error rates of between 1 and 5 per cent (Abney, 1996). Tagging is required to successfully handle irregularities commonly found in natural language and be applicable to diverse texts with consistently high accuracy. Knowledge of the prior distribution of words greatly improves tagging accuracy and out-of-vocabulary words are responsible for most errors. Taggers may be divided into those that are rule-based, statistical taggers including Markov models, maximum entropy models and neural networks, and transformation-based taggers that combine these two approaches by using supervised machine learning to infer rules that are then applied as with a rule-based tagger. Statistical techniques rose to prominence in the late 1980s and 1990s (Samuelsson and Voutilainen, 1997) and are the current state-of-the-art, however other taggers remain popular for their portability to new domains and their applicability in situations where there is no sufficient tagged corpus with which to train a statistical tagger.

Lemmatisation is the process of assigning words their base form, or lemma, based on knowledge of their part of speech e.g. 'walk', 'walked', 'walks' and 'walking' all share the same lemma: 'walk'. Lemmatisation is closely related to stemming, which also finds the base form but without contextual knowledge such as the POS tag. Whilst easier to implement, stemming may incorrectly assign some tags (e.g. 'good' as the lemma for 'better').

Chunking is the task of dividing text into syntactically related groups of words, such as noun or verb groups. There are many variations on how such groups are defined, and thus many different styles of chunking, but essentially there are four types of systems: rule-based or transformation-based (although transformation-based taggers require prior statistical learning), memory-based systems, statistical systems including Markov models, maximum entropy and support vector machines, and combined

systems. At the CoNLL-2000 shared task on chunking, combined systems performed exceptionally well through weighted interpolation of system voting (Tjong Kim Sang and Buchholz, 2000).

2.3 Data mining

2.3.1 Self-organising maps

The Kohonen self-organising map, commonly referred to as a self-organising map (SOM), was introduced by Kohonen in 1982 as a neural network algorithm for unsupervised learning that facilitates interactive visualisation and analysis of high-dimensional data through its representation in a lower dimensional space. Each data point becomes associated with a node that is simultaneously associated with a point on a regular two dimensional grid that is rectangular or hexagonal depending on the purpose of visualisation. Hexagonal formations provide visually even distances to neighbouring units making clusters easier to view, while rectangular grids provide more space for clear labeling. Data points that are close together in high-dimensional input space are close to each other on the grid, and high density areas in input space are mapped to proportionately more network nodes.

The vector defining a data point determines the network node with which that data point is associated. Each network node is represented by a vector known as the 'prototype', 'codebook' or 'weight' vector, that exists in the same dimensional space defined by the data vectors. A data point is represented by the network node with the most similar prototype vector, where similarity is usually determined by Euclidean distance.

Map training begins with initialising the prototype vectors, which can be achieved in several ways. Vectors may be initialised randomly, or initialised linearly along the x or y axis, or along the greatest eigenvectors (Figure2.4). They can also be selected randomly or methodically from the input data set. The choice of initialisation is significant since it affects the local optima at which the network settles, resulting in variable clustering of the final map. Ideally, ten or more randomly initialised maps are trained and reviewed, and clusters are selected that are apparent in all, or most, of the final SOMs. Alternatively, one map is selected that best represents the average clustering across all maps, whilst differences between maps are observed as a valuable source of additional information. Various algorithms have been proposed to make the process

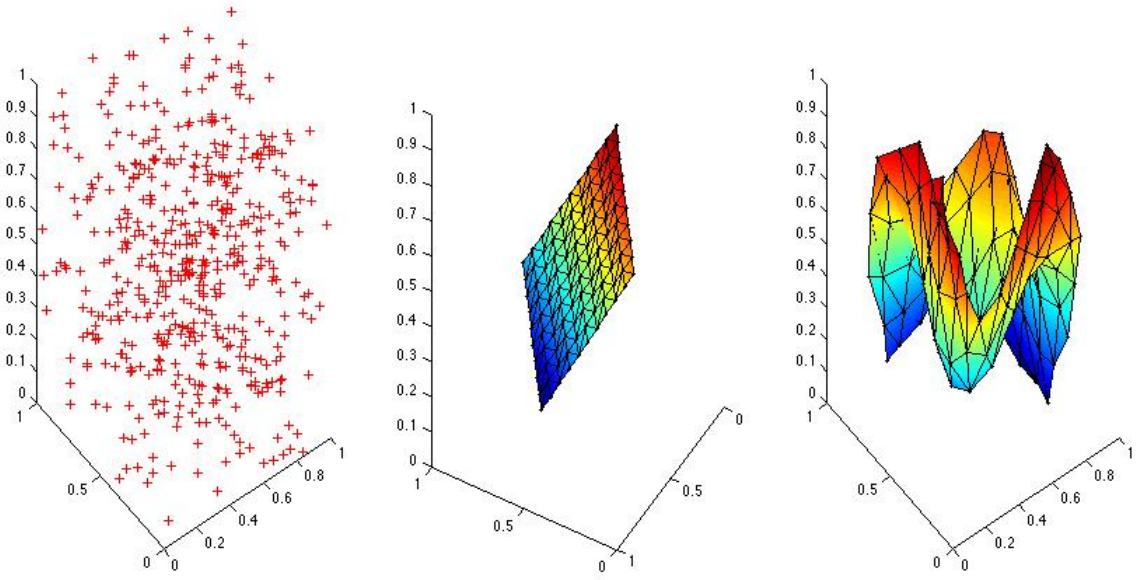


Figure 2.4: *Training an SOM using linear initialisation along the two greatest eigenvectors. During training the map folds to better represent the data. Figure reproduced from the SOM toolbox demo1.*

of SOM analysis less labour intensive and more robust. These include algorithms to assist in the selection of SOM clusters, such as those referred to by Samsonova et al. (2006) or the best representative SOM (Rousset et al., 2006).

Once prototype vectors have been initialised, the next step is to randomly select a vector from the input set and identify the prototype vector with which it is most similar. This is the vector for the data point's best matching unit (BMU). As each BMU is selected, learning is competitive, since the BMU prototype vector is updated to more closely resemble the data vector, and also cooperative, since the prototype vectors of neighbouring network units also are updated. Updates occur according to the following equation, which takes the difference between the input vector and the current prototype vector weighted in proportion to the learning rate and the neighbourhood function, and adds this to the current prototype vector:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)]$$

where m_i is the prototype vector for the current unit i, x is the input vector, h_{ci} is the time decreasing neighbourhood function, t is the iteration time step and α is the learning rate.

Basic variations on this training procedure are batch training, in which all data points are selected before the updating of prototype vector weights, and online or sequential training, in which one, or a number, of data points are considered before

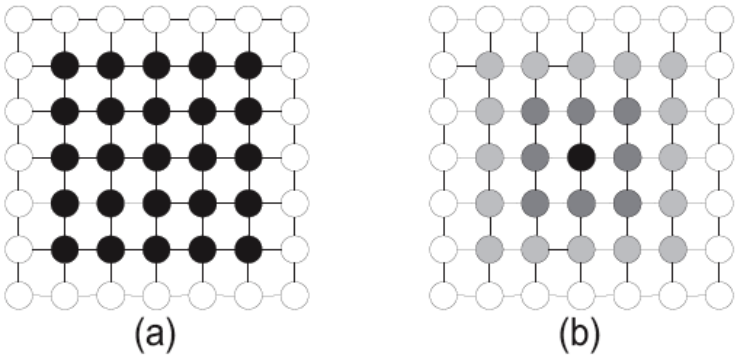


Figure 2.5: *Different SOM neighbourhood functions: bubble (left) and Gaussian (right). Figure reproduced from Roiger (2007).*

updating the prototype vector weights. Online training is better suited to very large data sets, whereas batch training is less dependent on the order in which data points are selected.

The neighbourhood function determines which prototype vectors are updated after each training procedure, whilst the learning rate regulates how much they are updated. Figure 2.5 shows examples of a bubble neighbourhood function (a) and Gaussian function (b), where units are shaded gray corresponding to how much their prototype vectors are adjusted when a data vector is associated with a node in the centre (Roiger, 2007). Neighbourhoods are commonly bubble-shaped, Gaussian or truncated Gaussian with a variable radius, and are distance dependent, so that whilst node A may be in node B's neighbourhood at the start of training it may not be within its neighbourhood by the end. In addition, the neighbourhood range may be purposefully reduced over the course of training so that all prototype vectors are updated in the first round of training, and only the vectors of very closely neighbouring nodes are updated later on (Roiger, 2007). Phased training with different neighbourhood settings and learning rates for each phase is a common alternative.

Learning rates are decreased over time so that the map adjusts itself more to the data in the early stages of training and less as the map becomes more stable (Roiger, 2007). Adaptation of the weights in the prototype vectors aims to achieve greater contrast in weight patterns, and is assisted by the neighbourhood function, which causes nodes only to respond to input data associated with neighbouring nodes if they have similar prototype vectors. Contrast enhancement is desirable as it emphasises the strongest correlations in the input data and de-emphasises the weakest ones, helping the neural network to selectivity represent key features in a parsimonious model. Ultimately, as units become more selective for certain features, they allow other units to become more

representative of other features, which means better overall performance. This is part of self-organised learning, in which units evolve their own conditionalising function as a result of complex interaction with other units. Training stops either when some stopping criterion is reached, such as a predefined number of iterations, when change in some quality metric is less than a given threshold, or when a stable map organisation is attained. Early stopping helps to avoid overfitting that is a common problem with neural networks.

SOMs are particularly suited to data analysis where there are a relatively large number of attributes, but a balance must be found between simplicity of the final representation and adaptability to fit complex data. Data representation is measured by average quantisation error between data vectors and prototype vectors, whilst topology accuracy assesses whether data points that are close in input space are also close in output space, and vice versa. Topological representation is measured by the percentage of data vectors for which first and second BMUs are not adjacent units. 'Folding' of the map to cover high-dimensional space reduces quantisation error, but increases topology error, so both measurements must be considered when assessing overall map quality. Other neural network SOM variants, such as the Neural Gas algorithm proposed by (Fritzke (1995), cited by Samsonova et al. (2006)) and growing hierarchical SOMs (Dittenbach et al. (2002), cited by Samsonova et al. (2006)) provide better topological representation than Kohonen SOMs, closely fitting the data. In comparison, Kohonen SOMs offer simpler and more systematic analysis independent of variations in the input.

2.3.2 Cluster analysis

SOMs are themselves a prototype-based means of clustering since any number of data points may be associated with a particular node in the network. It is sometimes desirable, however, to measure the validity of the SOM clusters and quantify the larger clusters that can be identified by viewing the map.

Clusters should divide the data into groups that are meaningful and useful. There are many objective metrics of clustering validity, of which the Davies-Bouldin Index (Davies and Bouldin, 1979) is one that is commonly used (Tan et al., 2005). The Davies-Bouldin Index takes into account the average error for each cluster in an environment, measuring intra-cluster distance as the average of all pairwise distances between a data vector and the cluster's centroid vector, and the inter-cluster distance

as the pairwise distance between two clusters' centroids. The core idea is that points within a cluster should have high internal similarity whilst simultaneously maintaining low intra-cluster similarity. Good clustering minimises the Index computed according to the equation:

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\left\{\frac{\Delta(C_i)+\Delta(C_j)}{\delta(C_i,C_j)}\right\} \tag{2.1}$$

where $\Delta(C_i) = \frac{\sum_{i=1}^{n_i}\|g_i - g_{ic}\|}{n_i}$, $\delta(C_i,C_j) = \|g_{ic} - g_{jc}\|$ and n_i is the number of points in cluster C_i. The centroids of clusters C_i and C_j are g_{ic} and g_{jc}, and k is the number of clusters.

Chapter 3

Data

The raw data for this study took two forms: song lyrics and song playlists. Personalised internet radio service Pandora were originally proposed as the source for lyrics through their existing relationship with the University of Edinburgh, however this did not prove feasible. Instead, lyrics were obtained from public internet websites `www.azlyrics.com` and `www.oldielyrics.com`. These sites acquire lyrics through user upload, and may also download from other lyrics websites. Due to the manner of acquisition, the lyric data was freeform, irregular and required substantial cleaning before it could be analysed.

Publicly available data on song co-occurrence in user-authored playlists from the Art of the Mix website (*aotm02*) was the basis of the gold standard similarity metric. This data was one of four sources of human opinion used to compute music artist similarity in studies by Columbia University (Columbia, 2007). The other three data sources were inappropriate since two focused on artist similarity ignoring distinctions between individual songs, and the third, co-occurrence in OpenNap online personal music collections, was encoded using a mapping from numeric codes to songs and artists that was not compatible with the key used for the *aotm02* data and not publicly distributed. In addition, it was clear that personal music collections often include a series of complete albums, or sets of albums, by a given artist, and hence may not be as useful for judging the similarity of individual songs as song clusters provided by playlists.

3.1 Gold standard playlist data

Data for user-contributed playlists collected from the *aotm* web site in January 2003 was downloaded as a single file with one playlist per line. This was reformatted to appear in the configuration 'artist:song artist:song' to ensure that songs which had the same name, but were by different artists, were uniquely identified. Statistics for the *aotm02* data set are shown in Table 3.1, however the actual number of unique artists is lower than suggested here due to duplication found within the

The numeric artist:song identifiers from the playlist data were translated into a dictionary that mapped each song identifier to a list of all locations in which it was found. Locations were described by the playlist number followed by the playlist index (e.g. first, second, third song). Using this dictionary, it was possible to quickly compute the number of times any two songs appeared in a playlist together, as well as their separation distance. All songs that only occurred in one playlist were discarded, reducing the initial set of 58,639 songs to 45,030.

It was hypothesised that songs appearing close together might be more similar than those that were further apart. To test whether proximity was a usable attribute, various thresholds on separation distance were tested. Unfortunately, it became apparent that the data was too sparse to support such restrictions, and all requirements for song proximity were removed. Even when considering all possible playlist pairings, similarity counts were sparse, with the vast majority of songs co-occurring only once and almost all song pairs appearing a maximum of six times.

Smoothing was considered as an alternative means of taking proximity into account, by adding an 'alpha factor' to each co-occurence count equal to one divided by the separation distance between the two songs in a pair. This type of smoothing would give extra weight to songs that were close together, whilst helping to overcome limitations imposed by discrete data values. It was not clear, however, that the adjustment represented a plausible way to compute lyric similarity. Songs that were next to each other would have a separation of one, and therefore an alpha factor equal to one, disproportionately raising the total count of the song pair compared to counts across the data set. Whilst it would be possible to scale the alpha factor to reduce this effect, the choice of scaling would be heuristic at best and was therefore avoided.

For more a robust means of comparison that also helped overcome data sparsity, each song was represented as a vector of co-occurrence counts between itself and the top 100 most frequent songs, where frequency was determined by co-occurrence with

Total playlists (201 are empty after parsing)	29,164
Total unique songs	218,260
Total artist names	60,931
Distinct regularised artist names	48,169
Average songs / non-empty playlist	19.8
Average artists / non-empty playlist	17.1

Table 3.1: *Statistics for the aotm02 data set (Columbia, 2007).*

songs that appeared in more than one playlist. This restricted the data to just those songs that co-occurred with at least one of the top 100 songs, and reduced the data set to 35,843 songs. The top 100 songs were chosen since this was the cut-off value at which the size of the captured data set tended to stabilize. Further raising the cut-off value only served to increase computational demand by lengthening the song vectors without adding much in the way of information or size to the existing corpus. Whilst it was possible to also include a half count for songs that were indirectly related, e.g. if songs A and B are in different playlists that both contain song C they are indirectly related through C, however research into music similarity by Ellis et al. (2002) shows that such indirect relations are unreliable. *aotm02* code lists.

The approach described above helped reduce data sparsity and slanted the data set towards pop music that was more likely to co-occur with the most popular 'hub' songs. It is possible that music popularity dominated over similarity in the co-occurrence counts both due to the choice of data described and because the *aotm02* dataset is inherently biased to making more connections to hub artists as described in Cano et al. (2006) and discussed in detail in Chapter 2. What makes this data choice interesting, however, is that it selects songs that have proven difficult to distinguish in the earlier research using discrete, mainly homogenous feature sets (Logan et al., 2004; Knees et al., 2005a).

3.2 Lyric Data

Following the withdrawal of Pandora, it was necessary to download lyric data from the web. Dhanaraj and Logan (2005) used Astraweb Lyrics Search since the lyrics on this site are partially standardised, however both the artist and album names are required to submit queries to this site, introducing many possibilities for query error.

In addition, album information was not easily available. The popular lyrics website `www.azlyrics.com` used by Logan (Logan et al., 2004), and later its sister website `www.oldielyrics.com`, were selected as alternate sources and a crawler was written to submit queries to the site and retrieve any lyrics that were found. Starting with a superset total 218,260 songs and 48,169 unique artists, this was narrowed to 2392 songs and 496 artists. Due to time restrictions, it was not possible to obtain the genre for each song used, although such information would be obtainable publicly from the *All Music Guide* at `www.allmusic.com`.

Research by Logan et al. (2004) into the application of LSA to lyrics used 41,460 songs, however the quality of this data is questionable as there was no discussion of data cleaning and duplicates were found in the Columbia University code lists used in this work further reducing reliability. Mahedero et al. (2005) suggested that agreement between different transcriptions of the same song may be in the region of 82% to 98% for very well-known songs, and Knees et al. (2005) acknowledge that lyric quality can be an issue, proposing a novel method of simultaneously downloading and cleaning lyrics using multiple sequence alignment. The technique resulted in median recall above 98% on 258 songs, but precision was not discussed and the authors note that the procedure would benefit from prior cleaning of lyrics to remove words and sequences that are not alignable.

Fewer lyrics are used in this research, largely as a result of challenges inherent in cleaning the code lists and playlist data (see next Section) and also due to ambiguity in the format of artist names in web databases. More data might have been obtained by sorting songs for download by artist, seeking an artist's page, and matching available song titles to desired songs using a non-binary similarity measure. Future research might also search other lyric sites. For the purposes of this study, however, a smaller number of songs was suitable as it reduced the time involved in data cleaning and enabled the full data matrix to be manipulated in memory, making calculations and visualisations faster and easier.

3.3 Data Cleaning

3.3.1 Playlist data and code lists

Raw data obtained from the *aotm* playlists was converted into numeric codes by Columbia University who initially gave a different number to each unique name string. Every ef-

Identified by alphabetical sorting		Not identified by alphabetical sorting (shown: subset of 12 entries)	
papa s got brand new bag	#213107	sargent pepper	#12856
papa s got brand new bag part 1	#213109	sargent pepper s lonely hearts club band	#12927
papa s got brand new bag part one	#213141	sgt pepper	#12858
papa s got brand new bag pt 1	#213100	sgt pepper s lonely hearts club band	#12612
papas got brand ne	#213076	sgt pepper s reprise	#12935
papas got brand new bag	#213203	sgt peppers lonely heart club band	#13045

Table 3.2: *Alphabetical sorting to remove song number duplicates.*

fort was made to de-duplicate these entries for the artist code list used in their research so that all typographical variations of a given name were mapped to a single number. For example, this author found eight variations of 'Faith Hill and Tim McGraw', which were eventually mapped to two codes: one for 'Faith Hill and Tim McGraw' and another for 'Tim McGraw and Faith Hill'. Despite this, some cases persisted in which artists were associated with two or more codes. The Cure and REM are two popular bands featuring strongly in the data corpus that were associated with three codes each: ('1985 cure' #33984; 'thecure' #35572; 'cure' #2324) and ('rem' #374; '10 rem' #805; 'r e m' #10449) respectively. Until this duplication was identified and amended, there was considerable confusion when converting between song names and codes in order to download lyrics and construct the gold standard similarity matrix. In total, 35 duplicate songs were removed from the data set for this research representing 1.4% of all data.

In addition, initial expectations were that both the song and artist code lists had been cleaned, but song code lists had not been de-duplicated, presumably because this was not necessary for the Columbia University study. Sorting alphabetically highlighted some duplications, such as those for the song "Papa's got a brand new bag", but other irregularities, such as varying use of articles, conjunctives, prenominal numbers and abbreviations, were not always identified by this process (Table 3.2). Approximately one quarter of songs in the code list had multiple entries due to spelling errors and naming variations.

Lack of a master list with standardised spellings for artist names further complicated conversion between codes and accurate names that could be used to query the selected lyrics websites. The scale of the problem was considerable. Out of 35,843 queries submitted to `www.azlyrics.com`, artist:song pairs derived directly from the Columbia University data resulted in only around 150 successful downloads. Repeat-

Before correction	After correction
jimmie hendrix	jimi hendrix
mavin gaye	marvin gaye
somewhere over rainbow	somewhere over the rainbow
what s going on	whats going on
jesus and mary c	jesus and mary chain
why does always rain on me	why does it always rain on me
i am poseur	i am a poseur
elton johh	elton john
21 john lennon	john lennon
london suede	suede
rod steward	rod stewart
beastieboys	beastie boys
white srtipes	white stripes
ccr	credence clearwater revival
kinks	the kinks
death car for cutie	death cab for cutie
portis head	portishead

Table 3.3: *Examples of spelling and typographical errors in original code lists.*

ing the queries to sister website `www.oldielyrics.com` on the hypothesis that songs from 2002 may have been archived from the main site, expanded the retrieved set by around 100 lyrics - still far short of the required data.

Blind mapping of codes to names produced poor download results, but neither was it necessary or desirable in the given time frame to create a master code list for all 218,260 songs and 48,169 artists in the *aotm02* data. Instead, a random subsection of around 14,000 songs from the 35,843 possibles were roughly edited for errors based on the author's fair knowledge of music and grammatical considerations. Where possible, corrections were automated, such as the insertion of the word 'the' before band and song titles. Corrections included re-introduction of removed stop-words from the set $\{$*the, a, and, it*$\}$ and amendment of typographical errors including additional, skipped or switched letters, abbreviations, phonetic mistranslations, erroneous numbers and words, truncated words and incorrectly conjoined or separated words, in addition to variable interpretations of song titles. Some example errors are shown in Table 3.3.

The resulting partially clean list of song titles was used to download a total of 2,461 lyrics that were further cleaned to produce the final data set. Song and artist names were then re-encoded in their numeric representation to compute the desired song similarity vectors for the gold standard similarity matrix. In order to convert back to numeric codes, for the portion of data included in the final set, a duplication

dictionary was created that mapped from all possible representations of a song to its standardised entry. Care was taken to ensure that combined song and artist titles were correctly identified, since one song title could be covered by many different artists, e.g. 13 versions of the song "Stagger Lee" were identified, resulting in 13 different codes. This dictionary was used also to standardise playlist entries before computing similarity counts to ensure that as much data was extracted as possible, helping to overcome data sparsity.

Any further research on the *aotm02* data should consider full de-duplication of the code lists and construction of a separate clean index of numerical codes and correct names, for both songs and artists. This would improve lyrics download and considerably speed preprocessing.

3.3.2 Lyrics

A study by Dhanaraj and Logan (2005) observes that common inconsistencies in lyric transcription, such as the denotation of a chorus by the word 'chorus', were the probable cause of poor performance for repetition features in their work. Knees et al. (2005) discuss the subject of lyric quality in detail, highlighting multiple errors that may arise in downloaded lyrics. They note spelling errors and slang spelling, such as *cause, coz* and *cuz* for the word 'because'; differences in semantic content when words of recorded lyrics cannot be clearly distinguished; multiple versions of lyrics, such as might occur with 'live' and 'radio' edits; annotation of background singers, spoken voices and sounds; annotation of lyric structures such as the chorus, verse and performing artist; and abbreviated repetitions that minimise the effort involved in lyric transcription.

Consistent with these findings, downloaded lyrics for this research contained many aberrations that were likely to detract from results. This was a particular concern since data was not being pooled by artist, helping to marginalise error. The following faults were corrected:

- **Annotations:** Lyrics were searched for notation that indicated chorus repetitions and replaced with the chorus written out in full. The search and extraction procedure was robust to variants in chorus notation, such as indications of chorus repetition by phrases such as *chorus x2, repeat, 2x repeat, repeat 2 times* and *chorus*. In addition, annotations identifying a change in performing artist (for groups and duets), information about individual repeated lines (e.g. marked by

x2), guest appearances (*feat xxx*) and variation in song tails, where some songs repeated the artist and title at the end of lyrics, were removed or replaced with the full lyric or blank lines as appropriate.

- **Instrumentals:** Lyrics websites appeared to be automatically created from album track listings, resulting in web pages for instrumental songs that contained only minimal holding text. Instrumentals were removed following parsing to extract lyrics information. Files that were empty after this process were checked and removed. No false positives were found.

- **Single-phrase lyrics:** Some lyrics were inappropriate for inclusion as they were too short, e.g. "Futterman's Rule" by The Beastie Boys, for which the lyric is, *"'When two are served, you may begin to eat', Gene Futterman"*, and "Wild Honey Pie" by The Beatles, which simply repeats the phrase *"Honey pie"*, ending with *"I love you honey pie"*. Such short lyrics were deemed to be instrumental and not included the data set. Single phrase lyrics were identified by very low statistics for the number of words in class categories (e.g. verbs, adjectives), and were checked and deleted from the data set.

- **Foreign language and UTF characters:** Songs that were entirely in another language or contained non-standard UTF-characters were filtered out during preprocessing as the `txt2onto` chunker (see Section 4.1) used would only accept standard text. This included non-standard line returns and punctuation that may have originated from non-English keyboards. The few illegal UTF characters found were converted manually, and foreign language songs removed from the data set. Songs that contained some foreign language but were predominantly in English were converted to plain text.

- **Single paragraph:** Some songs had been posted to the web with no line returns and had to be converted manually to a poetry format. Single paragraph files were identified during extraction of repetition features, where they were distinguished by unusual or undefined repetition ratios for unique whole lines to repeated lines. Many of these instances were attributed to a single artist, indicating that the source of the problem was initial lyric upload.

The clean-up procedure described above reduced the data corpus from 2461 songs to a final count of 2392 songs. Further aberrations may still exist but could not be found and corrected in a reasonable time period. The total percentage of corrupted

Total songs before clean-up	2461
Duplicate (deleted)	35
Instrumental (deleted)	19
Single-phrase lyric (deleted)	10
Foreign language (deleted)	4
Unknown UTF-characters (corrected)	27
Single paragraph (corrected)	14
Total songs after clean-up	2392

Table 3.4: *Data cleanup statistics.*

	Total data	Ebonics on
Number of documents	2392	1037
Number of artists	496	228
Number of words (excl. punctuation)	497,063	225,964
Average words per document (excl. punctuation)	208	218
Dictionary size (excl. punctuation, ignoring case)	19,578	12837

Table 3.5: *Final data statistics.*

data removed or corrected from raw internet download was 4.4%. For future work, this should be considered and checks put in place to handle the faults identified above. A breakdown of data clean-up is presented in Table 3.4, and statistics for the final data set are shown in Table 3.5.

3.4 Ebonics

Lyrics were expected to contain a fair amount of slang, but it transpired that the vast majority of exceptions to regular English were either spelling and typographical errors or ebonics terms. Ebonics, also known as AAVE (African American Vernacular English) differs from regular English in pronunciation, grammatical structures and vocabulary, as well as unique and informal slang expressions. Many ebonics words and turns of phrase, such as *ain't, gimme, bro* and *lovin* have now become common parlance, as suggested by Figure 3.1 which shows the word frequency of word corrected by the ebonics dictionary (see next Section) in red. Regular English is shown in blue.

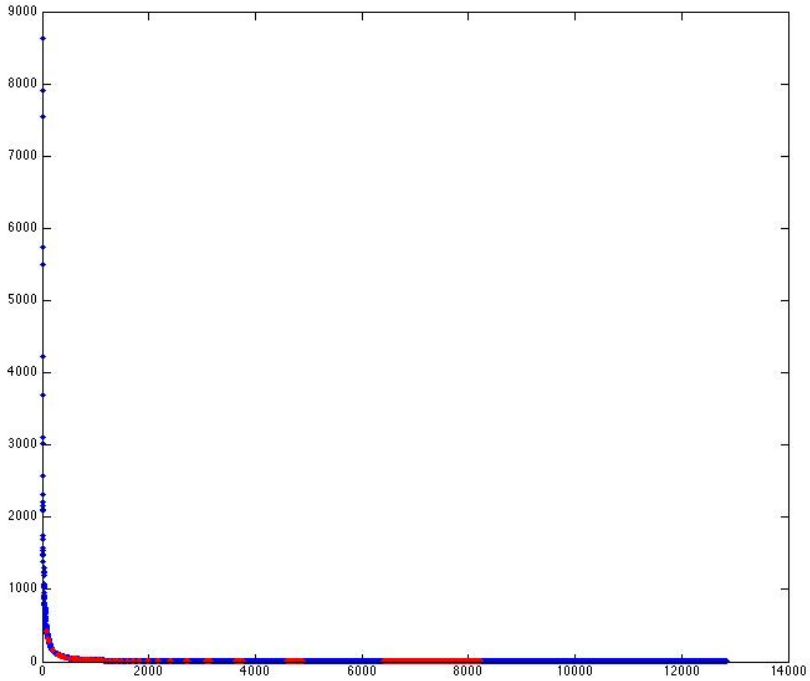


Figure 3.1: *Frequency of ebonics in lyrics. Word frequency is shown on the y axis, and dictionary entries appear along the x axis sorted first by frequency and then alphabetically by artist. Ebonic words are in red, and regular English is in blue.*

Although some of the words in the ebonics dictionary are misspellings, the high proportion of red in the curve suggests that many frequent and differentiating words used in lyrics are ebonics terms that have largely replaced their regular English counterparts e.g. *cos, coz, cuz* and *cause* for 'because'. Red streaks in the tail suggest that certain artists have very high proportions of spelling mistakes and ebonics terms, whilst others have negligible ebonics content or only use very common derived terms.

Overall, the graph indicates that ebonics could potentially have a substantial impact on the results of lyric analysis. Ebonics irregularities would likely present as OOV words to a POS tagger and result in poor performance of features relating to grammar. In addition, a large number of POS tagging errors would have a detrimental effect on chunking since the rules used by the `txt2onto` chunker (see Section 4.1 to identify phrase groups check first the nature of the word elements themselves and then their part of speech. For example, rules that permit optional adverbials and adjectives in passive verb groups, such as 'was probably sent' or 'was sent away', require that the tag for 'was' be VBN or VBD (since POS taggers cannot reliably distinguish the two) (Grover and Tobin, 2006).

It was hypothesised that removal of ebonics irregularities would improve part-of-speech tagging, and thus improve both chunking and clustering achieved with grammar features. In addition, it was desirable to understand the impact of ebonics upon results achieved with semantic analysis techniques such as IS and sentiment analysis.

3.4.1 Ebonics dictionary

To facilitate ebonics correction, an ebonics dictionary was created mapping OOV words to standard English. The dictionary encompassed slang, ebonics, spelling errors, unintended word conjunctions, phonetic translations, named entities and abbreviations that represented approximately 15% of all words. Examples of contexts, translations, and final dictionary entries are shown in Table 3.6. In addition, whilst not ebonics per se, the appearance of many named entities as OOV words highlighted the use of reference to cultural icons, famous people, places and brands as a salient attribute of black urban music (see examples in Table 3.7) that warrants further investigation. Unfortunately, such work was beyond the scope of this research.

Effort was made to correct every ebonics, spelling and related error in the 1037 songs identified as *ebonics on*: all songs in the alphabetically sorted data set from 112 "Anywhere" to Kottonmouth Kings "Bump". Correction was subject to constraints, however, such as instances in which words had regular English definitions but were used in a slang context. Many of these instances were merely curiosities since they would be tagged with the same POS regardless of their interpretation, e.g. *grinders* meaning *sandwiches*, or *deco umbrella* meaning *average uterus*. Others would produce inaccurate tag counts, e.g. *sloppy joe* meaning *hamburger*, but did not dominate text. Of more concern were phrases that would result in multiple incorrect POS tags, especially since incorrect tags were likely to negatively affect a tagger decision on subsequent words. Phrasal errors were sometimes due to surprising entries in the CELEX database (an online English lexicon; see next Section), e.g. for *"shama lama ding dong dip dip dip"*, all words but *ding* and *dong* were found in the CELEX lexicon. Slang was also a problem where it could not be translated or constituted a one-to many relation, e.g. *"doing hunny in a sixty-five"* meaning *speeding* (recklessly on an American freeway). Finally, certain words were manually edited into the ebonics dictionary as they were common enough to be identified by CELEX (*cos* $\Rightarrow$ because ; *wanna* $\Rightarrow$ want to ; *wit* $\Rightarrow$ with). These additions brought the size of the dictionary to 2940 entries.

Context	Translation	Example entry
Slang		
like knievel he weeble	he wobbles like Evil Knieval	weeble / wobble / 1 0
pack of strohs	group of women resembling whores	strohs / whores / 1 0
fot eh revy horton heat	asshole eh, racy, sexual guns	fot / asshole / 1
the mise is broken	the luck is broken	mise / luck / 1
my jalapeno's red	my girlfriend is sexy	jalapeno's / girlfriend is / 1 0 2 0
Ebonics		
the bro throw up	the guy throws up	bro / guy / 1
you've gotta move	you've got to move	gotta / got to / 1 0
rock'n'roll ain't noise	rock and roll isn't noise	ain't / is n't / 1
gimme a break	give me a break	gimme / give me / 1 0
who you wannabe	who do you want to be	wannabe / want to be / 2 0 1
happ'ning	happening	happ / / 'ning / happening / 2 0 1
Phonetic transcription		
in the citaaay of good	in the city of good	citaaay / city / 0 1
yeahhh	yeah	yeahhh / yeah / 1
shama lama ding dong	la la la la	ding / la / 1
don't knooooooooww ohh ohh	don't know oh oh	ohh / oh / 1
Typographical / abbreviations		
atleast you know	at least you know	atleast / at least / 1 1
c.o.d.	cash on delivery	c.o.d / cash on delivery / 1 1 1
cleanging	cleaning	cleanging / cleaning / 1 0

Table 3.6: *Token correction and stress pattern encoding in the ebonics dictionary.*

Context	Translation	Example entry
Cultural icons		
like Captain Picard we come	n/a	Picard / Picard / 0 1
I robbed Liberace	n/a	Liberace / Liberace / 1 0 2 0
untouchable like Elliot Ness	n/a	Ness / Ness / 1
be my Yoko Ono	n/a	Yoko / Yoko / 1 0
Brands		
served like Smuckers jam	Smuckers: American jam brand	Smuckers / Smuckers / 1 0
Graffix bong sing along	Graffix: marijuana smoking paraphenalia brand	Graffix / Graffix / 1 0
there's a Hardee's at the next stop	Hardee's: an American mini-mart	Hardee's / Hardee's / 1 0
Locations		
Oakland to Sacktown	Oakland (San Francisco)	Oakland / Oakland / 1 0
uh, Longbeach in tha	Long Beach (Los Angeles)	Longbeach / Long Beach / 1 0
hahaha Frisko, Frisko	Frisko (San Francisco city)	Frisko / San Francisco / 1 0

Table 3.7: *Named entities referred to in songs.*

3.5 Stress patterns

Concurrent with the correction of ebonics instances and typographical errors, entries in the ebonics dictionary were also assigned stress patterns. These patterns were of a format derived from the stressed and syllabified phonetic transcriptions in CELEX, a comprehensive computer database and lexicon listing wordforms (inflected words) along with their lemmas, word class, phonetics and stress information.

In the first instance, stress patterns were transcribed into a dictionary, henceforth called the *CELEX dictionary*, for all regular English words in the lyrics corpus. These patterns used numbers from the set $\{0,1,2\}$ to represent reduced, primary and secondary stress respectively. The new format made stress patterns easier to read compared with the direct CELEX representation, which used single inverted commas to mark points of primary stress (’), double quotes to show points of secondary stress (”), and hyphens to mark syllable boundaries. For example, the word *oceanic* is listed in CELEX as *”@U-SI-’&-nIk*, which would be converted to ‘2 0 1 0’ in the CELEX dictionary.

CELEX often had multiple entries for one word, in which case the most frequent English pronunciation was listed first and the simplest pronunciation appeared to be last in many cases (e.g. entries for *there* in order were ‘1 0 0’ and ‘1’). There were also many instances of entries for multiple words that were identically transcribed but had different pronunciations e.g. *discount* meaning *‘to disregard’* or *‘to reduce in price’*. Since the CELEX database was mined automatically, the blunt heuristic of taking the last entry for each word was applied. It was hypothesised that this was the most probable pronunciation used in singing, but as discussed in Chapter 7, there can be no certainty that this was the correct choice.

Following creation of the CELEX dictionary, stress patterns were assigned to entries in the ebonics dictionary maintaining as much consistency with CELEX pattern encoding as possible. There was no guide to CELEX transcription, however, so this cannot be guaranteed. In addition, in many cases it was not possible to discern the correct stress pattern from the words alone. For example, the pattern for the lyric *“oh-oh-oh-ohh”* might easily be ‘0 1 0 1’ or ‘1 0 1 0’. Many similar examples of such ambiguity were found throughout the data set, including phonetic transcriptions such as *“li-i-i-i-fe”* which might have been translated in any number of ways. When building the ebonics dictionary, it was noted that entries were predominantly of the pattern ‘1 0’ and that part of the role of ebonics appeared to be the conversion of more com-

plex stress patterns in standard English to this simple form. Whether this is a feature of ebonics or symptomatic of the use of words to reinforce a regular beat, it was taken as a rule of thumb and in cases where there was doubt regarding the correct placement of primary stress (e.g. for *"oh-oh-oh-ohh"*), it was placed on the first syllable in every case.

Chapter 4

Feature extraction

This research considers features relating to language form, grammar, sentiment, and repetition of both acoustic (stress patterns) and non-acoustic phrases in lyrics. Further, all non-acoustic features are investigated for lyrics in their raw form, as well as following ebonics correction.

This Chapter outlines the classes of features used, and where relevant, how they were extracted from the data. It also details any decisions made regarding feature extraction, in particular for sentiment analysis. Detailed lists of features are shown in Tables 4.1 and 4.2 separated by whether they are deemed to be *acoustic* or *non-acoustic*. This distinction will continue to be made throughout this document. *Acoustic combined* feature sets are those containing only features from Table 4.1, whilst *non-acoustic combined* feature sets are made up of features taken solely from those presented in Table 4.2. Unless otherwise specified, *combined* feature sets contain a mixture of acoustic and non-acoustic features as described here. Statistics for the mean, standard deviation and minimum and maximum values for each feature using all available data are recorded in Tables 4.3 and 4.4.

4.1 Formatting and form

Prior to feature extraction, lyrics were converted into two formats: a one-token-per-line format for dictionary look-up, including ebonics correction and sentiment analysis, and a sentence format for POS tagging and chunking. The sentence format was produced with the aim of improving accuracy of text pre-processing tools that expected text sentences, and based on the intuition that lyrics with long sentences would differentiate a more narrative, traditional style in comparison with lyrics comprised of short, dis-

Form	
sentNum	number of sentences created by joining lines beginning or ending with conjunctions
avLineLen	average line length (characters)
minLineLen	minimum line length (characters)
maxLineLen	maximum line length (characters)
wordNum	number of words
Language	
contr	number of regular contractions (e.g. don't)
abbrFront	number of frontal abbreviations (e.g. 'cause)
abbrBack	number of dropped endings (e.g. ol') - nb: does not include glottal stops
backup	number of phrases sung by backup singers, as indicated by round brackets ()
oov	number of oov words from regular dictionary
Grammar	
formality	percentage of formal (NN, JJ, IN, DT) versus informal (PRP, RB, UH, VB) word classes
complexity	noun-verb/adjective-adverb ratio (NVAA) or modification index
wvariety	type to token ratio
vbact	number of active verbs
vbmodals	number of modal verbs (i.e. could, should, would)
vbpassives	number of passives verbs (i.e. the girls were watched)
Sentiment	
stpos	number of strongly positive words from a sentiment lexicon lookup
wkpos	number of weakly positive words from a sentiment lexicon lookup
stneg	number of strongly negative words from a sentiment lexicon lookup
wkneg	number of weakly negative words from a sentiment lexicon lookup
totalPos	total number of positive words from a sentiment lexicon lookup
totalNeg	total number of negative words from a sentiment lexicon lookup
Repetition general	
lineRep	number of lines that are repeated in full
lineNum	number of unique (non-repeated) lines
replineRatio	number of unique lines divided by the number of repeated lines
Phrase repetitions (ordered by descending phrase frequency)	
ph1rep	number of repetitions for the most frequently repeated phrase
	...also defined for phrases 2-10
Phrase length (ordered by descending phrase frequency)	
ph1len	length in characters of the most frequently repeated phrase
	...also defined for phrases 2-10
Average repetition distance (ordered by descending phrase frequency)	
ph1distAv	average distance in characters between phrase repetitions
	...also defined for phrases 2-10

Table 4.1: *Non-acoustic features.*

Global repetition	
strPattsNum	number of unique, repeated stress patterns emcompassing more than six stresses
stressOn	total number of emphasised stresses, defined as a one or two
stressTwo	total number of dominant stresses, defined as a two (only has a value where applicable)
stressOff	total number of de-emphasised stresses, defined as a zero
relStress	relative stress, defined as stressOff divided by stressOn
Pattern repetitions (ordered by descending pattern frequency)	
Rstress1rep	number of repetitions for the most frequently stress pattern
	...also defined for phrases 2-10
Pattern length (ordered by descending pattern frequency)	
Rstress1len	length of the most frequently repeated stress pattern
	...also defined for phrases 2-10
Minimum repetition distance (ordered by descending pattern frequency)	
Rstress1distMin	minimum distance in characters between stress pattern repetitions
	...also defined for phrases 2-10
Maximum repetition distance (ordered by descending pattern frequency)	
Rstress1distMax	maximum distance in characters between stress pattern repetitions
	...also defined for phrases 2-10
Pattern repetitions (ordered by ascending minimum separation distance)	
Dstress1rep	number of repetitions for the stress pattern that appears most consecutively
	...also defined for phrases 2-10
Pattern length (ordered by ascending minimum separation distance)	
Dstress1len	length of the stress pattern that appears most consecutively
	...also defined for phrases 2-10
Minimum repetition distance (ordered by ascending minimum separation distance)	
Dstress1distMin	minimum distance in characters between stress pattern repetitions
	...also defined for phrases 2-10
Maximum repetition distance (ordered by ascending minimum separation distance)	
Dstress1distMax	maximum distance in characters between stress pattern repetitions
	...also defined for phrases 2-10

Table 4.2: *Acoustic repetition features.*

#	Name	Min.	Mean	Max.	Std
1	strPattsNum	48	1,100.0	15,000.0	1,200.0
2	stressOn	25	210.0	1,200.0	150.0
3	stressTwo	0	0.0	0.0	0.0
4	stressOff	13	140.0	960.0	100.0
5	relStress	0.33	0.7	1.2	0.1
6	Rstress1rep	11	64.0	290.0	37.0
7	Rstress2rep	7	53.0	240.0	30.0
8	Rstress3rep	7	47.0	200.0	28.0
9	Rstress4rep	7	43.0	190.0	25.0
10	Rstress5rep	6	39.0	180.0	23.0
11	Rstress6rep	6	36.0	170.0	21.0
12	Rstress7rep	5	33.0	160.0	19.0
13	Rstress8rep	5	31.0	140.0	18.0
14	Rstress9rep	5	29.0	140.0	17.0
15	Rstress10rep	5	28.0	130.0	16.0
16	Rstress1len	7	7.1	21.0	0.7
17	Rstress2len	7	7.3	17.0	1.0
18	Rstress3len	7	7.6	21.0	1.2
19	Rstress4len	7	7.8	35.0	1.8
20	Rstress5len	5	7.9	31.0	1.8
21	Rstress6len	7	8.2	27.0	1.9
22	Rstress7len	7	8.3	27.0	2.0
23	Rstress8len	5	8.6	37.0	2.6
24	Rstress9len	5	8.8	33.0	2.7
25	Rstress10len	5	8.8	37.0	2.4
26	Rstress1distMin	0	1.1	63.0	2.9
27	Rstress2distMin	0	1.6	39.0	2.7
28	Rstress3distMin	0	2.0	100.0	5.1
29	Rstress4distMin	0	2.6	89.0	5.8
30	Rstress5distMin	0	2.9	80.0	5.5
31	Rstress6distMin	0	4.0	620.0	20.0
32	Rstress7distMin	0	4.0	150.0	8.7
33	Rstress8distMin	0	4.2	89.0	7.9
34	Rstress9distMin	0	5.0	140.0	9.9
35	Rstress10distMin	0	5.1	150.0	10.0
36	Rstress1distMax	16	71.0	470.0	37.0
37	Rstress2distMax	12	78.0	300.0	38.0
38	Rstress3distMax	9	83.0	420.0	41.0
39	Rstress4distMax	17	90.0	530.0	45.0
40	Rstress5distMax	0	96.0	380.0	47.0
41	Rstress6distMax	1	100.0	620.0	53.0
42	Rstress7distMax	0	110.0	380.0	49.0
43	Rstress8distMax	0	110.0	780.0	60.0
44	Rstress9distMax	0	120.0	370.0	53.0
45	Rstress10distMax	0	120.0	540.0	59.0
46	Dstress1rep	1	11.0	110.0	10.0
47	Dstress2rep	1	19.0	180.0	17.0
48	Dstress3rep	1	24.0	130.0	20.0
49	Dstress4rep	1	26.0	180.0	23.0
50	Dstress5rep	1	26.0	190.0	25.0
51	Dstress6rep	1	24.0	210.0	26.0
52	Dstress7rep	1	22.0	190.0	24.0
53	Dstress8rep	1	22.0	250.0	25.0
54	Dstress9rep	1	22.0	290.0	23.0
55	Dstress10rep	2	21.0	200.0	19.0
56	Dstress1len	1	11.0	53.0	7.5
57	Dstress2len	1	11.0	47.0	6.3
58	Dstress3len	3	11.0	65.0	5.7
59	Dstress4len	1	10.0	45.0	4.2
60	Dstress5len	1	9.6	37.0	3.7
61	Dstress6len	3	9.4	35.0	3.0
62	Dstress7len	3	9.7	35.0	3.5
63	Dstress8len	3	9.6	37.0	3.6
64	Dstress9len	3	9.9	39.0	3.9
65	Dstress10len	5	9.7	33.0	3.4
66	Dstress1distMin	0	0.0	1.0	0.2
67	Dstress2distMin	0	0.2	4.0	0.4
68	Dstress3distMin	0	0.3	6.0	0.5
69	Dstress4distMin	0	0.5	6.0	0.7
70	Dstress5distMin	0	0.7	7.0	0.7
71	Dstress6distMin	0	0.9	8.0	0.8
72	Dstress7distMin	0	1.0	10.0	0.8
73	Dstress8distMin	0	1.2	12.0	1.0
74	Dstress9distMin	0	1.4	12.0	1.2
75	Dstress10distMin	0	1.5	12.0	1.3
76	Dstress1distMax	0	140.0	2,000.0	180.0
77	Dstress2distMax	0	160.0	1,800.0	170.0
78	Dstress3distMax	0	160.0	2,200.0	160.0
79	Dstress4distMax	0	140.0	1,000.0	120.0
80	Dstress5distMax	0	140.0	1,700.0	130.0
81	Dstress6distMax	0	150.0	1,300.0	130.0
82	Dstress7distMax	1	160.0	1,300.0	140.0
83	Dstress8distMax	1	160.0	1,500.0	140.0
84	Dstress9distMax	1	160.0	1,200.0	140.0
85	Dstress10distMax	1	160.0	1,600.0	160.0

Table 4.3: *Data statistics for all acoustic repetition features.*

#	Name	Minimum	Mean	Maximum	Standard deviation
1	contr	0.0	28.0	130.0	15.0
2	abbrFront	0.0	11.0	69.0	9.1
3	abbrBack	0.0	0.6	24.0	1.7
4	backup	0.0	1.4	64.0	4.0
5	oov	0.0	0.9	42.0	3.2
6	sentNum	0.0	7.4	150.0	14.0
7	avLineLen	6.0	31.0	130.0	9.0
8	minLineLen	2.0	14.0	91.0	7.5
9	maxLineLen	10.0	53.0	350.0	17.0
10	wordNum	11.0	210.0	1,100.0	120.0
11	lineNum	3.0	32.0	160.0	17.0
12	stpos	0.0	4.2	99.0	5.3
13	wkpos	0.0	3.9	55.0	4.2
14	stneg	0.0	2.6	33.0	3.2
15	wkneg	0.0	3.5	54.0	4.1
16	totalPos	0.0	8.1	100.0	7.3
17	totalNeg	0.0	6.0	55.0	5.7
18	formality	-120.0	45.0	150.0	20.0
19	complexity	0.8	3.7	85.0	3.2
20	wvariety	1.0	2.3	18.0	0.8
21	vbact	0.0	39.0	220.0	24.0
22	vbmodals	0.0	4.0	42.0	4.1
23	vbpassives	0.0	2.6	42.0	3.4
24	lineRep	0.0	3.9	48.0	3.6
25	replineRatio	1.0	16.0	270.0	22.0
26	ph1rep	1.0	3.2	23.0	2.3
27	ph2rep	1.0	2.2	13.0	1.3
28	ph3rep	1.0	1.8	10.0	1.0
29	ph4rep	0.0	1.6	8.0	0.8
30	ph5rep	NAN (-1)	1.4	7.0	0.7
31	ph6rep	NAN (-1)	1.2	6.0	0.6
32	ph7rep	NAN (-1)	1.2	6.0	0.5
33	ph8rep	NAN (-1)	1.1	5.0	0.4
34	ph9rep	NAN (-1)	1.0	3.0	0.4
35	ph10rep	NAN (-1)	1.0	3.0	0.4
36	ph1len	3.0	30.0	100.0	14.0
37	ph2len	2.0	32.0	120.0	13.0
38	ph3len	3.0	33.0	150.0	14.0
39	ph4len	4.0	32.0	140.0	14.0
40	ph5len	NAN (-1)	33.0	150.0	14.0
41	ph6len	NAN (-1)	33.0	170.0	14.0
42	nph7len	NAN (-1)	33.0	170.0	14.0
43	ph8len	NAN (-1)	33.0	140.0	14.0
44	ph9len	NAN (-1)	32.0	140.0	15.0
45	ph10len	NAN (-1)	32.0	97.0	14.0
46	ph1distAv	0.0	280.0	3,900.0	390.0
47	ph2distAv	0.0	400.0	3,900.0	460.0
48	ph3distAv	0.0	480.0	5,100.0	520.0
49	ph4distAv	0.0	540.0	5,100.0	540.0
50	ph5distAv	NAN (-1)	590.0	5,100.0	570.0
51	ph6distAv	NAN (-1)	620.0	5,100.0	580.0
52	ph7distAv	NAN (-1)	650.0	5,100.0	590.0
53	ph8distAv	NAN (-1)	670.0	5,100.0	600.0
54	ph9distAv	NAN (-1)	690.0	5,100.0	610.0
55	ph10distAv	NAN (-1)	700.0	5,100.0	620.0

Table 4.4: *Data statistics for all non-acoustic features.*

junctive phrases, which would be symptomatic of pop music. Basic features such as average line length and number of words were extracted during formatting, along with features on contractions (part of the *language* feature class).

Tokenisation and other text-processing tasks were achieved using the `txt2onto` pipeline available within the University of Edinburgh Institute of Communicating and Collaborative Systems. `txt2onto` is implemented using `LT-XML2`, which is a collection of generic XML text parsing tools that are well-suited for easy construction of natural language processing (NLP) pipelines. The modules provided as part of `txt2onto` include tokenisation, sentence splitting, POS tagging, noun and verb group chunking, and named entity recognition.

The POS tagger in `txt2onto` is an XML wrapper for the C&C maximum entropy tagger (Curran and Clark, 2007) that offers excellent efficiency and state-of-the-art accuracy on unseen newspaper text (97% per-word accuracy). Although the C&C tagger may be retrained, there was no tagged lyric data available for retraining so as version that had been previously trained on the Penn Treebank was used. This was not ideal as lyrics can differ substantially from standard text, but it was the best option available in the given time. The chunker is rule-based and achieves a level of performance that is comparable with the statistical systems developed for the CoNLL-2000 shared task on chunking. The remaining modules are also rule-based, and whilst not always as accurate as statistical techniques, they are competitive with the state-of-the-art. The main proviso is that output from the chunker can be negatively affected by poor performance of earlier text processing modules. Several of the components of `txt2onto` will soon be distributed as part of the first official release of `LT-TTT2`.

4.1.1 Tokenisation

The one-token-per-line format was produced with the assistance of the `txt2onto` tokeniser, followed by correction of instances in which apostrophes were incorrectly separated from text. For example, where the tokeniser would split *lookin'* into two tokens, *lookin* and ('), this would be corrected to its original form to enable optimal spelling correction to *looking* without leaving residue punctuation. The process of error correction was heuristic due to the varied use of apostrophes, both grammatically correct and incorrect, used in place of quotation marks, to mark slang and for regular abbreviation and contraction. Apostrophes were classed as backward (*lookin'*), forward (*'cos*), backward/forward (*mem'ries*) or grammatical (*say 'I love you'*) based

upon spaces and punctuation in immediately adjacent characters. Due to inconsistent representation of spaces by the tokeniser, there was difficulty with phrases such as *"get in 'cuz we're ready to go"* which was tokenised as *get / in / ' / cuz / we / 're / ready / to / go* and automatically interpretable as either *"getting because..."* or ‘*'get in because..."*. In addition, to correctly handle quotations, which were often preceded by punctuation (e.g.‘*'he said, 'I love you,' to the girl"*), it was determined that an apostrophe was independent if the immediately preceding token was also punctuation.

Some error in re-attachment was inevitable due to inconsistency in the raw data, but where amendment errors were made, most of these were later corrected using the ebonics dictionary. For example, incorrect appendage of the apostrophe in the phrase *lookin ' good* would manifest in the ebonics dictionary as an unknown word *'good* whilst *lookin* would be corrected to *looking* regardless of whether an apostrophe was present. Heuristics were correct in the majority of cases but stumbled upon quotations that started a new line and cases where forward speech marks were represented by two single quotation marks. Nevertheless, the re-attachment procedure reduced the task of hand-correction and facilitated data collection on the number of front and back abbreviations.

4.1.2 Sentence formatting

Prior to sentence construction, existing periods and commas, but not ellipses, were removed from the ends of lyric lines as they were inconsistently transcribed. In addition, all-uppercase words were converted to lowercase since they were predominantly capitalised for emphasis only (e.g. *JUMP! JUMP!*). A sentence was then defined by the heuristic that one lyric line represented a full sentence unless a word from the loosely termed set of 'conjunctives' {*of, but, and, plus, while, meanwhile, meantime, like, with, until*} appeared at the end of the line or beginning of the following line, in which case two or more lines were joined.

Care was taken to ensure that capitalisation was correct for sentence structure and consistent with remaining punctuation, however sentences created in this way were not wholly correct. There were exceptional cases in which conjunctives from the set might be used at the start of a new sentence (*"Like Knieval he weeble words"*), and many more cases in which conjunctives could not be included in the set because they were frequently observed at the start or end, and in the middle, of sentences (*"I want to tell you that \n I love you"* versus *"That's youth! That's all!"*. Nevertheless, an informal

review of the data suggested that the sentence structure created in this manner was sensible for approximately 85% of cases. For this reason, and because the sentence structure provided information in the preferred format for the part-of-speech tagger, the sentence format was retained.

Sentences were counted by identifying all word tokens which were not acronyms or initials or the start of an ellipsis and for which the last character was an end-of-sentence marker from the set $\{!?.\}$. This did not count sentences in which end-of-sentence markers were inside punctuation (‘*‘What cha doin?”*), but it was observed informally that this was a minor issue since such quotes were often already embedded in a sentence.

4.2 Language

Iterating through tokenised lyrics, each word was checked in the CELEX dictionary. For those words that were not found, if they were appended by one of six suffixes identified as being common misspellings, either because they were written in American English (the CELEX dictionary is primarily British English) or due to ebonics transcription (e.g. *brotha* instead of *brother*), automatic correction was attempted using the rules below and the word was re-checked in the CELEX dictionary.

in / in’	ing	runnin becomes running
a / a’ (word length > 2)	er	brotha becomes brother
er	re	center becomes centre
or	our	color becomes colour

Any corrected words and their original transcriptions were written out to file and subsequent checking of the output found no errors. Words that remained OOV were those to which no rule applied or which were not found in the CELEX dictionary on the second attempt at lookup. These words were written to a list of OOV tokens that would become the ebonics dictionary, and output to a file containing the token plus a surrounding five-word window of context to assist correction. At the same time a count of the number of OOV words for each lyric was recorded as a feature.

Parts for back-up singers appeared to be consistently and uniquely marked by round brackets e.g *“We get around (round, get around, we get around)”*. Back-up parts were therefore counted as the number of left round brackets observed in tokenised text.

4.3 Grammar

To extract grammar features, sentence format lyrics were processed using `txt2onto` tagging each word with POS, verb type and chunk information where relevant. Output was parsed and counts determined for verb types (*active, modal* and *passive*) and each of eight pivotal POS categories identified by Heylighen and Dewaele in their 2002 paper on language contextuality. These were the formal word classes of *nouns, adjectives, prepositions* and *articles*, and informal word classes defined as *pronouns, adverbs, interjections* and *verbs*. Where `txt2onto` produced multiple tags for one category, such as *NN, NNS, NNP* and *NNPS* for nouns, these counts were compiled into a single count for the relevant category, in this case *NN*.

POS counts were not used as individual features as in Li and Ogihara (2004), but were combined into the more complex measures of formality and complexity. Data suggest that the level of language formality is dependent on the need for clear understanding. Formal language is often used in situations where unambiguous interpretation is desired, such as in a court of law, for educated discourse or where understanding is difficult to achieve, such as when two parties are separated by time, distance or background. On the other hand, informal language is appropriate for a conversation with a friend and might be expected in lyrics. An empirical measure of formality proposed by Heylighen and Dewaele (2002) was used that defined formality according to the equation:

$$Formality = \frac{(NN + JJ + IN + DT - PRP - VB - RB - UH + 100)}{2}$$

Each word category (*NN*, *JJ* etc.) stands for the frequency of POS tags belonging to that category, represented as a percentage of all words in a given song. The more formal the language, the higher the formality, with average text scoring in the region of 40 to 70. Formality levels for song lyrics achieved a mean score of 45 indicating that lyrics tend towards informality, but more than 100 points variation in both directions indicates substantial variation.

Cognitive complexity is related to the intellectual capabilities of a speaker and has been inferred from use of language, where more highly developed cognitive skills are suggested by a greater number of word tokens, greater type-to-token ratio (*word variety*) and low noun-verb to adjective-adverb ratio (NVAA), or modification index (Powers et al., 1979) (*complexity*). High complexity is indicated by an increased number of qualifiers relative to their referents. Results reported by Powers et al. suggest

that the number of words in a document accounts for 33% of the variance in psychologically assessed cognitive complexity, NVAA accounts for a further 8%, and word variety accounts for 1%. In this research, the first of these was a feature of lyric *form* (wordNum), whilst the latter two were features of *grammar*.

4.4 Sentiment

To see how successful a simplistic approach could be, sentiment features were based on the work of Turney (Turney, 2002), who achieved competitive results using a simple sum of orientations for all words found to have affective meaning in a document. Word orientations were extracted from a lexicon of 8221 words applied by Wilson et al. (2005) in their work on contextual sentiment analysis (Wilson et al., 2005, 2007). The lexicon draws on the first two dimensions of Osgood's theory as discussed in Chapter 2, namely *evaluation* or prior polarity (positive/negative) and *potency* (strong/weak), and includes the adjectives identified by Hatzivassiloglou and McKeown (1997). Adjectives only comprise on average 7.5% of text (Boiy et al., 2007), however, so additional 'opinion words' are included in the list that particularly draw on the POS classes of adjectives, adverbs, verbs and nouns (Bethard et al., 2004)

In order to take account of context, POS labels were used to constrain lookup. This was imperative to distinguish between intended and incidental affect, particularly with regard to the word 'like' that was frequently observed in rap and hip hop in a neutral context referring to named entities, e.g. *"So beat me like Betty Crocker cake mix"*. In this case, *beat* was not in the lexicon but it highlights one of the shortcomings of sentiment analysis: it can be very hard to distinguish context dependent word polarity. For example, *beat* is commonly found in neutral contexts (*Chelsea beat Arsenal 2-1*) although it can also be used in negative context (*The thugs beat the man to death*). No effort was made to account for this kind of contextual misinterpretation in feature extraction apart from constraining POS tags.

In contrast to Turney, since informal review of lyrics suggested that most songs contained a balance of positive and negative words, positive and negative counts were separated into distinct features: *strong positive, weak positive, strong negative, weak negative*, and separate totals were calculated for positive and negative words. In addition, several passes were made through each lyric. These extracted in order, regular inflected words with affective orientation, any words found in the lexicon following ebonics correction, and from the remaining set, if word lemmas were different from

the original transcriptions, word lemmas were also checked for affective orientation. Lemma information was extracted using `txt2onto`. Five 'street' words were also added to the lexicon to avoid obvious error: *f**k, f**ked, f**king, sh*t*, and *gangster*.

A lexicon was chosen over machine learning methods for determining semantic orientation as there was no domain-specific data with which to train a classifier, and a lexicon had the advantage of portability and ease of application. Other publicly available lexicons considered, but not used, included *SentiWordNet* (Esuli and Sebastiani, 2006), the *General Inquirer* (Inquirer, 2007) and the list of adjectives used by Taboada and Grieve (2004) available via the *SentimentAI Yahoo!* group (SentimentAI, 2007).

SentiWordNet is a lexical resource that assigns to each synset of WordNet three sentiment scores, taking into account the sense in which a word is used. For example, the scores for the word 'estimable' for the sense *'deserving of respect or high regard'* are positive=0.75, negative=0.0 and objective=0.25. These scores change for the sense *'may be computed or estimated'*. Whilst useful, it was felt that the *SentiWordNet* approach lacked the necessary precision that would be required for lyric documents that contain on average only between 100 and 350 words. It is not possible to distinguish whether attributes, such as the positive score, arise because the terms in a synset are only partially positive, because they are only used in the positive sense some of the time, or because some terms in the synset are more positive than others. For this reason, a more straightforward lookup was preferred.

The list of adjectives used by Taboada and Grieve (2004) was also deemed less appropriate than a lexicon because it was directed towards research on reviews where the attitude of a speaker towards an object is more important than the emotional state of the speaker themselves. In addition, the author's expanded list, as eventually applied in their research, was not available.

Finally, the *General Inquirer* is a manually compiled resource with 182 categories developed for social-science content-analysis research. It is not directed towards natural language processing (NLP) objectives, and due to its complexity, the user must specify what contrasts are to be statistically compared. Considering the task at hand, this process appeared unnecessarily detailed and onerous.

4.5 Repetition

The SRI Language Modeling Toolkit (SRILM) ngram-count routine was one of two tools used to extract information for repetition features. SRILM is a collection of C++ libraries, executable programmes and helper scripts made publicly available in 1999. Although its more complex statistical language modelling capabilities are still under development (Stolcke, 2002), its ngram-count routine is a powerful and stable tool providing counts for all possible ngrams of variable length. For the purpose of extracting lyric repetition, maximum ngram length was set to ensure ngrams were delimited by line breaks. In this way, SRILM output was used to count the number of times every line in a lyric was repeated in full, as well as record repetition for any selected phrases. Information about line repetition was the basis for features including the number of unique lines and the ratio of unique to repeated lines.

The `txt2onto` chunker was used to identify phrases that might be pertinent to a study of repetition. Although it was initially intended that verb phrases and noun phrases would form separate groups, since they overlapped in the chunker output it was decided only to distinguish between words that were *in phrase* and *not in phrase*. This resulted in some phrases that were verb groups, some that were noun groups, and some that were a combination of the two. Phrases that contained contracted words that had been split in two as a pre-processing step for the chunker, had these tokens recombined in order to ensure matching of the chunker-identified phrases with those output from the SRILM routine that used non-tokenised text.

Following phrase selection and cleaning, count information for each phrase was extracted from SRILM output, and a list of all potential phrases and full lyric lines compiled with count information. These entries were then cleaned, first by removing all one-word phrases, since these were likely to occur with disproportionate frequency, and then by removing any phrases that were subsets of longer phrases. This list was then sorted by count frequency, and the top ten phrases used to output features on phrase length, number of phrase repetitions and separation distance (defined as the number of characters between two manifestations of a phrase). If a phrase only appeared once, the length and number of repetitions were assigned to be -1 and the separation distance was the length of the song in characters.

Identifying repeated phrases in this manner appeared to extract accurate and meaningful information about the degree of repetitiveness of lyrics. For example, the Kottonmouth Kings, “Bump”, a rap song, only has one identified phrase, *“come on, come*

on", which is repeated twice. In comparison,"Knowing me, knowing you" by Abba has multiple repeated phrases, all of which appear in the chorus. These phrases and their counts are:

6	*"knowing me, knowing you"*
2	*"there is nothing we can do"*
2	*"we just have to face it"*
2	*"this time we're through"*
2	*"breaking up is never easy i know"*
2	*"but i have to go"*
2	*"it's the best i can do"*

4.6 Stress Patterns

Stress patterns for each word were listed in the CELEX and ebonics dictionaries, with three types of stress marker - primary, secondary and reduced, equating to '1', '2' and '0' respectively as outlined in Chapter 3. It was not known whether this form of stress marking would result in patterns that were too sparse to extract meaningful 'phrases', so an alternate form of stress encoding was also introduced that replaced every instance of '2' with a '1'. These two formats were called *numeric* and *binary* respectively. In addition, since the pattern of punctuation and pausing was thought to be an important aspect of stress, two additional formats were introduced for both the numeric and binary branches. The first inserted a dash (-) in the place of every comma (*comma* version), and the second inserted a dash for every comma, and an additional dash for every other type of punctuation marking a pause from the set {?!.:;-} (*allpause* version). Table 4.5 shows how this would translate into a continuous stress pattern for a lyric from the Faithless song "Salva mea".

Songs in their poetic form were transcribed into each of these six formats: *binary, bcomma, ballpause, numeric, ncomma* and *nallpause*. For words with regular contractions, such as the *n't* in *isn't* or the *'s* in *it's*, the contractions were automatically assigned a reduced stress if they were not included in the CELEX dictionary. Following conversion into stress format, lyrics were processed using the SRILM ngram-count routine described above. In addition to extracting the counts for all full lines, all 'phrases', defined after experimentation to be a series of seven or more stress markers, were extracted with their counts. The minimum length on these stress 'phrases'

Yeah, can you feel it?	
Binary	1 1 0 0 0 1 0 1 0 1 1 1
Bcomma	1 - 1 0 0 0 1 0 1 0 1 1 1
Ballpause	1 - 1 0 0 0 1 0 1 0 1 1 1 - -
Numeric	1 1 0 0 0 2 0 1 0 2 1 1
Ncomma	1 - 1 0 0 0 2 0 1 0 2 1 1
Nallpause	1 - 1 0 0 0 2 0 1 0 2 1 1 - -

Table 4.5: *Example ebonics transcription from "Salva mea" by Faithless.*

was defined in order to avert very short sequences (e.g. '1 0') from drowning true repetition.

The stress patterns and stress encoded lines were ordered for each format type using two distinct methods. The first method ordered by count frequency, as used to order linguistic phrases (*R* type, for the number of repetitions). The second method ordered by minimum separation distance (*D* type, for distance) on the intuition that stress sequences that appeared back-to-back were more important than those that simple appeared regularly e.g. '0 0 1 0 0 1 1 0 0 1 0 0 1 1' would constitute a stress pattern '0 0 1 0 0 1 1' repeated back-to-back with a minimum separation distance of zero. Since the combined pattern might equally be recorded as one instance of a pattern with twice the length, for example, if it represented a full lyric line repeated three times throughout a song, the length of the stress pattern was also taken into consideration. The features extracted for stress patterns determined by each ordering method were therefore the count, length and the minimum and maximum separation distances.

Chapter 5

Methodology

5.1 Self-organising maps

The publicly available SOM Toolbox, developed in the Neural Networks Research Centre of the Helsinki University of Technology, was used for all data visualisations (Vesanto et al., 1999). The SOM Toolbox is a free function library for MATLAB 5 implementing the SOM algorithm and various auxiliary scripts in a modular fashion that makes tailoring to specific tasks easy and efficient. MATLAB has excellent support for graphics and is well suited to fast prototyping and customisation, especially in comparison with SOM_PAK, which is implemented in C.

The SOM grid size determines the number of input units, which are represented as hexangonal nodes in the SOM map. There is no rule regarding the number of input units, however it is important to make a selection that is large enough to reveal patterns and clusters, but not so large that node assignment becomes random. Figure 5.1 shows the distancing that can occur when there are too many input nodes for the available data. It becomes equally probable that a data point is assigned to a neighbouring node with the same, or very similar, prototype vector as clusters are exploded. In this case, two points may be neighbours in one iteration of the SOM and separated on another iteration, even when the SOM is discovering the same clustering pattern.

In all visualisations, SOMs were trained with 20x12 units representing approximately one unit for every ten data points for the full data set (2392 songs). Linear initialisation was used, with a hexagonal grid, 'sheet' topology, Gaussian neighbourhoods and Euclidean distances. The SOM grid size was set after comparing the distance matrices generated using the 20x12 grid for gold standard, IS and language features sets, with the distance matrices resulting from the 17x14 grid automatically calculated for

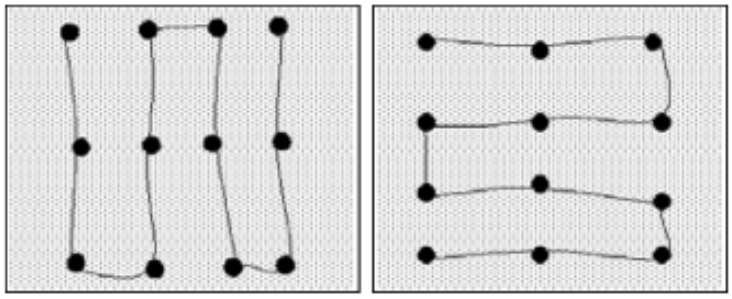


Figure 5.1: *Two equally likely outcomes of node assignment in SOMs with a doubled grid size (here, one to two dimensions). Two data points may be assigned neighbouring units with one initialisation and non-neighbouring units with a second, different initialisation. Figure reproduced from Rousset et al. (2006).*

the data using a built-in toolbox function. The 20x12 grid presented a similar number of nodes but was more easily interpretable and was thus preferred. Whilst this was appropriate for the number of data points in the full data set, a reduction in the data set, as for the *ebonics on* and acoustic SOMs, would normally correspond with a comparable reduction in the grid size. Since this would compact the grid to such an extent that clusters might no longer be visible, however, the grid was maintained at the same size as used for the full data corpus.

As outlined in Chapter 2, ideally ten or more randomly initialised maps are trained and reviewed, and clusters are selected that are apparent in all, or most, of the final SOMs. Alternatively, one map is selected that best represents the average clustering across all maps. When there are many maps to review, however, this is a time-consuming task and it becomes necessary to accept one map as the final result (Samsonova et al., 2006). Since this research required many visualisations, whilst not ideal, this was the approach taken. Linear initialisation was selected as a means of systematically analysing the data in a limited amount of time. Two sequential phases of batch training followed initialisation, for rough and fine-tuning respectively. Batch training was preferred to online training as the data set was a manageable size that made this possible. Learning rates for training were automatically calculated by the SOM toolbox software, as were the number of training epochs and parameters for the neighbourhood function.

5.2 Model search and selection

5.2.1 Gold standard

Song vectors for gold standard data were created as described in Chapter 2 and reduced to 25 dimensions using principle component analysis (PCA). The number of dimensions in the lower dimensional space was determined by reviewing the reconstruction error on the matrix created by the song co-occurrence counts, which increased when either 24 or 26 eigenvalues were selected.

Reducing the dimensionality of the gold standard data was necessary to overcome data sparsity. It also made visualisation and storage of the matrix easier, and tended to reduce overfitting when the matrix was analysed using SOMs that have a tendency to overfit. On the other hand, there remains the possibility that any reduction in dimensionality may have thrown away information that was pertinent to clustering, and this should be borne in mind when reviewing the results of both the gold standard and IS models.

5.2.2 Information Space

When analysing newspaper text or similar documents, a common choice is to sue in the region of one thousand content-bearing (indexing) words, and for these words, calculate co-occurrence counts for every document term. The document vectors constructed from these counts will then be projected into a lower dimensional space, probably between 100 and 300 dimensions (Infomap, 2007a). Content-bearing words from prose are often taken to be the one thousand most frequent words after stop-word removal, but for song lyrics, the most likely, or optimal, number of indexing words was unknown since the data was considerably different from standard prose.

Using the Text to Matrix Generator (TMG) (Zeimpekis and Gallopoulos, 2006) Indexing module, the number of indexing words was automatically calculated whilst various constraints were placed on valid document terms. TMG is a MATLAB toolbox for data mining and information retrieval that takes advantage of MATLAB's sparse data infrastructure whilst offering complete compatibility with the SOM toolbox used for visual analysis. Infomap (Infomap, 2007b), which uses a variant of LSA, was considered as an alternative for this portion of data analysis, however it did not offer the same degree of customisation as TMG or enable output of the required document vector matrix.

Customised settings for indexing lyrics were as follows. Minimum term length was three characters and maximum was 30, whilst minimum local and global frequencies were both one (maximum infinity). Both term and document normalisation were applied since Tang et al. (2004) found that this significantly improves both precision and recall, with a 76% overall improvement in performance. Normalised log was used for local term weighting and entropy was used for global term weighting. Normalised log was selected for local weighting as it is suited to repetitive documents; it gives lower weights than regular log and distinguishes terms that appear once from terms that appear many times (not offered by binary weighting) without placing too much importance on repeated terms, as with frequency weighting (Chisholm and Kolda, 1999). Global weighting draws on the idea that the less frequently a term appears in a document collection the more discriminating it is. Entropy weighting was selected as this gives a higher weight to terms that appear fewer times in a small number of documents, as would likely be the case for lyrics, whereas the popular alternative Inverse Document Frequency (IDF) simply awards higher weight to terms appearing in a small number of documents (Chisholm and Kolda, 1999). Finally, stemming was applied using the Porter stemming algorithm (Porter, 1980) to help deal with sparsity arising from very short documents.

Using these settings, lyrics were indexed using two stop-lists: a list of 526 commonly occurring words, including those expected to be found in lyrics (e.g. *oh, la, baby* and *love*), and a shortened stop-list containing only 170 very high frequency words. Visualisation of the distance matrices resulting from SOM analysis of the subsequent IS output clearly showed the longer stop-list resulted in superior performance. This was consistent with the reported application of stop-words by Logan et al. in their 2004 paper on LSA in lyric analysis, where they noted improved performance with the addition of 263 'lyric-specific' stop-words (no figure was given for the length of the total list).

Dimensionality reduction was investigated using SVD and PCA, and SOM analysis was applied to the output. The resulting maps for data reduced to output dimensions from the set $\{50, 75, 100, 125, 150\}$ were visually assessed and measured for quantization (q) and topographic error (t). Although the outputs were very similar (e.g. PCA 100: q=6.024, t=0.035 and SVD 100: q=6.003, t=0.034), it appeared visually that PCA 75 offered the best performance and map error confirmed this observation. PCA 75 was therefore selected for further analysis with visualisations for dimensions in the set $\{55, 65, 75, 85, 95\}$. No distinct pattern of change was observed and PCA 75 was

Number of documents	2,392
Number of terms	47,523
Average number of terms per document (before the normalization)	30.462
Average number of indexing terms per document	22.483
Sparsity	0.04729%

Table 5.1: *Statistics for the Information Space PCA 75 model.*

selected as the final model. This meant the technique being applied was Information Space, rather than LSA (see Chapter 2). Statistics for the final IS model are provided in Table 5.1.

5.2.3 Lyric histories

To facilitate exploration of different feature combinations, each lyric was associated with a unique 'history' file that held a short version of the artist and song title for SOM labelling, along with a series of feature-value pairs for every feature investigated. These pairs took the format 'feature=value' making them easy to identify and extract. Values were numerical and positive in every case except where they indicated an undefined number, in which case their value was -1.

For each SOM investigated, values of the desired features were extracted from the lyric history files and compiled into a matrix of song vectors. All attributes were then normalised to mean 0, standard deviation 1, so that one feature would not dominate over another due to differences in scale. These matrices were then input to SOMs and the results visualised.

5.2.4 Feature editing

5.2.5 Phase 1: Feature editing

For the first training of an SOM, features were divided into classes of ten or fewer features, input to the SOM and visualised with histograms and pairwise scatterplots to reveal highly correlated variables and data distributions. Feature classes are shown in Tables 4.1 and 4.2 and an example of scatterplots and histograms for the five features examined in connection with global stress patterns are shown in Figure 5.2. The scatterplots clearly reveal that there is a strong correlation between *stressOn* and *stressOff*

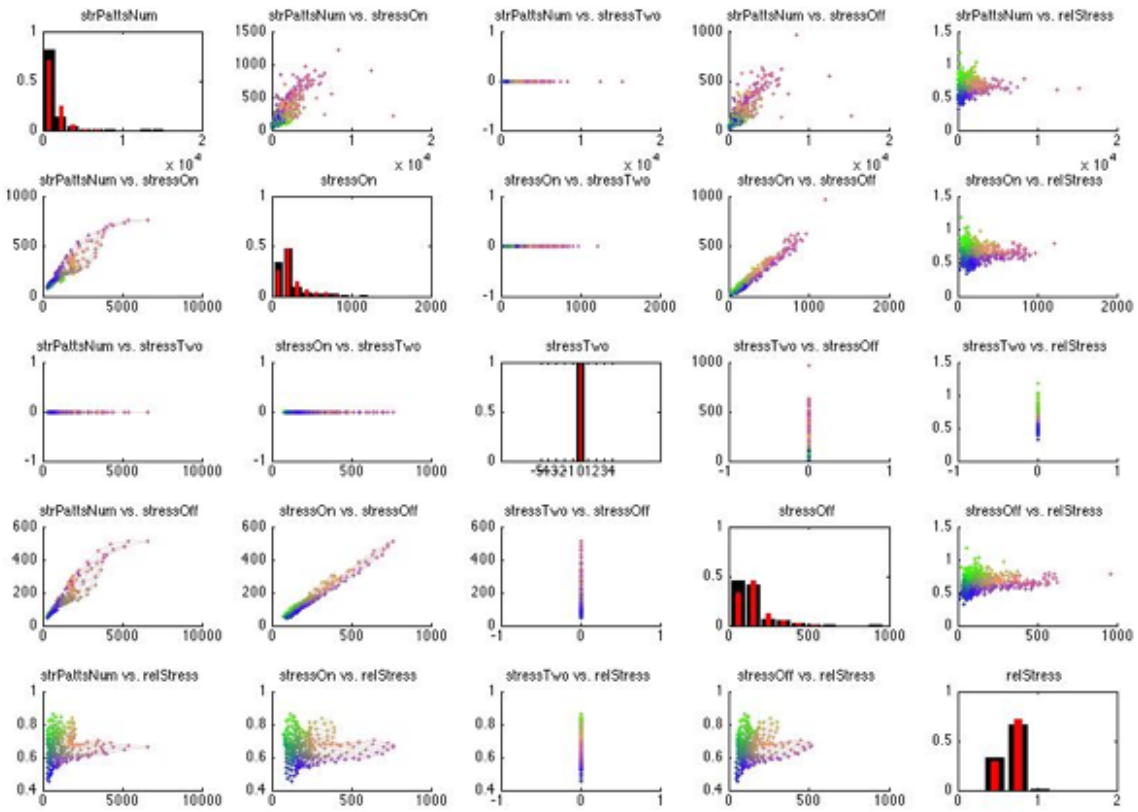


Figure 5.2: *Pairwise scatterplots and histograms for five attributes considered regarding global stress patterns. On the basis of these plots only relStress and stressOff were retained for further analysis. The histograms show distribution of values, black for the data set and red for the SOM prototype vectors.*

and reasonable correlation between *stressOn* and *strPattsNum*. *StressTwo* is not present since only 'binary' stress pattern data are shown. Highly correlated variables can cause neural network weights to become extremely large and result in overfitting, so where they were found, one or more features were discarded to avoid multicolinearity. For example, on the basis of these scatterplots, only *relStress* and *stressOff* were retained for further analysis.

Two dimensional plots were chosen for scatterplot analysis as it can be difficult to see the effect of a particular feature in multi-dimensional visualisations. Data points were encoded in colour by laying a colour plane over the matrix describing the feature values. In this way, when several pairwise combinations produced very similar results it was possible to observe which pair provided the best data separation; this was the pair that best clustered and separated the coloured data.

The first columns for each class in Tables 5.2 and 5.3 show the results of preliminary scatterplot analysis. Initially all features are present (A/C), and in the second

column only retained features are marked in black (B/D).

In Table 5.3, blue bars refer to features selected for *binary* stress patterns, whilst yellow bars refer to *ncomma* patterns. These pattern formats were selected as representative of the six stress patterns constructed from text following all scatterplot analyses since they appeared to provided the best colour separation of the data. Generally, there was not much difference observed between plots for the *comma* and *allpause* patterns, possibly because line breaks served as lyrical proxies for most end of sentence punctuation. Although line breaks marked the end of sentences in many instances, this was not captured by *comma* and *allpause* encoding since there was no way to distinguish these cases from mid-sentence returns. Some distinction could be made between binary and numeric cases, however, probably due to an increased chance of stress pattern repetition when there were fewer variables. To capture these findings whilst narrowing the field for further analysis, binary and ncomma were selected as the simple and complex cases of stress pattern incidence.

5.2.6 Phase 2: Fine-tuning

Combining the features selected in the first phase of analysis produced three classes of *non-acoustic* features and two classes of *acoustic* features. These were the *language, sentiment* and *repetition* classes for non-acoustic features and the *Rstress* and *Dstress* classes for acoustic, or stress pattern, features. Global stress patterns were combined with Rstress features in the first instance, and later with Dstress features (shown in Phase 3).

The retained features for these classes were used to construct new matrices that were input to SOMs. Once again, data were normalised to mean 0 standard deviation 1, then visualised using distance matrices, three-dimensional clusters and component analysis:

- **Distance matrices:** Distance matrices map distances between neighbouring units of an SOM, so that clusters are suggested by nodes with small distances between them and long distances to other nodes. Distance matrices are shown in full-sprectrum colour, where royal blue indicates that nodes that are close together and red means they are far apart. In Figure 5.3, distance matrices are shown in two dimensions to clearly illustrate the difference in clustering between ordering stress patterns by frequency and ordering by minimum separation distance.

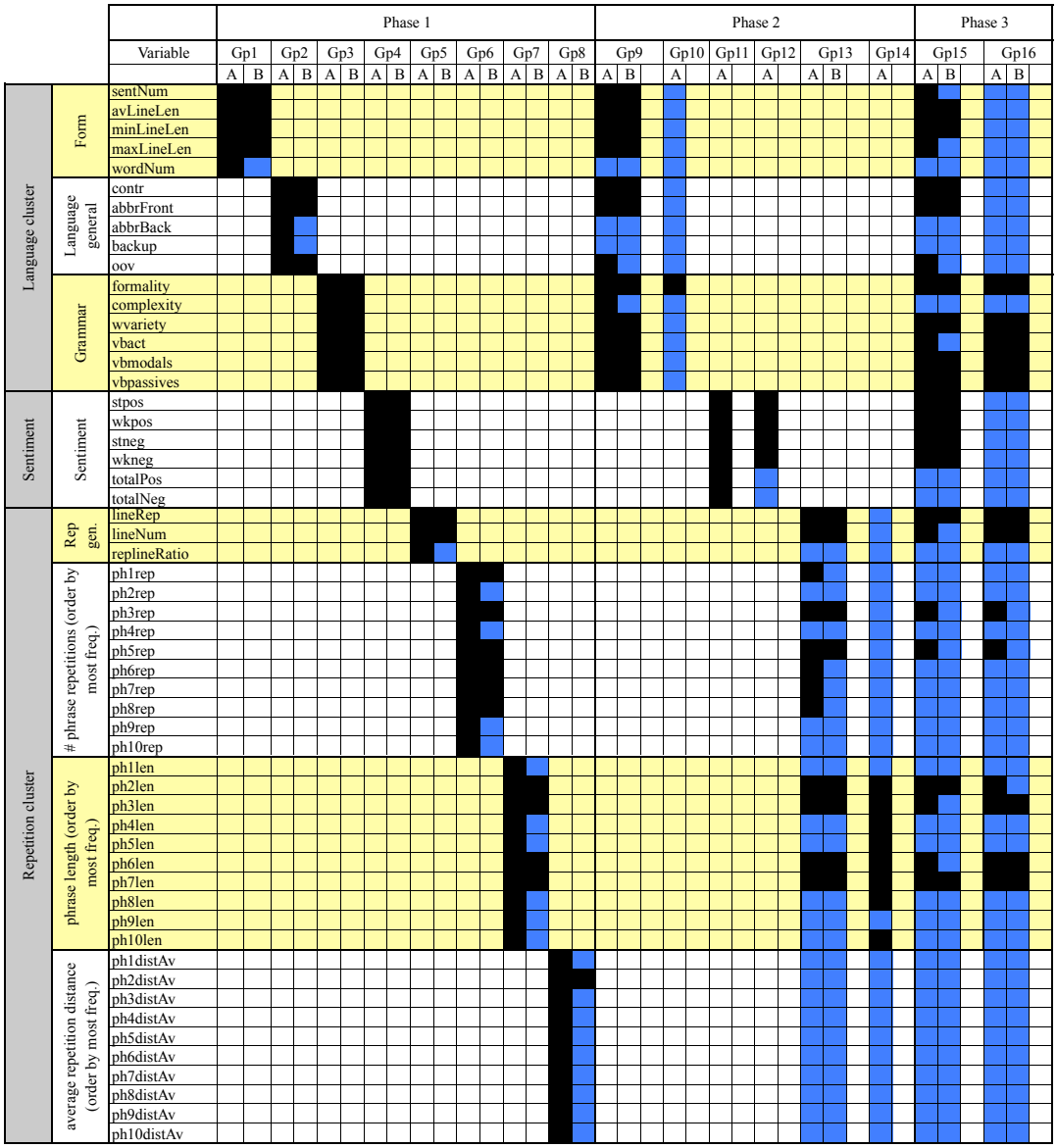


Table 5.2: *Non-acoustic song feature sets.*

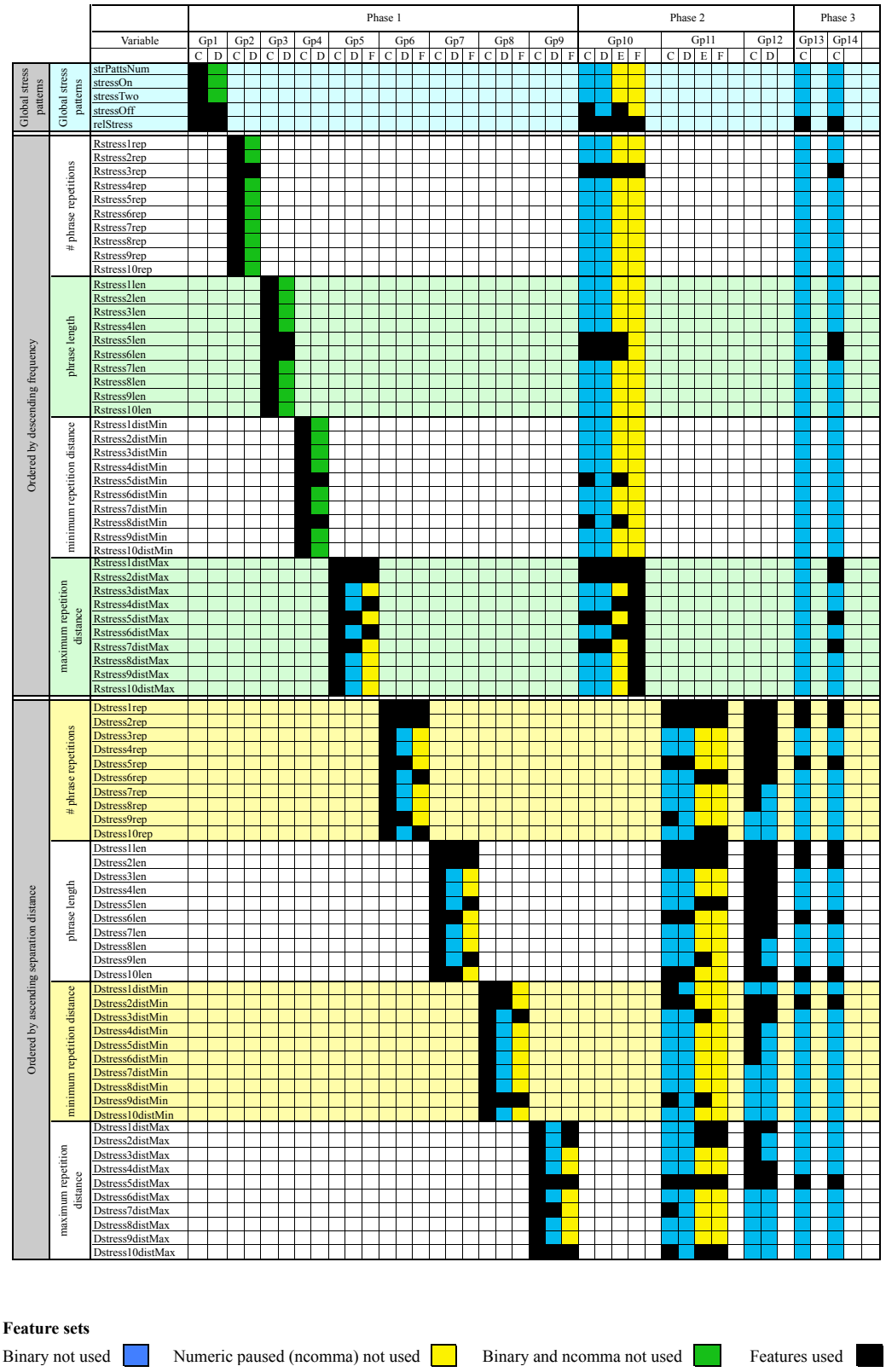


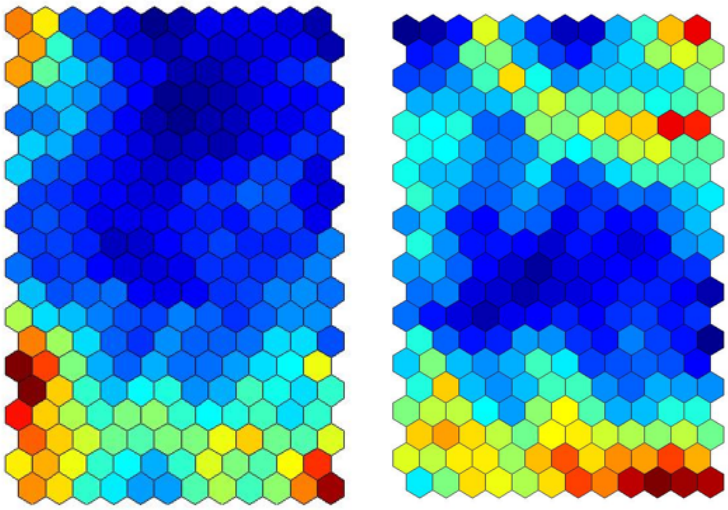
Table 5.3: *Acoustic song feature sets.*

- **3D clusters:** For three-dimensional (3D) visualisations, prototype vectors of SOM nodes were projected onto the two greatest eigenvectors to give an approximate representation of clusters. In these figures, nodes are dots and lines represent connections between neighbouring nodes. Figure 6.1 gives an example of how this appears for the gold standard and IS data. Short lines indicate that two nodes are close together, and long lines show that they are far apart. This enables clusters to be easily spotted by the bunching of dots in a particular area. 3D visualisations can be misleading, however, due to the choice of perspective taken on the graph and the fact that pertinent information may be thrown away with minor eigenvalues. Figure 5.5 gives an example of this, where it is hard to see whether Rstress or Dstress clustering is preferred. To help overcome this limitation, projections were made onto the three greatest eigenvectors and rotated in three dimensions to gain a better understanding of the characteristics of the map.

- **Component analysis:** Component plots reveal the kinds of values SOM prototype vectors have for different features, and always show each map unit in the same place. In this way they facilitate easy understanding of the data. Figure 5.4 shows the component planes for the same five features used in the scatterplot example, converted into three colour maps for easy interpretation. Looking at the pattern of values for each feature it is once again easy to see the correlations. It is also apparent that for three features, most of data have similar values (are mainstream) whilst the rest are exceptions to the rule. This is a tendency observed across all feature classes.

These visualisations were used for fine-tuning of features would be selected for further analysis in combination with other feature classes. The first bar marked for each class in the Phase 2 sector of Tables 5.2 and 5.3, shows the results of secondary analysis. Initially all features selected in the first Phase are present (A/C), and in the second column (B/D) certain features have been removed. This is because component analysis indicated that combination of classes following Phase 1 analysis brought together several correlated variables, one or more of which needed to be removed.

In some cases, one or several features seemed particularly interesting in component analysis, and these were selected for a further round of investigation using the same methodology described above. The features input to SOMs in these further investigations are shown in the second bar of marked features (A) for each class. Distance



(a) Rstress, 1037 songs (b) Dstress, 1037 songs

Figure 5.3: *Acoustic repetition features: distance matrices on a 2-dimensional grid.*

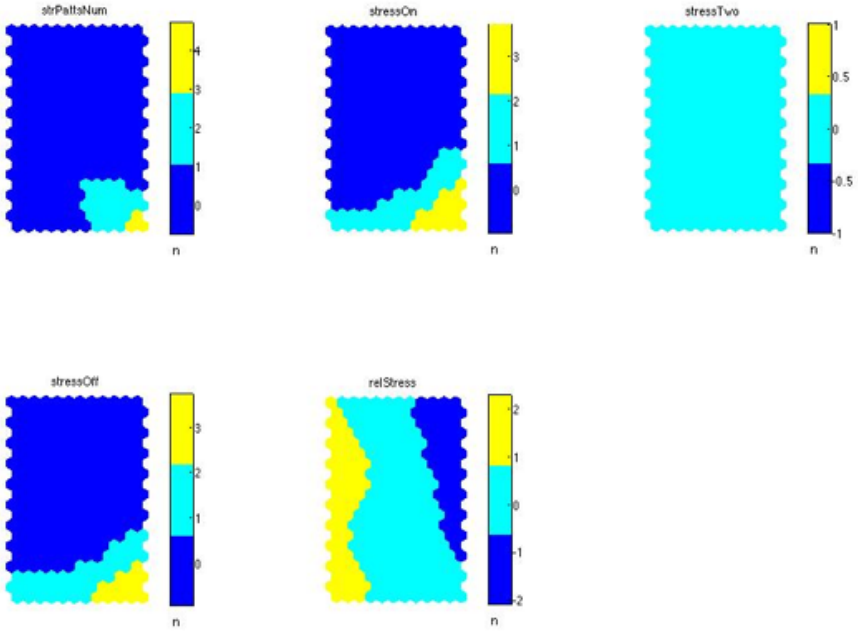
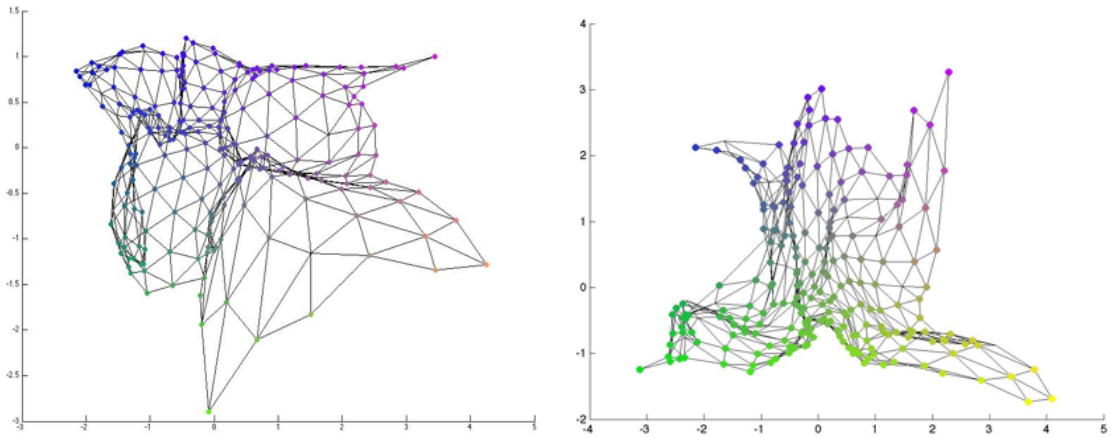


Figure 5.4: *Component analysis such as those investigated in phase 2, shown here for global stress patterns. There is visible redundancy in attributes stressOn, stressOff and some redundancy with strPattsNum.*



(a) Rstress, 1037 songs (b) Dstress, 1037 songs

Figure 5.5: *Acoustic repetition features: clusters shown by projection onto top two principle components.*

matrices and three dimensional clustering were reviewed for all SOMs to determine the quality of clustering. For the language, repetition and Dstress classes, visualisations showed that these groups were inferior to the first Phase 2 combinations tried. For the sentiment class, it became clear there was a correlation between *totalPos* and *wkpos*, and between *totalNeg* and *wkneg*, since these features pulled excessively at the edges of the map making it difficult to observe the clustering caused by other variables. Since the weak sentiment features showed better clustering, *totalPos* and *totalNeg* were removed.

The distance matrices also served a second function, which was to identify outliers for investigation that presented as singular dark red nodes surrounded by less extreme values. Outliers can be a problem for neural networks since the network expends an unjustified amount of effort trying to account for them. For this reason, each potential outlier was checked by finding songs with maximum and minimum values for the variable of interest, and reviewing the lyrics for these songs. Outliers were discarded when there were justifiable reasons for their removal. For example,"Euroboy" by the Pet Shop Boys contained a single-line lyric repeated many times with some background noises (*"Viva viva"*), and "Warszawa" by David Bowie was foreign language, so they were both removed. In contrast, whilst 'Do you want more" by The Roots contained a lot of contractions and a highly repetitive second verse, it was still a genuine lyric and thus was retained. In addition, three cases of lyrics which had been transcribed predominantly in prose form were corrected, and the data for these songs was re-run.

Feature combinations producing the best SOM for each class, as determined by

review of the distance matrices, component analyses and 3D clustering, went on to be combined into more complex models and analysed using test song pairs (see below). All non-acoustic feature sets were tested with *ebonics off* (full and 1037 song data sets) and *ebonics on* (1037 songs) to compare the impact of data cleaning and reduced data size. These comparisons were not possible for acoustic features since stress pattern transcription was part of the ebonics dictionary and therefore could only be applied to cleaned data. In addition, *binary* stress patterns appeared to result in better clustering than *ncomma* patterns and were selected as the final model for comparison.

5.2.7 Phase 3: Combinations

5.2.8 Results visualisations

Combining the features selected in the second phase of analysis produced two classes of features: *non-acoustic combined* and *acoustic combined*. The retained features for these classes were used to construct new matrices that were input to SOMs in the same way they were during Phase 2. Correlations were removed following visualisation of distance matrices, three-dimensional clusters and component analysis, resulting in the combinations reported in the Phase 3 sector of Tables 5.2 and 5.3. Initially all features selected in the second Phase are present (A/C), and in the second column (B/D) certain features have been removed. Once again, a potentially interesting combination of non-acoustic features was subject to a further round of investigation using the same methodology. The features input to this SOM are shown in the second bar of marked features (A/B) in Table 5.2.

Further combinations are shown in Table 5.4, including a *combined* model encompassing both the full *non-acoustic combined* and the *acoustic combined* feature sets. In addition, SOM input was investigated with a model that concatenated *IS* and *combined* features. It was not clear from earlier visualisations whether Dstress or acoustic combined features offered better clustering, so IS combinations were investigated using both feature sets and the same methodology described above. The four combinations including IS data are shown in Table 5.4, all of which were investigated using both *ebonics off* and *ebonics on*.

Further to visualisations outline above, the following were used for reporting results (Chapter 6).

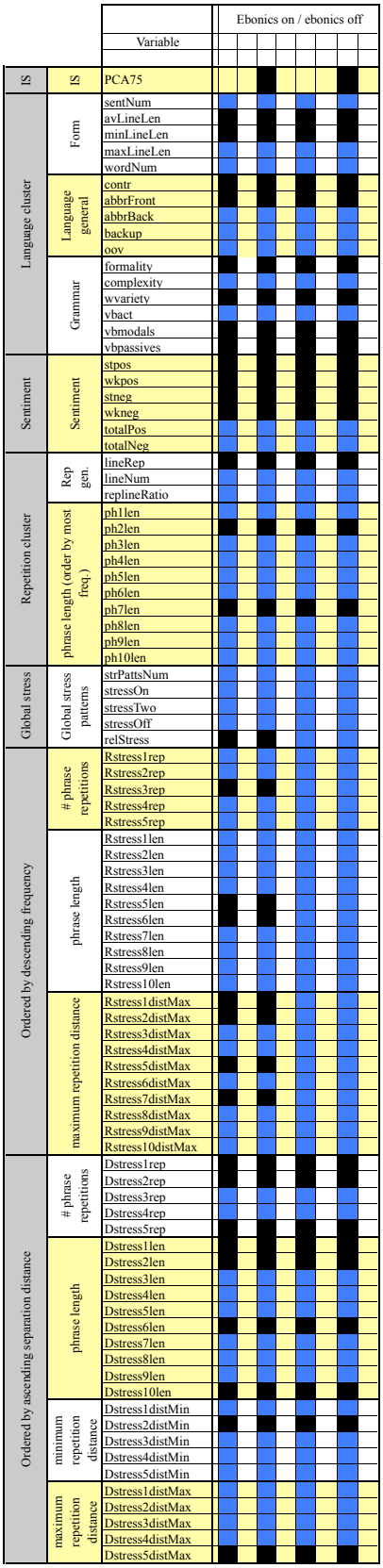


Table 5.4: *Combination features.*

- **Distance matrices:** Distance matrices were combined with a topological representation in which valleys are areas of close proximity and peaks indicate that nodes have a much greater separation distance. Coloured spheres represent the location of nodes associated with test songs in colours corresponding to the legend in Figure 5.6. These nodes are not always fully visible due to the topology of the map, are are best interpreted in combination with the hit maps.

- **U-matrices and hit maps:** A commonly used relative of the distance matrix is the U-matrix, which shows the median distance from each map unit to its neighbours, as well as the distance between neighbouring units. The U-matrices are presented in black and white as background for 'hit maps'. Hit maps reveal the number of data points associated with each node in the SOM; the more data points associated with a node, the bigger the white hexagon that appears in the visualisation. For analysis, hit maps are used to show the position of test songs through colour coding of the hits to reflect the songs they represent. In cases where several test songs from different pairs are clustered on the same node, or many non-test songs are clustered on the same node as a test song, the coloured hits may not all be visible. In these cases, where a written explanation could not be easily provided, additional visualisations for the hidden nodes are shown.

5.3 Test song pairs

A small set of eight test pairs was used to gain insight into the SOMs selected as representative of each feature class and combination. This type of approach was applied successfully by Whitman and Smaragdis in his 2002 paper on the application of audio and textual web data to music classification. Whitman and Smaragdis used a small test set known as the Minnowmatch music testbed, comprising 25 songs from 25 artists, five from each of five music styles. This set covers a range of genres not reflected in the current corpus, hence it was not appropriate for this research. Instead, test pairs were hand-crafted by the author based on an informal review of song lyrics for which ebonics correction was available. The selection process aimed to produce a sample of each major, identifiable lyric style observed. In addition, it was desired that all similarities within a pair were intentionally selected, and discrepancies that might invalidate results were kept to a minimum. The songs that were selected by this process are shown in Figure 5.6, along with the colour codes assigned to facilitate analysis.

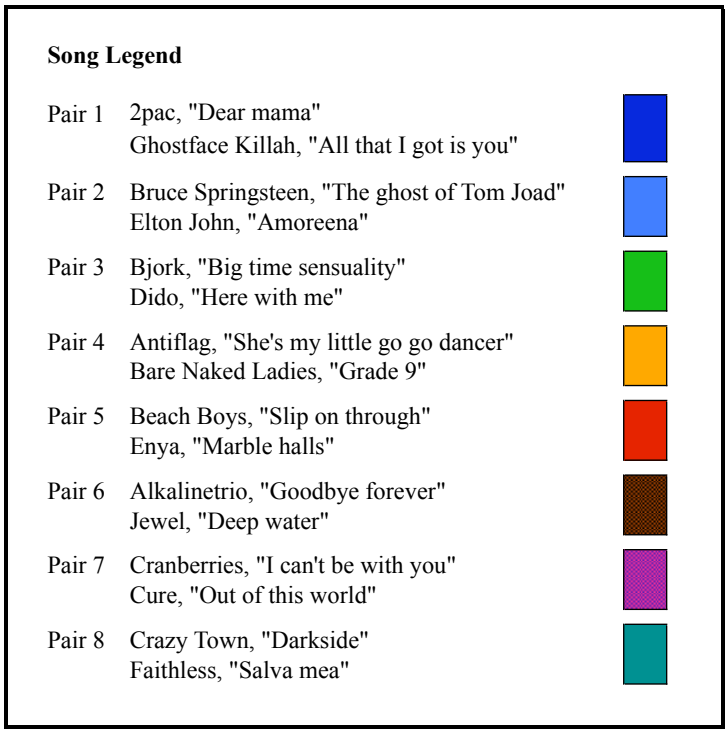


Figure 5.6: *Lyrically similar song pairs used in error analysis.*

Reasons for the selection of each pair were similarities determined by judge 1 (the author), which were purely lyrics based. Pairs were subsequently checked by a colleague and where these opinions differed from those of the author, they are recorded as differences perceived by judge 2. In addition, judgments of human music experts were obtained for each song from the *All Music Guide* in the form of genre and style information (Table 5.5). All three opinions contributed to the final decision regarding whether it was desirable for the lyrics selected to be paired by SOMs. This decision, along with a summary of all judgments, is provided in Table 5.6.

5.4 Clustering analysis

Initially, it was intended to evaluate SOMs by applying the Davies-Bouldin Index (DBI) to assess validity of clustering produced using a standard K-Means algorithm over the map prototype vectors. Whilst K-means is not a very sophisticated clustering algorithm, the data points were previously clustered by the SOM, and the algorithm was included with the SOM toolbox, so it would be efficient to apply.

Despite initial expectations that it would be possible to measure clustering valid-

	Artist	Genre	Styles	Track title	Track audio description
Pair 1	2pac	Rap	Gangsta rap West coast rap G-funk Hardcore rap	Dear Mama	Rap over a sparse, slow electronic melody with a loud snare drum
	Ghostface Killah	Rap	Hip-hop East coast rap Harcore rap	All that I got is you	Hip-hop love song effect, heavily backed by strings and piano
Pair 2	Bruce Springsteen	Rock	Rock & roll Singer/songwriter Pop/rock Album rock Heartland rock	The ghost of Tom Joad	Harmonica and folk guitar
	Elton John	Rock	Rock & roll Singer/songwriter Adult contemporary Soft rock Pop/rock Album rock	Amoreena	Mainstream pop with piano: prototypical Elton John
Pair 3	Bjork	Rock	Experimental Rock Electronica Experimental Alternative pop/rock Alternative dance Trip-hop Club/dance	Big time sensuality	Trance music overlaid with a slightly reverberated, strong vocal
	Dido	Rock	Rock Adult alternative Alternative s/songwriter Contemporary s/songwriter	Here with me	Pop music with electronic dance elements and a rock drum beat
Pair 4	Anti-flag	Rock	Punk revival Hardcore punk Oi!	She's my little go go dancer	Youth punk pop
	Barenaked Ladies	Rock	Alternative pop/ rock Adult alternative pop/rock Post-grunge	Grade 9	Comedy pop, ska beat
Pair 5	Beach Boys	Rock	Pop Surf Pop/rock Sunshine pop Rock & roll Psychedelic pop Psychedelic	Slip on through	Pop with less harmony than you might expect from the Beach Boys and a slightly funky sound
	Enya	New Age	Celtic Adult alternative Ethnic fusion World Contemporary instrumental Adult alternative pop/rock Contemporary Celtic Celtic new age	Marble halls	Ethereal, almost an a capella mediaeval hymn
Pair 6	Alkaline Trio	Rock	Punk revival Punk-pop Emo	Goodbye Forever	Youth indie punk (Emo)
	Jewel	Rock	Singer/songwriter Adult alternative pop/rock Pop/rock Contemporary s/songwriter	Deep water	Acoustic guitar and soulful vocals with a touch of country
Pair 7	Cranberries	Rock	Alternative pop/ rock Adult alternative pop/rock Pop/rock Celtic rock	I can't be with you	Classic indie pop
	Cure	Rock	Alternative pop/ rock Goth rock Post-punk Dance-rock New wave College rock	Out of this world	Indie pop (by a modern standard)
Pair 8	Crazy Town	Rock	Heavy metal Alternative metal Rap-metal Rap-rock	Darkside	Mixture of rap-metal and pop
	Faithless	Electronica	House Club/dance Progressive house Progressive trance Dance-pop	Salva mea	Trance with a smooth rap vocal

Table 5.5: *Genre, style and audio information for the test song pairs as obtained from the All Music Guide. Song selection was blind to genre, style and audio information.*

		All Music Guide			Subjective evaluation		Hypothetical evaluation
	Artist	Audio similarity	Same genre	Possible style match	Subjective similarity (judge 1)	Subjective difference (judge 2)	Desirable lyric similarity based on genre/style
Pair 1	2pac	none	yes	yes	hard rap, almost identical song topic	none	yes
	Ghostface Killah						
Pair 2	Bruce Springsteen	none	yes	yes	folk style, similar verse and rhyming structure	song length	yes
	Elton John						
Pair 3	Bjork	medium	yes	no	very short lines/lyric, simplified expression	verse form	maybe
	Dido						
Pair 4	Anti-flag	none	yes	no	tongue-in-cheek, upbeat teenage rebellion	song length	maybe
	Barenaked Ladies						
Pair 5	Beach Boys	none	no	no	unusually upbeat sentiment	verse form, song length	no
	Enya						
Pair 6	Alkaline Trio	none	yes	no	black outlook, depression	verse form	maybe
	Jewel						
Pair 7	Cranberries	high	yes	yes	classic pop lyrics	none	yes
	Cure						
Pair 8	Crazy Town	none	no	no	mainstream rap, rolling stress	song length, line length, different type of stress pattern	no
	Faithless						

Table 5.6: *Analysis of test song pair meta-data and subjective lyric similarity. The last column shows the desirability of a lyric match based on audio and genre information.*

ity in this way, it transpired that the Davies-Bouldin Index was unable to distinguish between preferred maps, as determined by visual analysis, and those with very poor pairing of test songs. In part, this may be due to clusters being defined as exceptions to the mainstream, hence clustered points were actually quite distant. This resulted in a similar intra-cluster distance to inter-cluster distance, and a relatively static DBI value of 0.9 to 1 for all maps.

To solve this problem, an adapted version of the Davies-Bouldin index was used in which every pair of test songs was assumed to be a cluster. The intra-cluster distance was taken to be the Euclidean distance between the SOM prototype vectors assigned to each song in the pair. This approach was consistent with the finding of Logan (2004), who determined that minimum distance was the most effective means of measuring song similarity based on musical acoustics of 18,647 songs. There were four possible pairwise Euclidean distances between two clusters containing two songs each, and the minimum of those four distances was defined to be the inter-cluster distance. The pairwise similarity S was therefore calculated according to the equation:

$$S = \frac{1}{8}\sum_{i=1}^{8}\max_{i\neq j}\left\{\frac{\Delta(C_i)+\Delta(C_j)}{\delta(C_i,C_j)}\right\} \tag{5.1}$$

where $\Delta(C_i) = \|n_p - n_q\|$, $\delta(C_i,C_j) = \min_{x\in X}\|n_{ix} - n_{jx}\|$ and $X = \{p,q\}$ so that n_{ip} is the points represented by the prototype vector for the node associated with song p in pair C_i.

This clustering index provided an indication of the quality of pairwise matching by an SOM, but since it only considered the set of test pairs it cannot be assumed to reflect clustering validity over the entire data set. In particular, it may a provide better score for a map on which all the test songs are grouped together in a large cluster, than a map that has better pairwise matching but some songs positioned in areas of high dispersal. Further, the clustering metric includes the two song pairs that combined judgment of pair similarity suggested should not be matched. Despite these limitations, it provides a useful metric for analysis of results.

Chapter 6

Results

Results are reported for both visual and numerical analysis, starting with gold standard and IS data, followed by results for each of three classes of non-acoustic features (language, sentiment and repetition), and two acoustic repetition classes, one for stress patterns ordered by frequency and the other for stress patterns ordered by minimum separation distance. Finally, results are presented for feature combinations: non-acoustic, acoustic and combinations of non-acoustic, acoustic and IS features together.

Ebonics are only pertinent to non-acoustic feature sets and combinations since stress patterns were transcribed directly from non-ebonics corrected text. Results for these classes are therefore reported for *ebonics off*, in which lyrics are analysed without spelling, slang and other corrections, and *ebonics on*, in which these corrections are made prior to analysis. Since the ebonics dictionary covered approximately half the data, *ebonics off* results include both those for the full data set and those for the 1037 ebonics corrected songs. This enables fair comparison between *ebonics on* and *off* SOMs and help to gauge the impact of reducing or increasing the corpus size.

In all cases there is substantial change in the SOMs upon halving the corpus, indicating that much more data may be necessary to produce reliable results. In addition, since the SOM grid size for the *ebonics on* and acoustic SOMs was maintained at the same size as used for the full data corpus, comparisons between the full and half data sets, and between the *ebonics on*, and *ebonics off*, SOMs should be interpreted with caution. Some of the change in the SOM clustering is readily explained by variance in node assignment for two runs of the same SOM, even when using the same data and linear initialisation. To better understand how much variation between SOMs is due to the data, and how much is due to the self-organising behaviour of the SOMs themselves, it would be necessary to examine the differences between maps with ap-

	Group	q error	t error	q test error neighbour	q test error worst match	Pairwise clustering
Gold	aotm02	2.870	0.084	1.102	5.083	6.90E-02
IS	PCA75	4.286	0.014	1.172	20.068	1.47E-01
	PCA100	5.909	0.010	1.084	8.691	1.83E-01

Table 6.1: *Gold standard and IS baseline: quantised and topographic error for the whole SOM; average quantised error over all test songs for neighbouring and worst node assignment; similarity index for test pairs.*

proximately half the grid size or execute multiple (ten or more) initialisations using the same larger-scale map. The first option was undesirable for visualising clusters, and the second was not possible due to the time that would be required, hence the interpretations offered here pertain to general trends and are not conclusive.

6.1 Quantifying the 'goodness' of maps

6.1.1 Map quality

Overall SOM quality was assessed using average quantisation error (q) and topological accuracy (t). Quantisation error measures the distance between data vectors and prototype vectors, and topological accuracy is the percentage of data vectors for which first and second BMUs are not adjacent units. Generally, high-dimensional input spaces make it more difficult for SOMs to accurately represent the data, resulting in the higher quantisation error observed for the later feature combinations and IS maps. Quantisation error can be reduced by 'folding' the map, however, this increases topological error resulting in the trade-off between these measurements observed across all runs (Tables 6.1 - 6.4).

A general trend towards improved map quality was apparent with *ebonics on*, but this did not appear to be significant (e.g. q=1.677 and t=0.079 for the language cluster with *ebonics off*, full data, and q=1.597 with t=0.055 for *ebonics on*). The sentiment feature cluster, which was the smallest with only four attributes, produced the best overall map quality (q=0.564 *ebonics off*, full data, and q=0.525 *ebonics on*) although topological error was variable (*ebonics off*, full data, t=0.120 and half data t=0.051). Promising performance was seen for acoustic repetition features group 13C, repre-

	Group	Feature classes	q error	t error	q test error neighbour	q test error worst match	Pairwise clustering
All data (2392 songs)	9A	language cluster	1.677	0.079	1.026	5.195	2.34E-02
	12B	sentiment	0.564	0.120	1.122	13.433	1.24E-02
	13A	repetition cluster	1.144	0.086	1.094	6.897	3.66E-02
	15B non-acoustic all	form language grammar sentiment repetition gen. phrase length	2.338	0.084	1.032	3.298	2.20E-02
	16B non-acoustic limited	grammar repetition gen. phrase length	1.517	0.077	1.068	5.920	4.44E-02
Ebonic edit: ebonic off (1037 songs)	9A	language cluster	1.604	0.053	1.067	5.468	3.42E-02
	12B	sentiment	0.550	0.051	1.200	15.480	9.35E-03
	13A	repetition cluster	1.065	0.049	1.110	7.975	2.82E-02
	15B non-acoustic all	form language grammar sentiment repetition gen. phrase length	2.274	0.054	1.066	3.858	1.90E-02
	16B non-acoustic limited	grammar repetition gen. phrase length	1.456	0.045	1.066	6.096	4.67E-02
ebonic edit: ebonic on (1037 songs)	9A	language cluster	1.597	0.055	1.076	5.828	2.99E-02
	12B	sentiment	0.525	0.064	1.210	16.832	2.06E-02
	13A	repetition cluster	1.208	0.055	1.088	6.228	1.97E-02
	15B non-acoustic all	form language grammar sentiment repetition gen. phrase length	2.268	0.054	1.051	3.881	2.67E-02
	16B non-acoustic limited	grammar repetition gen. phrase length	1.596	0.041	1.058	4.616	1.51E-02

Table 6.2: *Non-acoustic repetition feature sets: quantised and topographic error for the whole SOM; average quantised error over all test songs for neighbouring and worst node assignment; similarity index for test pairs.*

	Group	Feature classes	q error	t error	q test error neighbour	q test error worst match	Pairwise clustering
Acoustic (binary)	10D global + Rstress	global stress R phrase rep. R phrase length R rep. dist. min. R rep. dist. max.	1.331	0.045	1.086	5.145	3.31E-02
	11D Dstress	D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	1.434	0.049	1.083	6.280	3.14E-02
	12D Dstress expanded	D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	2.640	0.041	1.050	3.276	5.57E-02
	13C global + Dstress	global stress D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	1.652	0.053	1.093	4.709	3.55E-02
	14C global + Rstress + Dstress	global stress R phrase rep. R phrase length R rep. dist. max. D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	2.604	0.068	1.067	3.030	3.82E-02
Acoustic (ncomma)	10F	global stress R phrase rep. R rep. dist. max.	to complete?				
	11F	global stress D phrase rep. D phrase length D rep. dist. max.					

Table 6.3: *Acoustic repetition feature sets: quantised and topographic error for the whole SOM; average quantised error over all test songs for neighbouring and worst node assignment; similarity index for test pairs.*

	Group	Feature classes		q error	t error	q test error neighbour	q test error worst match	Pairwise clustering
Ebonics on for group 15B	15B nonAc + 11D Ac (no Rstress)	form language grammar sentiment repetition gen.	phrase length D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	3.314	0.061	1.046	2.895	3.43E-02
	15B nonAc + 14C Ac (with Rstress)	form language grammar sentiment repetition gen. phrase length global stress	R phrase rep. R phrase length R rep.dist. max. D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	4.057	0.046	1.038	2.578	4.50E-02
	IS + 15B nonAc + 11D Ac (no Rstress)	LSA pca75 form language grammar sentiment repetition gen.	phrase length D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	5.747	0.025	1.081	3.547	2.97E-01
	IS + 15B nonAc + 14C Ac (with Rstress)	LSA pca75 form language grammar sentiment repetition gen. phrase length global stress	R phrase rep. R phrase length R rep.dist. max. D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	6.382	0.031	1.074	2.797	2.38E-01
Ebonics off for group 15B	15B nonAc + 11D Ac (no Rstress)	form language grammar sentiment repetition gen.	phrase length D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	3.299	0.057	1.051	2.931	3.89E-02
	15B nonAc + 14C Ac (with Rstress)	form language grammar sentiment repetition gen. phrase length global stress	R phrase rep. R phrase length R rep.dist. max. D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	4.059	0.047	1.039	2.542	4.04E-02
	IS + 15B nonAc + 11D Ac (no Rstress)	LSA pca75 form language grammar sentiment repetition gen.	phrase length D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	5.670	0.036	1.077	3.395	2.79E-01
	IS + 15B nonAc + 14C Ac (with Rstress)	LSA pca75 form language grammar sentiment repetition gen. phrase length global stress	R phrase rep. R phrase length R rep.dist. max. D phrase rep. D phrase length D rep. dist. min. D rep. dist. max.	6.341	0.030	1.083	2.994	2.66E-01

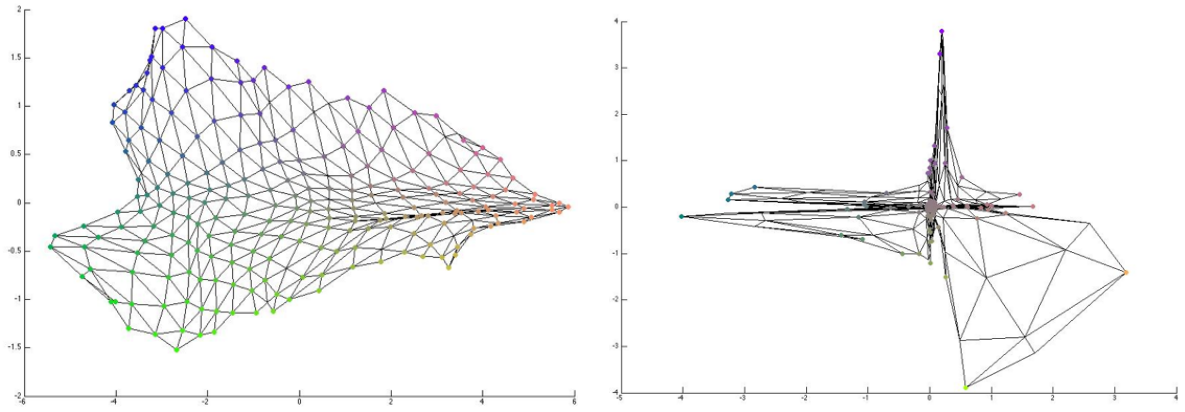
Table 6.4: *Combination feature sets: quantised and topographic error for the whole SOM; average quantised error over all test songs for neighbouring and worst node assignment; similarity index for test pairs.*

senting features for global stress patterns and patterns ordered by minimum separation distance (*Dstress*) and 14C, the same with the addition of features from patterns ordered by frequency (*acoustic*). Group 13C with 20 features produced comparable maps to clusters with half as many non-acoustic features (q=1.652 and t=0.053 for 13C, q=1.596 and t=0.041 for 16B with 10 features) whilst group 14C suffered only a marginal decrease in map quality with 27 features compared to the largest non-acoustic feature cluster with 15 features (q=2.604 and t=0.068 for 14C, q=2.268 and t=0.054 for 15B).

6.1.2 Map accuracy: test pairs

SOMs were interpreted visually with respect to test song pairs, so the accuracy of individual node assignments for these lyrics was evaluated. Quantisation error was calculated between each test song vector and its neighbouring BMU, and between the vectors and their worst possible node assignments. Best and worst BMU quantizations errors were averaged across all test songs and recorded in Tables 6.1 - 6.4. A large difference between the two values indicates good accuracy in node assignments, whilst a very small difference suggests low accuracy that should be investigated further. Unfortunately, there was not enough time to investigate the quantisation error. It is possible that no confidence could be placed in node assignment if the variation in quantised error for neighbouring BMUs following different initialisations of a map is greater than the difference between average quantised error for the best and worst matching nodes of every test song.

What can be observed in relation to test song node assignments, is that the accuracy of most single cluster maps (e.g. *language, sentiment, Dstress*) seems low, with differences of around 4 to 6. It is not clear if this is due to poor map quality or if homogeneity of the data is also a factor, in that songs are distinguishable only by very small differences. What is clear is that more data is required to elucidate these questions. The highest accuracy is observed in the IS map (best q=1.172, worst q=20.068), strongly followed by the SOM for sentiment features (best q=1.210, worst q=16.832, *ebonics on*). On the other hand, combination feature sets show deteriorating accuracy, and for maps with both non-acoustic and acoustic input features the average quantised error for the worst matching nodes is significantly less than the best quantised error over the whole map, raising concern about the degree of confidence with which these maps may be regarded.



(a) gold standard aotm02 (b) IS using PCA 75 dimensions

Figure 6.1: *Clusters shown by projection onto top two principle components.*

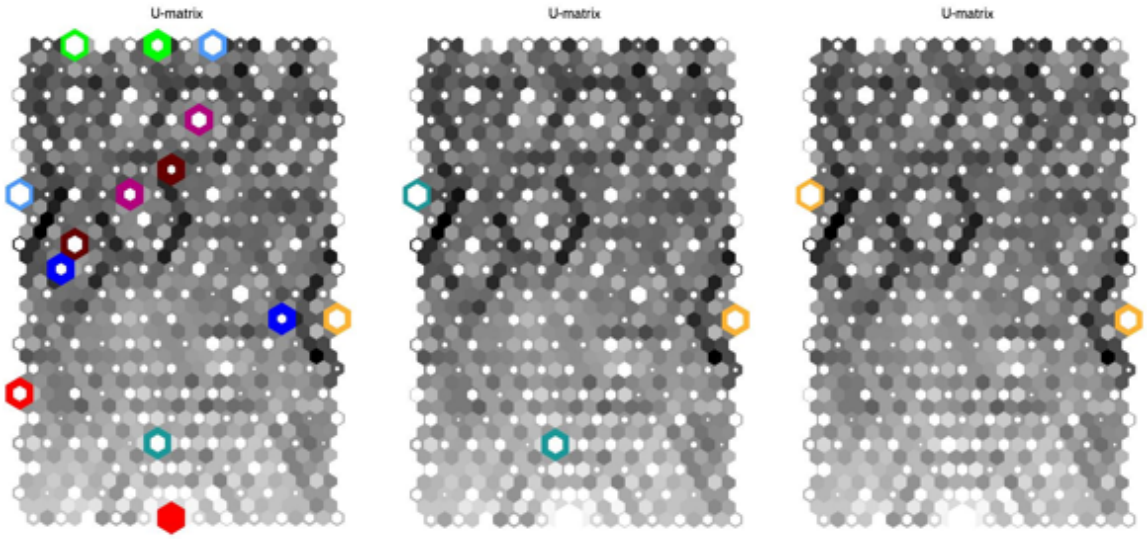
6.2 Pairwise clustering

SOM performance in matching test was assessed through visualisation and application of the clustering validity metric defined in Chapter 5. Visual representations were particular;y important to help understand the relationships between attributes and the final clustering. This Section will address interpretation of visualisations. Analysis of the types of errors made by each feature class and clustering will follow in the next Section.

6.2.1 Gold standard

The most remarkable characteristic of the gold standard SOM is an even spread nodes across much of the map, with bunching at one end that appears to represent mainstream pop music. This can be seen clearly in the three-dimensional graph (Figure 6.1), and again in the topological representation (Figure 6.3) where there is a relatively smooth gradient from yellow/orange down to the blue area of node bunching. Hit matrices (Figure 6.2) show that most of the test songs are some distance removed from the main cluster, and that many sit on the edge of the map where we would expect to find outliers. This is consistent with their being representative of identifiable styles, although perhaps surprisingly, the songs selected for having standard pop lyrics (magenta) also lie away from the main group. The obvious exception is the Beach Boys (red), which appears at the centre of the mainstream cluster.

The SOM strongly suggests that the playlists were determined by musical similarity, and that the lyrical similarity this research seeks to investigate is something unique



(a) song hits (b) overlap turquoise/blue (c) overlap yellow/turquoise

Figure 6.2: *Gold standard aotm02: test song pair hits.*

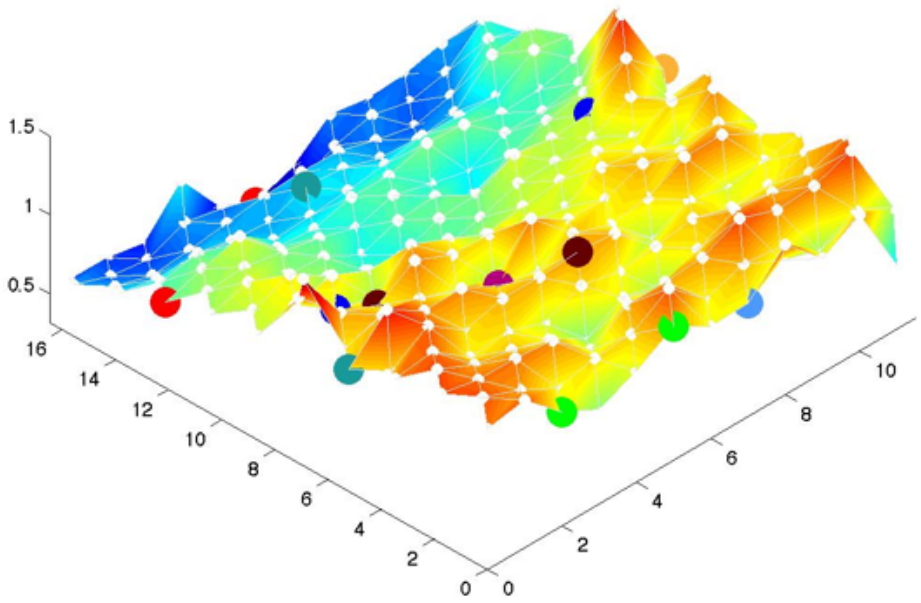


Figure 6.3: *Gold standard aotm02: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*

and quite separate. Observations about which songs are most closely paired by the SOM support this hypothesis, as shown in the error analysis chart in Table 6.5 . The closest pair are Bjork and Dido, who have different styles but share the same genre, whilst Crazy Town and Faithless are more strongly separated. The latter are indeed very different musically; Crazy Town is a mixture of rap-metal and pop, and the track from Faithless is set to trance. Anti-flag's youth punk pop and the Barenaked Ladies, whose recording might best be described as comedy ska, are also very separate. The overlap between one song each from the turquoise, light blue and yellow pairs is harder to understand but may be due to a consistent upbeat and a wide variety of instrumental sound.

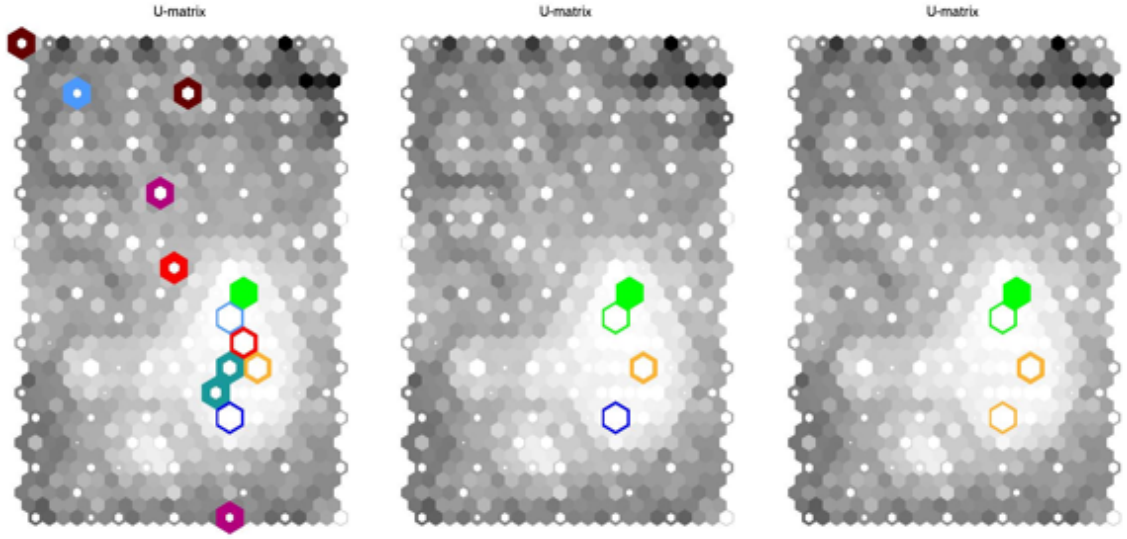
6.2.2 Information Space

The IS map clusters many songs into a central group, visualised in the topological representation as a single deep pool into which most of the nodes have sunk (Figure 6.5). IS performs well in identifying the two hardcore rap songs, which are shown on the same node in royal blue (Figure 6.4), consistent with their highly similar lyrical content. It also does a good job of matching the mainstream rap tracks (in turquoise), and separately matching Bjork and Dido (in green), however it does a poor job of distinguishing these from the more traditionally structured (light blue), tongue-in-cheek (yellow) and classic pop (Beach Boys, red) lyrics. What will be observed in comparison with other means of feature extraction is that IS appears to be good at clustering but fails to make fine-grained distinctions in a fairly homogenous data set with very short documents.

6.2.3 Language

The SOM for language suggests these features are of no use in identifying similar lyrics. The song pairs appear to be scattered across the map, but as will be shown later, they may have a role to play in combination models. The interesting aspect of language features is that of all the non-acoustic classes tried they responded most to ebonics correction. This can be seen clearly in the 3-dimensional representations of clustering, where there is a significantly tighter bunching of nodes for the ebonics corrected data (Figure 6.6). This is also reflected by the change in topology for the distance matrix shown in Figure 6.8.

The reason for the topological shift becomes clear when examining the distribution



(a) song hits (b) overlap light blue/green (c) overlap yellow/blue

Figure 6.4: *IS using PCA to 75 dimensions: test song pair hits.*

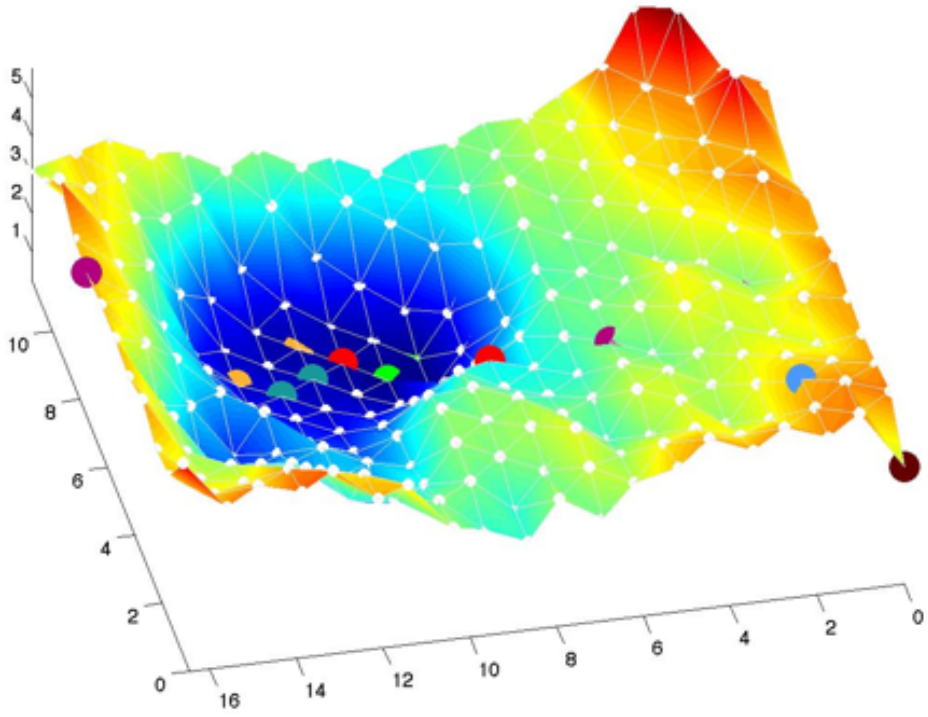
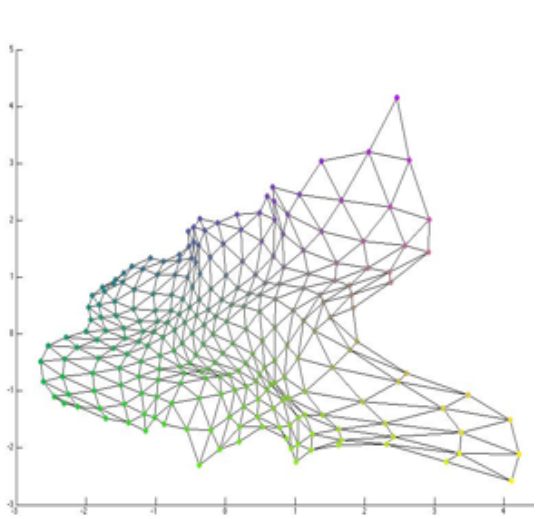
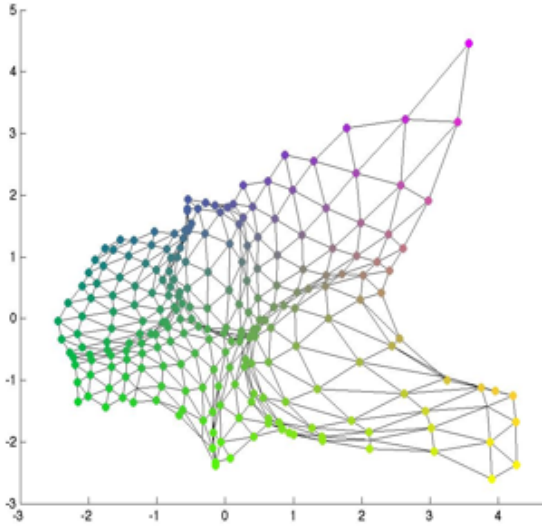


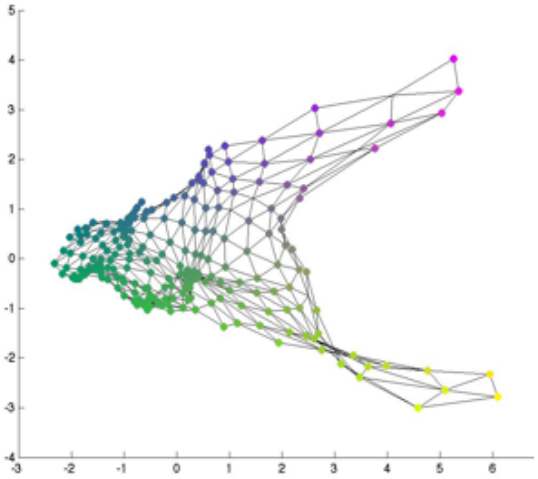
Figure 6.5: *IS using PCA to 75 dimensions: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*



(a) *ebonics off*, 2392 songs

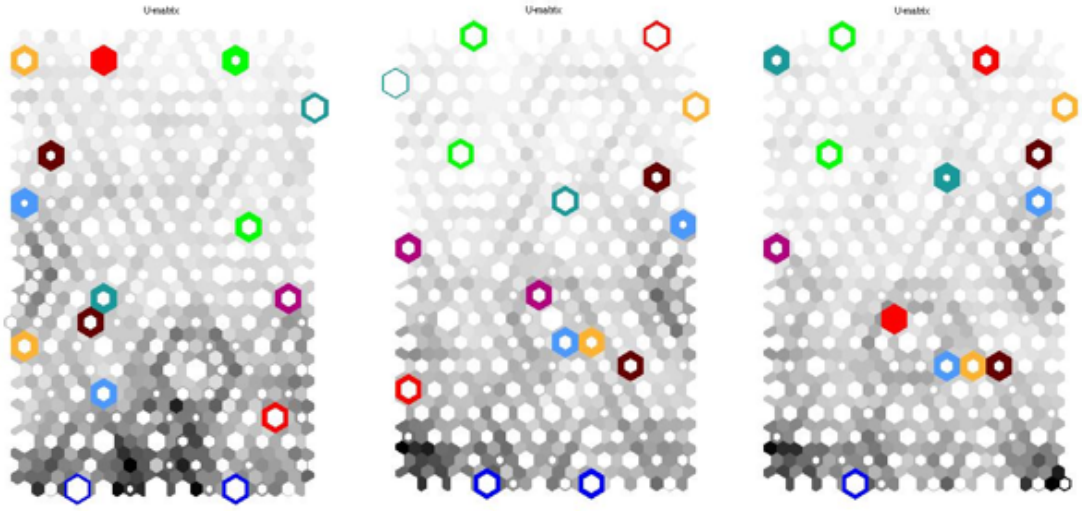


(b) *ebonics off*, 1037 songs



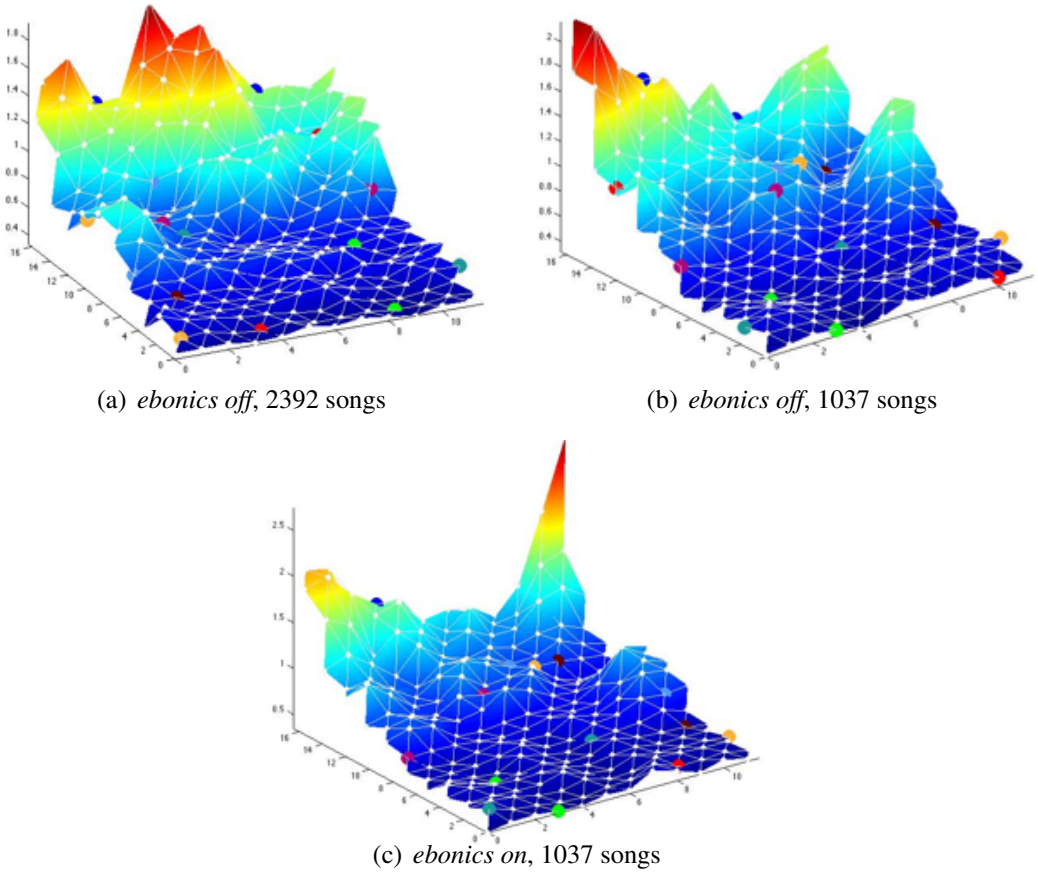
(c) *ebonics on*, 1037 songs

Figure 6.6: *Non-acoustic features: clusters shown by projection onto top two principle components.*



(a) *ebonics off*, 2392 songs (b) *ebonics off*, 1037 songs (c) *ebonics on*, 1037 songs

Figure 6.7: *Language cluster features: test song pair hits.*



(a) *ebonics off*, 2392 songs (b) *ebonics off*, 1037 songs

(c) *ebonics on*, 1037 songs

Figure 6.8: *Language cluster features: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*

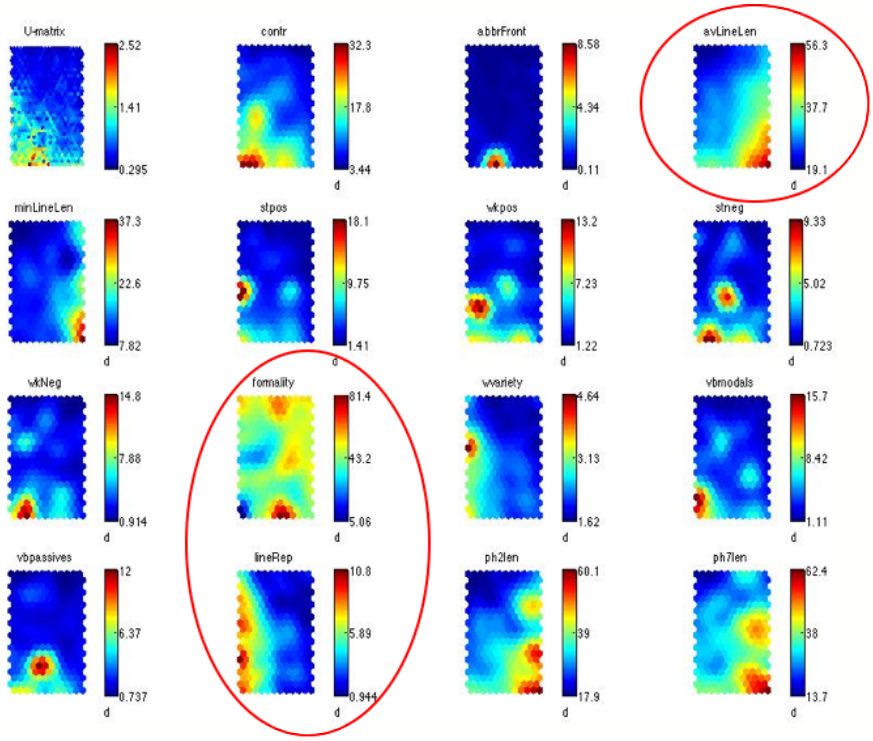


Figure 6.9: *Component analysis for 2392 songs, ebonics off. Components with greatest change circled.*

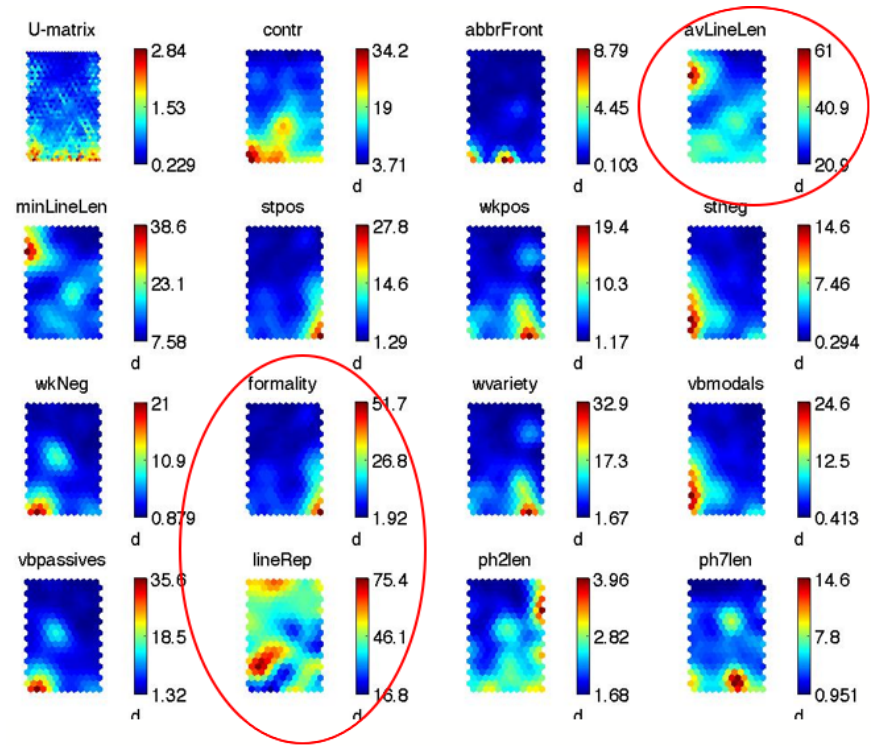


Figure 6.10: *Component analysis for 1037 songs, ebonics on. Components with greatest change circled.*

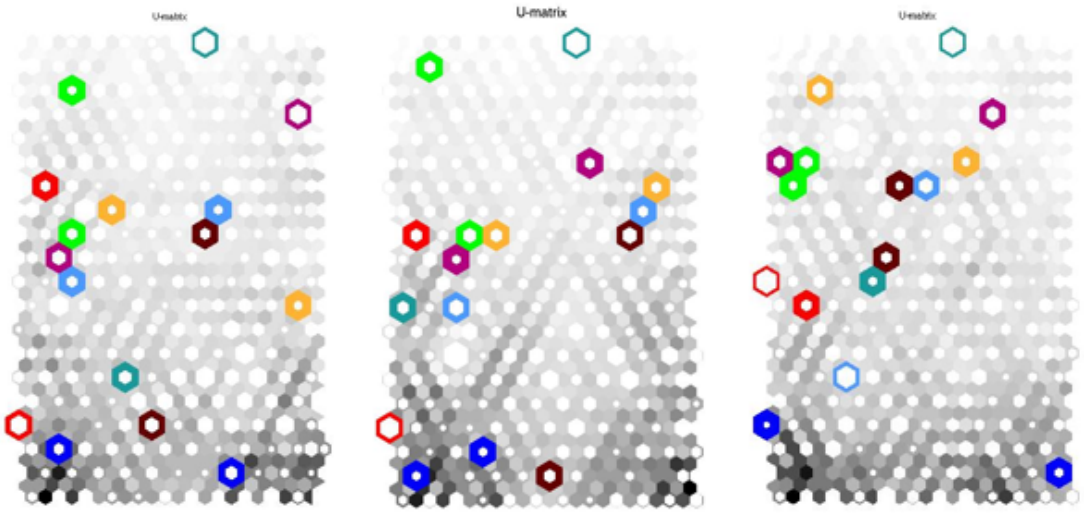
of nodes in the component matrices (Figures 6.9, 6.10). The components that alter most with ebonics correction are language formality, defined as the percentage of formal (*NN, JJ, IN, DT*) versus informal (*PRP, RB, UH, VB*) words, the number of lines that are repeated in full, and the average line length. Change in the first attribute can probably be ascribed to improved performance of the POS tagger, which would have struggled to correctly tag OOV words. More accurate tagging has made the formality ratio more consistent across the whole data set, indicative of a text genre. The ebonics dictionary also corrected many typographic errors such as *'therewas'*, increasing average line length, and standardising words such as *'oooh'* and *'ooh'* (to *'oh'*), resulting in more line repetition matches. The language class appears to perform better as a result, particularly with respect to the hardcore and mainstream rap pairs (royal blue and turquoise - see Table 6.5).

The topology for the distance matrix (Figure 6.8) also illustrates a common thread throughout the SOMs: a significant change in the maps corresponding to a change in the amount of data used. These changes suggest that the amount of data used was not sufficiently representative of all possible lyrics to ensure map stability. The marked spike on one corner of the map rising above a low blue area of node clustering is also typical of what will be observed in later maps, namely that the vast majority of lyrics are extremely similar and that in any feature category there are a minority of unique examples that make particular songs stand out.

6.2.4 Sentiment

Sentiment features fare little better than language features in clustering an SOM, although there is a marked improvement in the proximity of several of the pairs, particularly red, selected for unambiguous positivity, brown, selected for marked melancholy; and green, the pair best-matched by gold standard data. Sentiment is the only independent feature class that manages to bring the red songs together, although they are not close enough to be described as a match (Figure 6.11). These results are unsurprising, but the close proximity of the red and brown pairs is curious since they were intended to represent opposite ends of the sentiment spectrum. This suggests that they share something else in common, like extremity of emotion.

During selection of the red and brown song pairs, it was difficult to find lyrics that were unambiguously positive or negative from a purely linguistic viewpoint. Many lyrics could be described as bittersweet, containing emotionally charged language that



(a) *ebonics off*, 2392 songs (b) *ebonics off*, 1037 songs (c) *ebonics on*, 1037 songs

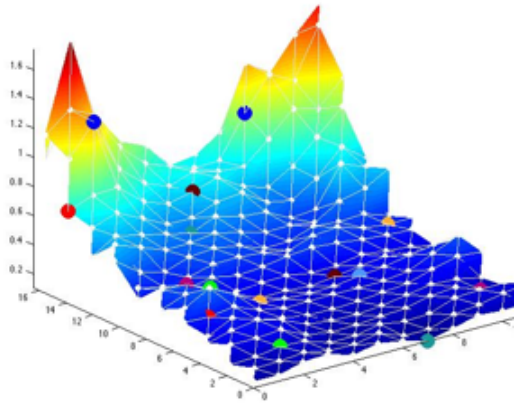
Figure 6.11: *Sentiment cluster features: test song pair hits.*

was roughly equivalent in negative and positive connotation. Statistics for all the data, *ebonics off* (Table 4.4), show most songs contain between 1 and 15 words associated with positive sentiment and 1 to 12 words indicating negative sentiment. Whilst results for sentiment analysis might have improved with the use of a feature that combined positive and negative orientation, such as relative sentiment or a sum over the orientation of all words, the large area of clustering indicated by a low flat plain in the distance matrix topology (Figure 6.12), suggests otherwise. Using this visualisation as a guide, it appears that the vast majority of songs contain a balance of positive and negative sentiment and that only a minority strongly favour each emotional extreme. The positive and negative poles of sentiment are represented by two spikes at opposite corners of the map that become less pronounced with ebonics correction.

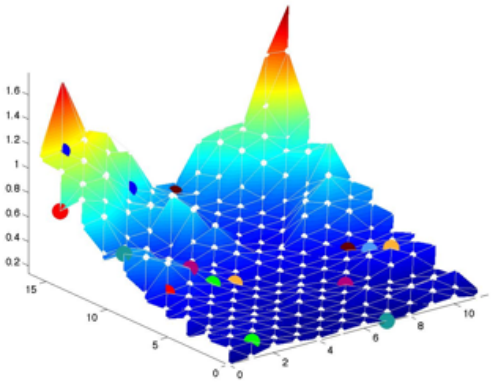
6.2.5 Repetition

The repetition features are not as effective as book recommendations on how to write a hit song suggest (Dhanaraj and Logan, 2005). This may be because the data is biased towards popular music, resulting in many songs being repetitive.

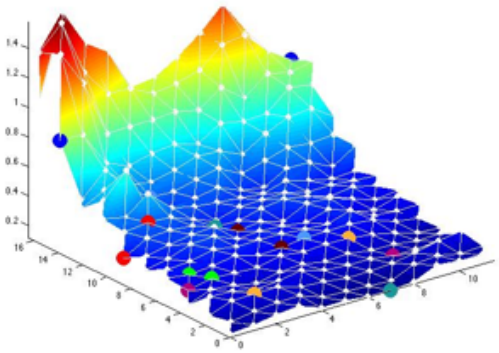
The topological representation of the distance matrix (Figure 6.14) shows a sharp spike for non-ebonics corrected lyrics and a node at the peak associated with a rap song. This suggests that the spike represents songs that have very little, if any, repetition, since rap is generally recounted in a non-repetitive, narrative fashion. This



(a) *ebonics off*, 2392 songs

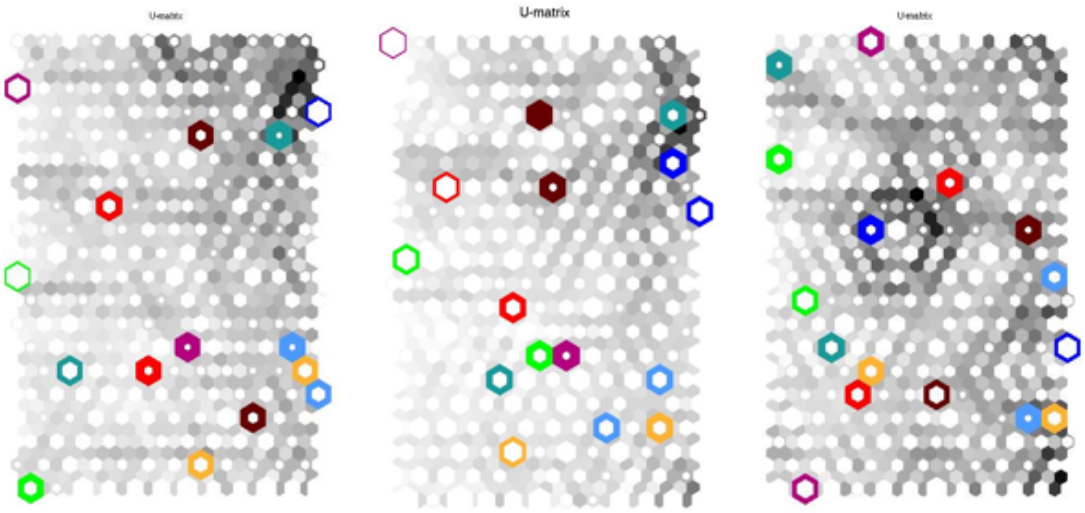


(b) *ebonics off*, 1037 songs



(c) *ebonics on*, 1037 songs

Figure 6.12: *Sentiment cluster features: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*



(a) *ebonics off*, 2392 songs (b) *ebonics off*, 1037 songs (c) *ebonics on*, 1037 songs

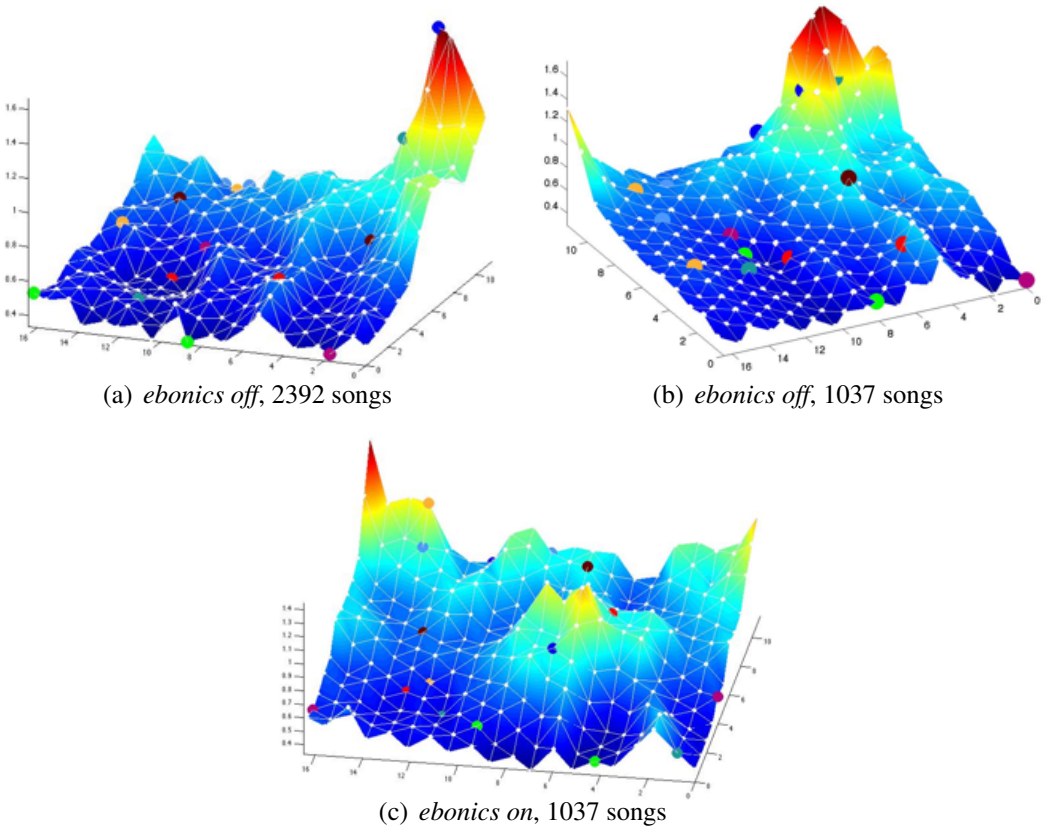
Figure 6.13: *Repetition cluster features: test song pair hits.*

interpretation of the graph is consistent with deteriorating ability of phrase repetition features to identify rap songs following ebonics correction. The ebonics dictionary standardises much rap slang and may introduce some repetition where previously there was none. Consequently the SOM expends less effort to accommodate non-repeating outliers, and becomes more sensitive to either very little or a great deal of repetition. This is shown by two peaks in the *ebonics on* topological map, one for songs with very little repetition and the other for songs with greater than average repetition.

6.2.6 Non-acoustic combination

Despite the shortcomings of language, sentiment and repetition features as individual sets, the SOM for non-acoustic combined features, *ebonics off*, is the best of all maps assessed (Figures 6.15, 6.16). Visible clustering is good, and both yellow songs are also close together, directly underneath the light blue pair on the right. In addition, although the red and turquoise songs are apart, there are no songs from other pairs in between them, as there are for many other maps. This reflects good organisation of lyrics by the SOM, which is especially apparent when looking at the topological representation of the distance matrix (Figure 6.16).

The SOM generated with *ebonics off*, full data, appears to be doing much better than the SOM generated with ebonics corrected lyrics, but this result needs to be confirmed. The map produced with *ebonics off* using the same amount of data as *ebonics*

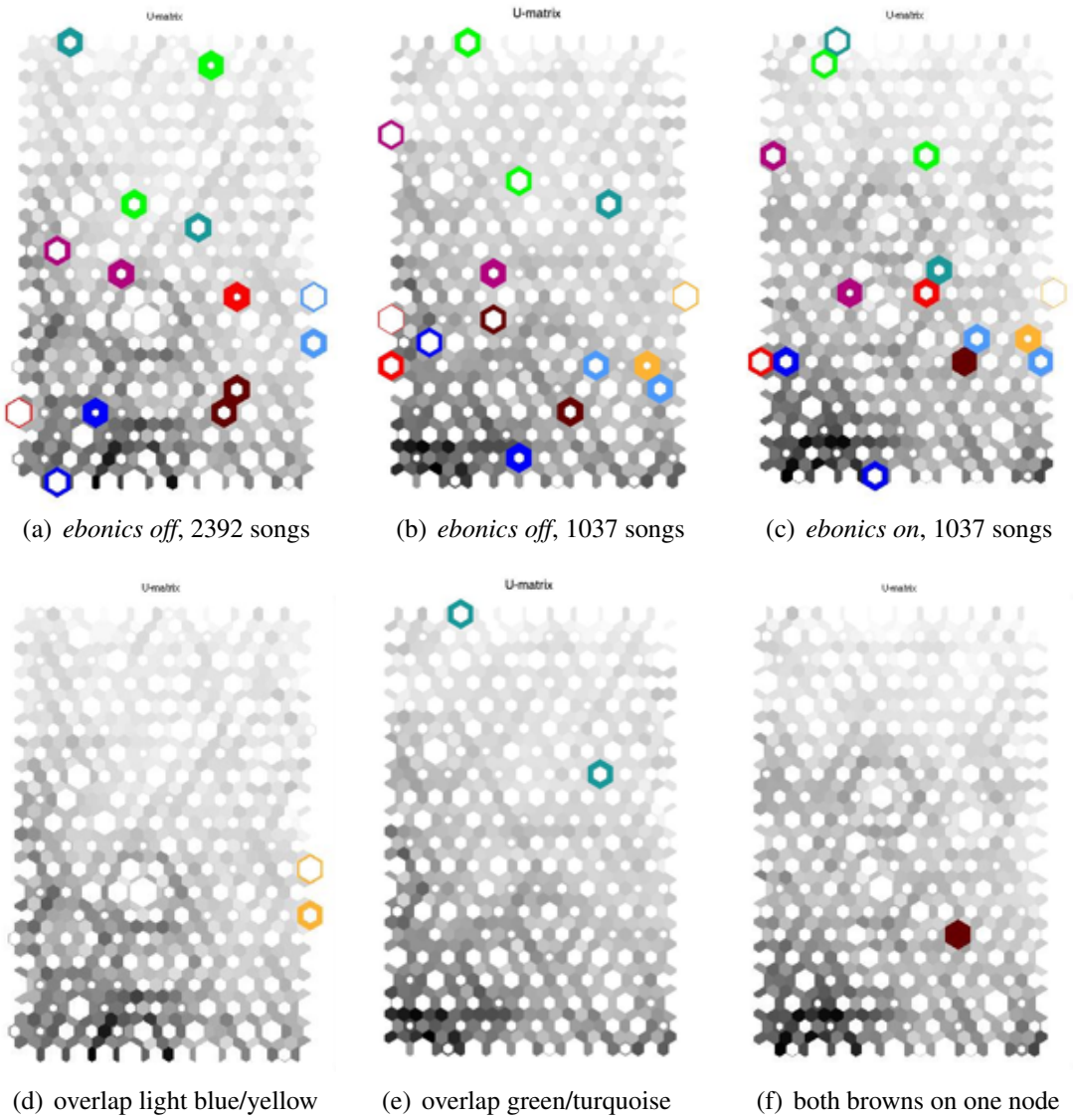


(a) *ebonics off*, 2392 songs

(b) *ebonics off*, 1037 songs

(c) *ebonics on*, 1037 songs

Figure 6.14: *Repetition cluster features: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*



(a) *ebonics off*, 2392 songs (b) *ebonics off*, 1037 songs (c) *ebonics on*, 1037 songs

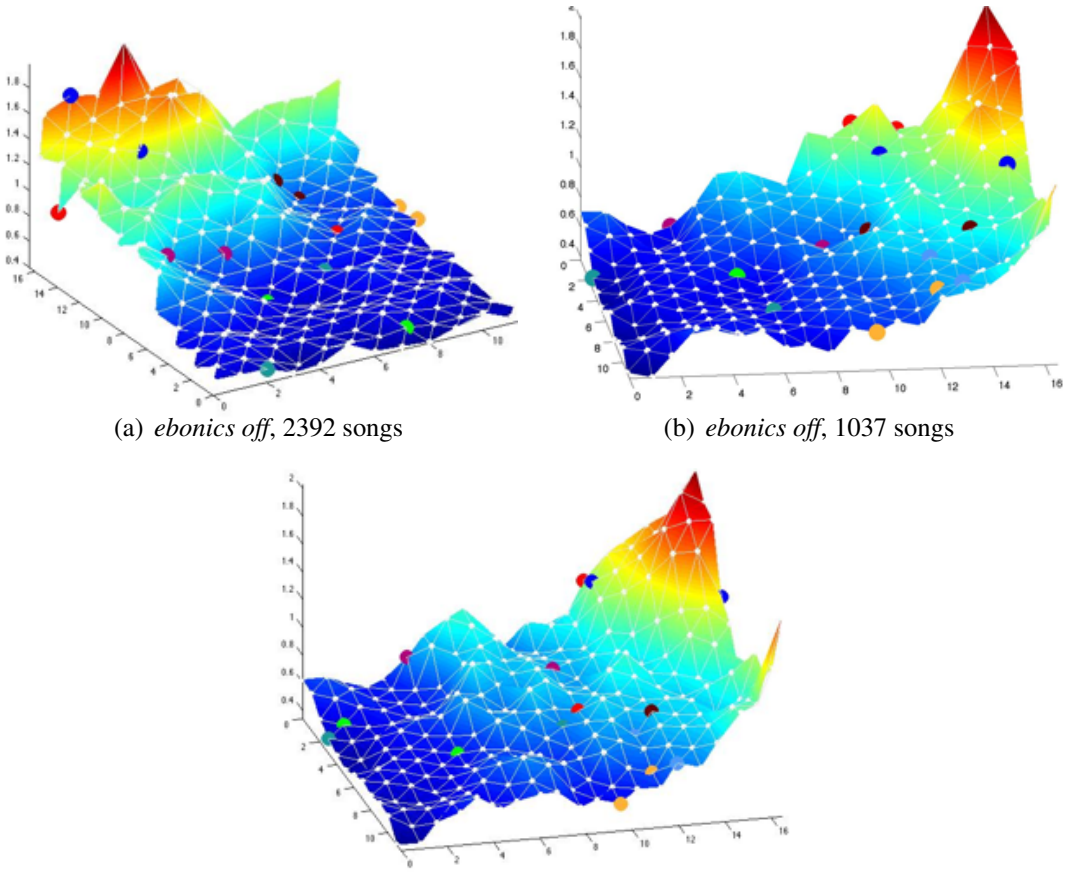
(d) overlap light blue/yellow (e) overlap green/turquoise (f) both browns on one node

Figure 6.15: *All non-acoustic features: test song pair hits.*

on shows significantly worse performance than the full data version. It seems likely that improved performance on the full data, *ebonics off* SOM may be due to the larger corpus, once again highlighting the need for more data to better study these interactions. It does suggest, however, what can be achieved without going to the effort of an ebonics dictionary.

6.2.7 Acoustic combination

Just as sentiment features were expected to do best on songs that heavily favoured one emotional extreme, acoustic features were expected to do best on songs for which the

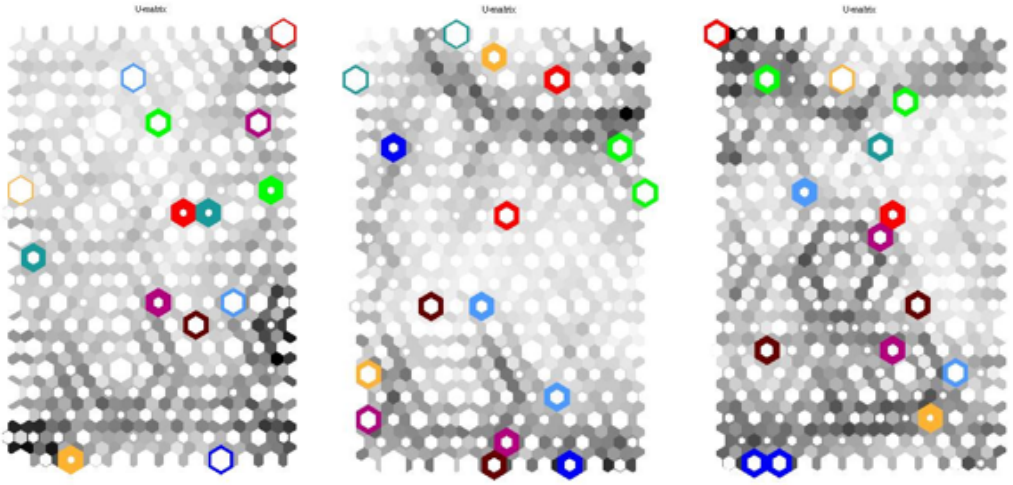


(a) *ebonics off*, 2392 songs

(b) *ebonics off*, 1037 songs

(c) *ebonics on*, 1037 songs

Figure 6.16: *All non-acoustic features: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*



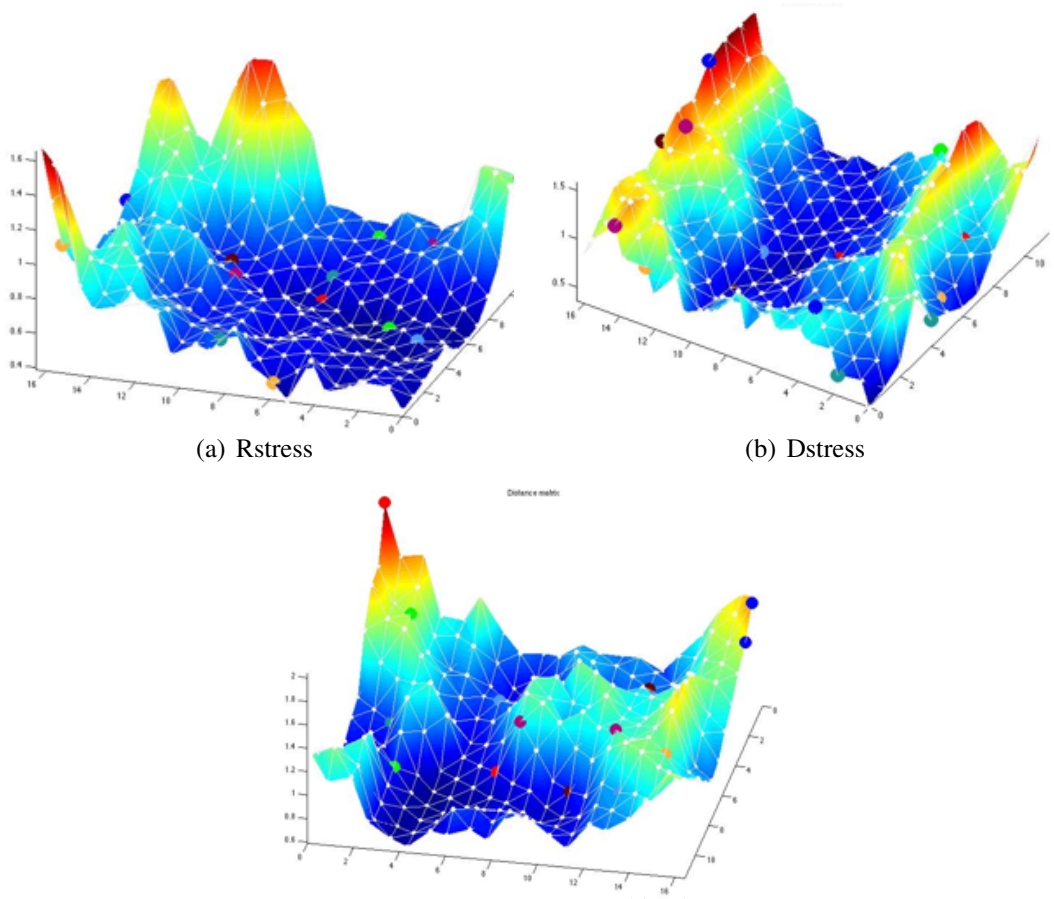
(a) Rstress (b) Dstress (c) Rstress/Dstress combined

Figure 6.17: *Acoustic cluster features: test song pair hits.*

rhythm of the lyric was most important. These were the hardcore rap songs (royal blue), mainstream rap (turquoise) and lyrics written in a folk style with regular stanzas and rhyming line-ends (light blue). Whilst the SOM clusters were not impressive, features for stress patterns ordered by frequency (*Rstress*) showed a distinctly different strength to those ordered by minimum separation distance (*Dstress*, see Figure 6.17). Ordering by pattern frequency placed the hardcore rap songs on the same node, whilst ordering by separation distance moved them apart. This success was reversed in the case of mainstream rap and folk-style lyrics, which came closer together using features ordered by minimum separation distance (note the second turquoise song is underneath the nearby yellow).

It appears that there are two types of important stress pattern repetition at work: regularity, or the back-to-back repetition of certain patterns, and periodicity, or the recurrence of certain stress patterns at interval. The first type characterises traditional, structured poetic form, whilst the latter is associated with the freeform pattern observed in rap.

The combination of these two types of features does well for hardcore and mainstream rap, but fails to bring together the folk style lyrics. The topological representation of the distance matrix for these SOMs (Figure 6.18) indicates there is something quite complex going on, and multiple blue pools suggest the potential for good clustering. The relationship between the lyrics and the topology here is poorly understood, however, and the location of the song pairs is not instructive.



(a) Rstress (b) Dstress

(c) Rstress/Dstress combined

Figure 6.18: *Acoustic cluster features: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*

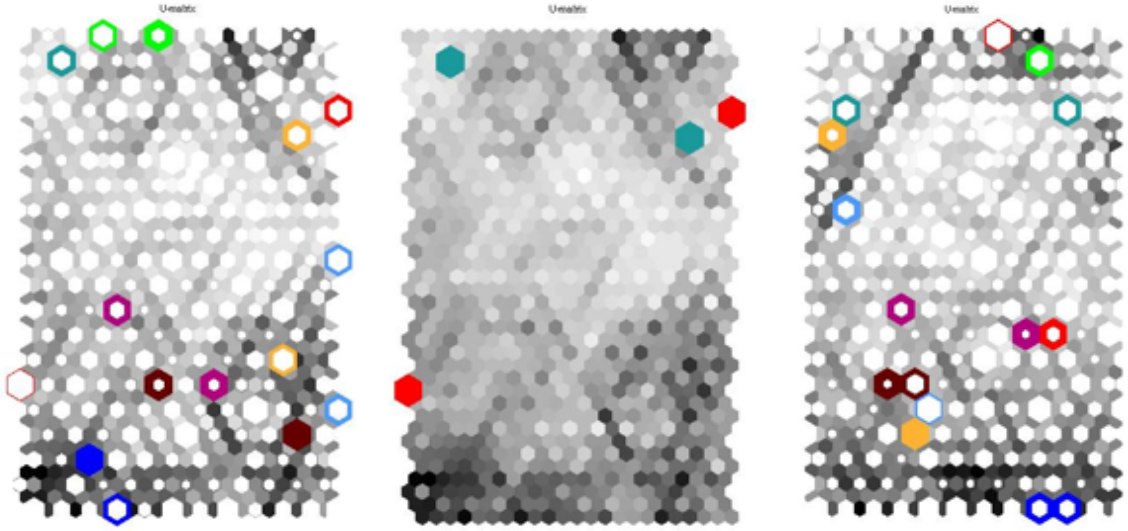
6.2.8 Non-acoustic and acoustic combinations

Combination SOMs were generally no better than those for individual clusters, or non-acoustic and acoustic groups on their own. Distant pairs were a mixture of those separated by the component features with a fairly equal degree of influence from each component set. This reflects a balance in the number of features in each: 18 features in the combined acoustic set and 15 features in the combined non-acoustic. As observed for several component classes, *ebonics on* was generally worse than *ebonics off*. In addition, groups including Rstress features (ordered by stress pattern frequency) made more errors than those without these features, although they also identified more pair matches (Figures 6.19 - 6.26).

Combinations including IS tended to be dominated by IS map characteristics and produced over-generalised clustering. This is probably due to the fact that there were 75 IS features compared to only 33 features for acoustic and non-acoustic combined. Curiously, topologic representations (Figure 6.24) using IS in combination with features not including Rstress resulted in a small bundle for hardcore rap splitting off the main cluster. On the other hand, when Rstress was included, the main cluster, represented as a deep blue pool, was split into two, the smaller of which contained many test songs whilst the larger pool contained only one of the folk-style lyrics (light blue). This pooling behaviour cannot be understood with the information available. The remaining songs were distributed towards the outer edges of the map, supporting the perspective suggested by the gold standard SOM that the lyrics selected for testing were somehow different from the majority.

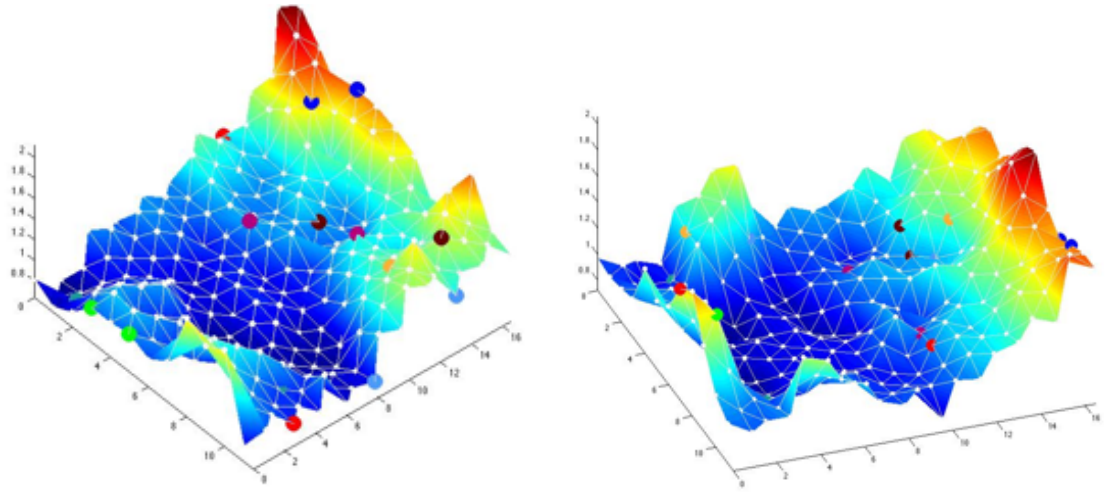
6.3 Error analysis

Tables 6.5 and 6.6 provide an overview of the types of errors made by each SOM and the song pairs to which these errors pertain. Basic errors were separation of a pair of songs that were deemed similar, the lower bound of which was lack of any clustering at all. This tendency was seen with the gold standard and language SOMs, suggesting that lyric relationships were based on some other metric. On the other hand, some maps clustered songs from two different pairs together, the extreme of which was over-generalised clustering, observed with the SOMs using input features from IS analysis. These maps clustered very well but failed to distinguish between lyrics that were similar, yet had different styles.



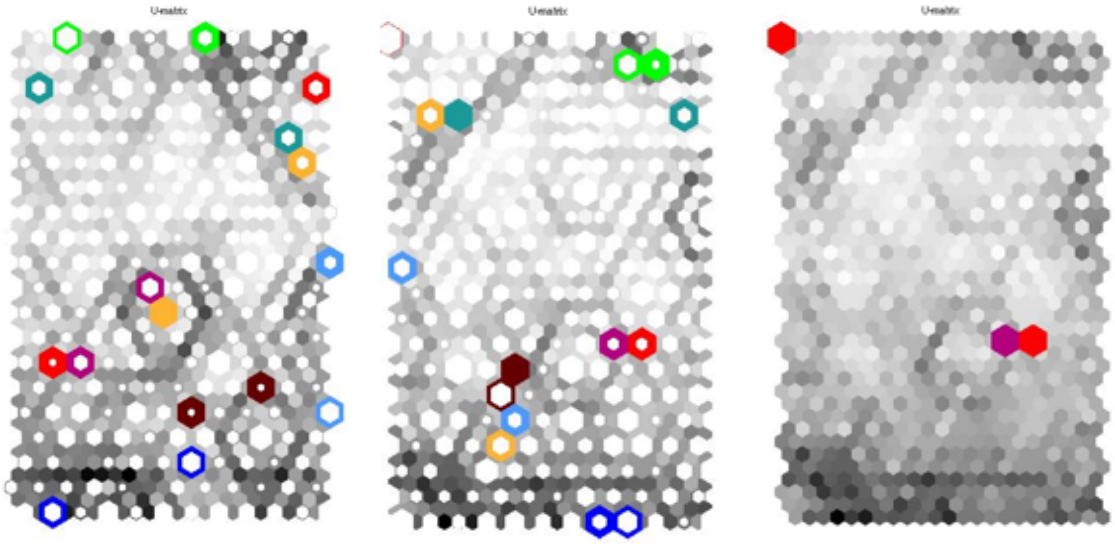
(a) no Rstress (b) overlap turquoise/yellow (c) with Rstress

Figure 6.19: *Non-acoustic and acoustic feature combinations, ebonics off 1037 songs: test song pair hits.*



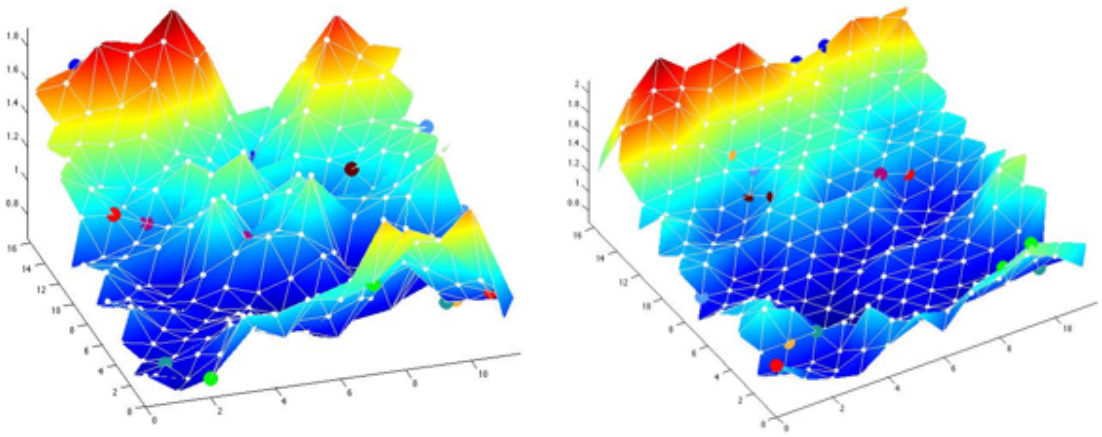
(a) no Rstress (b) with Rstress

Figure 6.20: *Non-acoustic and acoustic feature combinations, ebonics off 1037 songs: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*



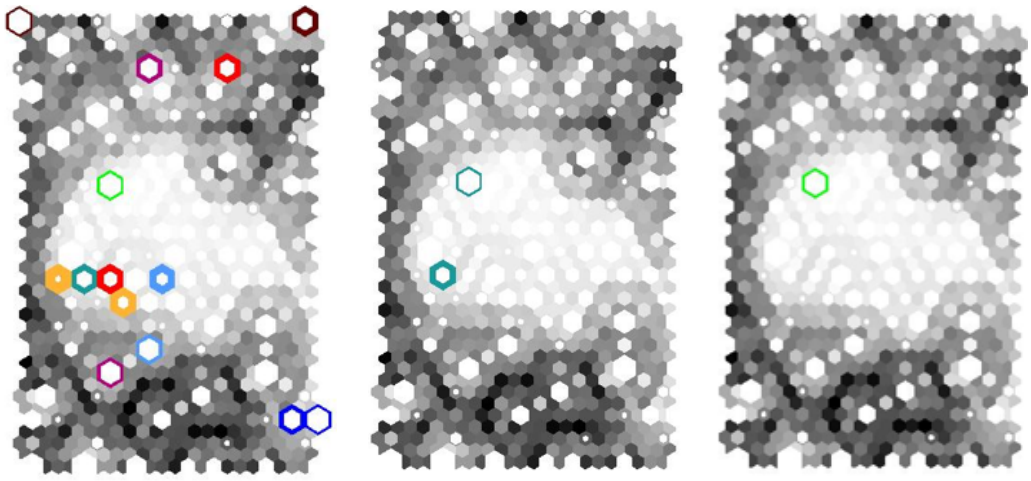
(a) no Rstress (b) with Rstress (c) red on the corner node

Figure 6.21: *Non-acoustic and acoustic feature combinations, ebonics on 1037 songs: test song pair hits.*

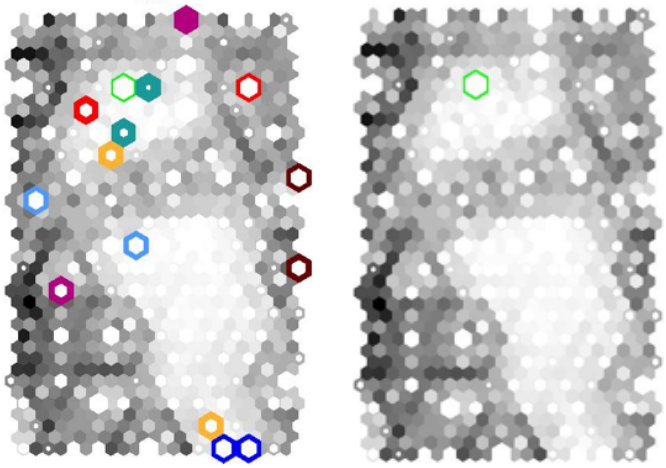


(a) no Rstress (b) with Rstress

Figure 6.22: *Non-acoustic and acoustic feature combinations, ebonics on 1037 songs: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*

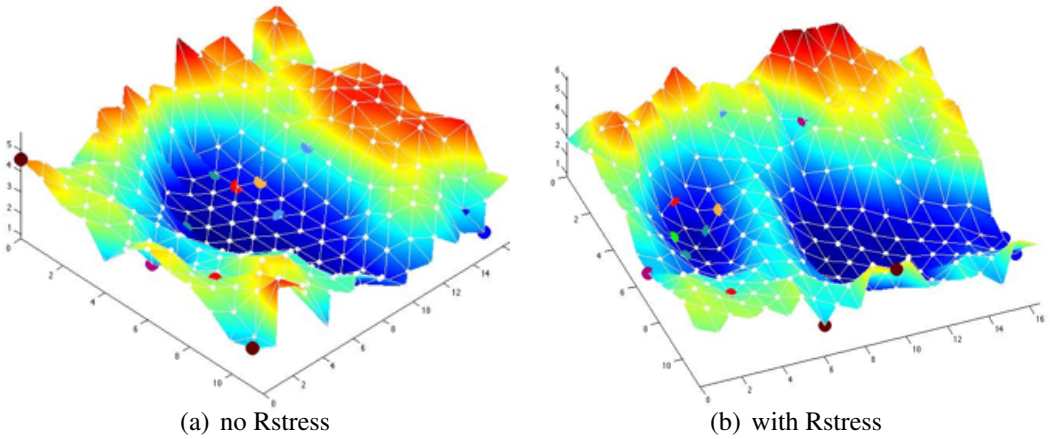


(a) no Rstress (b) overlap turquoise/green (c) two green on one node



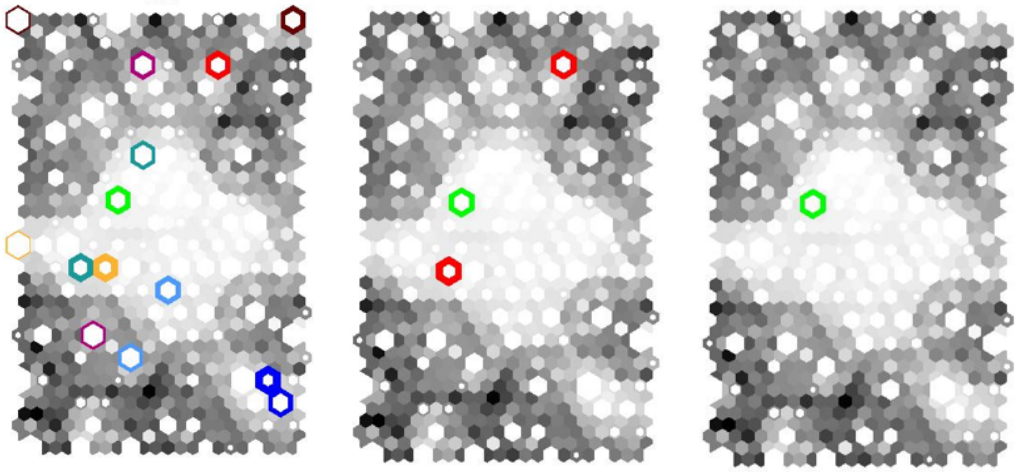
(d) with Rstress (e) two green on one node

Figure 6.23: *Non-acoustic, acoustic and IS features, ebonics off 1037 songs: test song pair hits. Not including Rstress ordered by pattern frequency (a-c), with Rstress (d-e).*

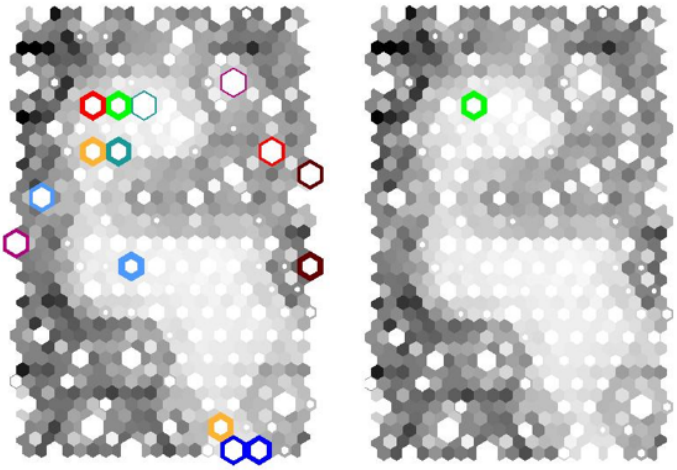


(a) no Rstress (b) with Rstress

Figure 6.24: *Non-acoustic, acoustic and IS features, ebonics off 1037 songs: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*

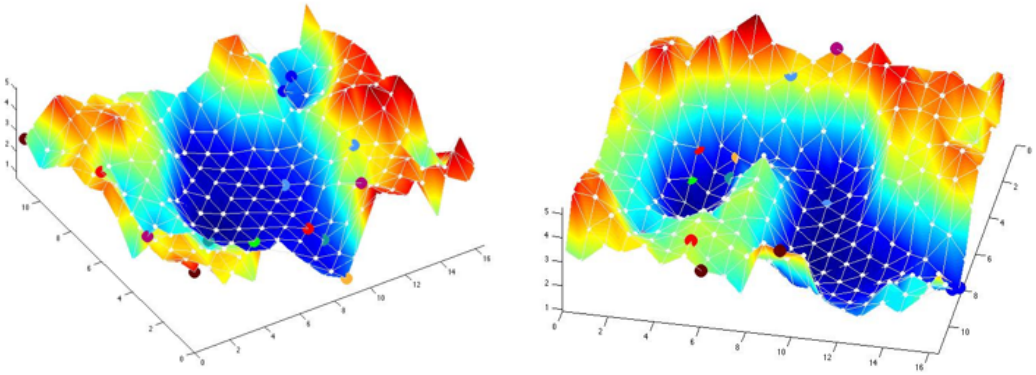


(a) no Rstress (b) overlap red/yellow (c) two green on one node



(d) with Rstress (e) two green on one node

Figure 6.25: *Non-acoustic, acoustic and IS features, ebonics on 1037 songs: test song pair hits. Not including Rstress ordered by pattern frequency (a-c), with Rstress (d-e).*



(a) no Rstress (underside to show dots) (b) with Rstress

Figure 6.26: *Non-acoustic, acoustic and IS features, ebonics on 1037 songs: distance matrix represented as topography, with test song pair hits indicated by coloured dots.*

	Error	Gold	IS	Language	Sentiment	Repetition	Non-acoustic	Rstress	Dstress	Acoustic
Ebonics off (2392 songs)	Separated pair									
	Incorrect pair									
	Generalised clustering		x							
	Lack of clustering	x		x						
	Correct pair									

	Error	Gold	IS	Language	Sentiment	Repetition	Non-acoustic	Rstress	Dstress	Acoustic
Ebonics on (1037 songs)	Separated pair	n/a	n/a					n/a	n/a	n/a
	Incorrect pair	n/a	n/a					n/a	n/a	n/a
	Generalised clustering	n/a	n/a					n/a	n/a	n/a
	Lack of clustering	n/a	n/a	x				n/a	n/a	n/a
	Correct pair	n/a	n/a					n/a	n/a	n/a

Table 6.5: *Acoustic and non-acoustic feature sets: Error analysis showing the types of salient errors made, colour coded according to which pair the error pertains. Pairs were deemed separate if they were more than approximately 4-5 nodes apart, or if they had songs from another pair between them. Incorrect pairs were counted for non-matched songs on adjacent nodes, or the same node. Correct pairs were between 0-2 nodes apart.*

	Ebonics off (1037 songs)				Ebonics on (1037 songs)			
Error	NA+A ebonic off Dstress	NA+A ebonic off w/ Rstress	NA+A+IS ebonic off Dstress	NA+A+IS ebonic off w/ Rstress	NA+A ebonic on Dstress	NA+A ebonic on w/ Rstress	NA+A+IS ebonic on Dstress	NA+A+IS ebonic on w/ Rstress
Separated pair								
Incorrect pair								
Generalised clustering			x	x				x
Lack of clustering								
Correct pair								

Table 6.6: *Combination feature sets: Error analysis showing the types of salient errors made, colour coded according to which pair the error pertains.*

The most difficult songs to identify together were the red (positive sentiment), yellow (tongue-in-cheek pop) and turquoise (mainstream rap). Sentiment features were the only class to identify the red pairing, yellow was picked up by both the IS and non-acoustic combined features, and the turquoise pair were assigned neighbouring nodes twice, once by the IS SOM and again by acoustic combined map. This is a positive result considering preliminary analysis of the song pairs (Tables 6.5 and 6.6). Both red and turquoise contained two songs from different genres, whilst the yellow pair were of the same genre but different styles.

The easiest pairs to identify were hardcore rap (royal blue) and the green pair (short and direct). The ease with which rap was picked out is unsurprising since this is a separate genre according to the *All Music Guide*. The green pair is an oddity that requires further investigation since their close proximity may be due to the fact that 'number of x' feature values were not normalised for song length. This was intentional since length was felt to be an important lyrical attribute, however the fact that two of the shorter songs in the corpus were strongly identified together suggests that song length may be overly dominating SOM organisation.

The best performing feature set was non-acoustic combined, *ebonics off*, which identified five out of eight pairs without making any incorrect pairings. The two pairs that were significantly separated, red and turquoise, were the two identified as being an undesirable match in preliminary analysis. The remaining pair that was neither strongly matched nor strongly separated was green, the colour uniquely identifiable by Dstress acoustic features. This level of success on the challenging task of clustering and differentiating songs in the rock/pop genre is a substantial achievement not previously reported. Questions may be raised about the reliability of this result, however, since there was considerable variability when altering the amount of data input to the SOMs (or the selection of artists) and when the same model using half the data with *ebonics on* produced some of the worse clustering.

The most errors were made by SOMs using IS features, and combination of non-acoustic features with other sets, including Dstress and IS, had a detrimental impact on performance. Acoustic pattern features also did not perform as well as hoped, but had the interesting attribute of making fewer errors regarding incorrect pairings than most other classes. The most frequently incorrectly paired songs were magenta (standard pop lyrics), which is sensible as pop music is generally thought to be a melting pot of influences, toned down to the widest possible audience.

6.4 Clustering validity

The pairwise clustering index described in Chapter 4 was applied to assess the quality of test pair matching. The best index scores were achieved by sentiment features (best for *ebonics off*, 1037 songs; S=9.35 E-03), reinforcing the findings from quantisation and topographic error analysis that these features resulted in maps with good test node accuracy and high overall map quality (Figures 6.1 - 6.4). According to visualisations, however, the sentiment maps were suboptimal in song matching and their low clustering index score is probably due to the majority of nodes being members of one cluster, the largest and most uniform cluster from all the maps reviewed.

The second best score was for non-acoustic features, *ebonics off* (1037 songs, S=1.90 E-02; 2392 songs, S=2.20 E-02), the same sets that visual observation indicated made the fewest errors and the most correct song pairings. In contrast to visualisations, however, the clustering index favoured less data, although once again, this may be the result of map topology.

Consistent with earlier observations, the gold standard SOM gave the worst clustering (S=6.90 E-02), followed by the full combination non-acoustic, acoustic and IS features (S=2.38 E-01 to 2.97 E-01) and the IS map alone (S=1.47 E-01). Combination non-acoustic and acoustic feature sets achieved better performance than combination sets including IS, but were not as successful as acoustic features alone (best combination S=3.43 E-02 vs. Dstress S=3.14 E-02). Of the possible acoustic features, the lowest index score was achieved by Dstress features.

The clustering index agreed with visual observation that non-acoustic feature performance deteriorated with ebonics correction, but disagreed with observations about the effect of ebonics on independent language, sentiment and repetition features. Whilst visualisation suggested that *ebonics on* was better for language and sentiment and worse for repetition, the clustering metric indicated the reverse. These three feature classes, when run independently, produced maps that were poorly clustered and it is likely that visual interpretation was not precise. The difference in the two results is suboptimal, however, and should be investigated further with a larger lyric corpus.

Chapter 7

Discussion

Can any single measurement capture music similarity? Previous research into 'ground truth' for music similarity suggests not, trather many truths with respect to different perspectives (Ellis et al., 2002). The gold standard data represents one such perspective for which song-level similarity has been determined using an SOM. Clustering strongly indicates that in the case of playlists, audio features such as timbre, instrumentation and tempo determine similarity, but the implications for music retrieval, recommendation and categorisation are less clear. As discussed in Chapter 1, many factors including topic, genre, lyrical style, audio, vocal and sentiment are important elements of musical description and similarity.

7.1 Information space vs. best combined model

The pairing achieved by non-acoustic combined features was significantly superior to IS in both visual analysis and clustering index score, confirming the hypothesis of this work that for lyrical content of music, combined features relating to a range of linguistic information provide better clustering of song similarity than LSA (IS) alone. In fact, any combination with IS diminished clustering performance as a result of its strong tendency to group the majority of data into one central cluster. Whilst IS was able to distinguish between songs with very different content, it lacked the finesse to also discriminate between songs with less obvious distinctions.

The challenge for IS is that on average, each of the documents for which a vector is constructed is only 200 words long. When documents are this short, important words are very rare, resulting in extremely sparse term counts that make it difficult to distinguish between songs. This sparsity can continue to be an issue even when

data is projected into a lower dimensional space. Additionally, it is possible that the abstractions IS made over word meaning were over-generalised, resulting in a loss of information to the detriment of clustering performance.

To resolve these difficulties, some form of count smoothing could be applied, such as adjusting counts of based on their count in a broad lyric corpus or a collection of the authoring artist's work. A simpler solution would be to use IS or LSA only in situations for which it is naturally suited. Sparse data is less of a problem when clustering songs by an artist, or for a given style or genre, since a single representative vector estimated from from a body of data possesses more information with which to make comparisons. Earlier work by Logan et al. (2004) has shown LSA to effectively identify broad distinctions under circumstances such as these. In contrast, a small data set would have direct and indirect effects on an IS model. Landauer and Dumais (1997) showed that presenting a new paragraph of text to an LSA model approximating the language acquisition of a child in late primary school results in about three times as much indirect knowledge gain about words not in the paragraph as direct knowledge about words that are in the paragraph. It is likely that correct representation of words with IS is based on correct representation of all words, so using a small corpus would be detrimental to overall performance.

In addition to the shortcomings of the data, vectors produced by IS and feature extraction were input to a neural network. The ability of a neural network to model small variations in data is reduced when IS strongly separates some songs from the main group since this causes the network to expend more effort trying to learn values for which the distance is large. Another concern regarding the quality of the SOM output is that without tagged test data there is no way of knowing whether the network is overfitting. Overfitting is a particular concern where there are many weights relative to the number of training cases, as there were for the *ebonics on* and acoustic SOMs.

In comparison, non-acoustic combined *ebonics off* features managed to pair all songs that were from the same genre. Both pairs containing songs of different genres were separated, and a pair with the same genre, but different styles, were separated but not distant. This graded behaviour is very encouraging, especially if it is indicative of performance over the whole data set. If such performance is consistent, these features represent songs in significantly fewer dimensions than IS (15 vs. 75) and could make computation of song similarity in a large lyric database substantially faster whilst at the same time reducing the cost of data storage. Analysis using data in which every song is tagged with genre and style information is warranted to confirm this finding.

7.2 Sentiment categorisation

Short documents could also mislead sentiment analysis by increasing the level of subtlety with which differences between song lyrics are marked. Such an effect would help explain the poor performance of sentiment features on matching up the test song pairs. It might also be the case that the sentiment features used were unable to make crucial distinctions between the type of emotion communicated, and specifically between the activation level associated with negative and positive language. The sentiment SOM identified the brown pair (Alkaline Trio and Jewel) as being more similar to turquoise and light blue (Bruce Springsteen and Crazy Town), than they were to each other. Although the brown pair were selected for sharing a feeling of dark melancholy, the SOM matched this mood most closely with Bruce Springsteen. The match confuses melancholy, or passive negativity, with the anger and outrage of active negativity. A sample of lyrics makes the difference clear. Bruce Springsteen writes:

> *Now Tom said "Mom, wherever there's a cop beatin' a guy*
> *Wherever a hungry newborn baby cries*
> *Where there's a fight 'gainst the blood and hatred in the air*
> *Look for me Mom I'll be there"*

On the other hand, the Alkaline Trio lyric is depressed:

> *Cannot categorize the nature of this sickness*
> *a miracle that you're alive-*
> *Stuck to the roof of my mouth with a staple*

Both moods are negative, but the degree of energy with which they are associated cannot be differentiated by simple strong or weak negativity; a more descriptive framework is required. The obvious choice would be to extend analysis to include Osgood's third dimension of *Activity* to help differentiate the core moods of sadness (passive negative), joy (active positive), contentment (passive positive) and anger (active negative), which informal review of the corpus suggest are frequently found in lyrics.

7.3 Stress pattern performance

Lyrical style referred to in music reviews can be interpreted to include any and all of the non-IS features tried, including those for acoustic repetition. Results do not

strongly support the idea that stress patterns form a signature for differentiating style, but neither is the possibility negated. Acoustic features made the fewest errors in pairing songs, and it may be that the song lyrics simply weren't long enough to form a reliable estimate of stress repetition, particularly since a minimal length was applied in stress pattern selection. Songs with very short lines might not provide sufficient stress repetition information for discrimination.

Alternatively, stress repetition may have been confused with phrasal repetition since it was observed that most of corpus contained repetition in the form of a chorus. Any line of the chorus would also appear as an identical repeating stress pattern, hence acoustic results might be improved by normalising with some measure of phrasal repetition to better encapsulate the global characteristics of heavy and light beats.

The simplistic approach taken in this research of looking up each word in a CELEX-style dictionary also may not correspond to the most pertinent stress information heard by a listener. Table 7.1 shows the difference between two patterns transcribed from the CELEX dictionary and the patterns perceived by this author when listening to the songs with music. Whereas in speech, stress patterns are abbreviated or condensed, CELEX presents a formal pronunciation guide that provides detailed pronunciation for each word. Further, this does not take into account the context of surrounding words.

In the same way that dimensionality reduction in IS enables discernment of topic clusters, condensing stress patterns into fewer stresses seems likely to facilitate detection of pattern similarities. The example in Table 7.1 demonstrates how this could be the case. The Crazy Town lines are combined in the final row to make a four line section comparable to the excerpt from Faithless, and both transcriptions are shown as they are heard by the listener. Both start with a '0 1 0 1' off-on-off-on repeating pattern and double the on-stress near the end of the line. The second line for both lyrics is comparable, but bitwise inverted so that an off-beat in Faithless is an on-beat in Crazy Town. This is an alternating pattern with a double beat two thirds of the way through switching the order of alternation from '0 1 0 1' to '1 0 1 0', or vice versa. Another potentially important similarity is the double on-beat to mark the end of the section.

Observations about inverted patterns and transitions between double and alternating beats represent a fraction of the possibly useful information that could be extracted from stress patterns. Six variations of the CELEX transcriptions were attempted in this study, with the binary pattern proving to be the most informative. There remains much room for further development. For example, many single syllable words are given purely off-beat emphasis (0) in spoken language to maintain a regular rhythm, but in

	Faithless, "Salva mea"	Crazy Town, "Darkside"
Lyric excerpt	I take a look at the world behind these eyes, Every nook, every cranny reorganize, Realize my face don't fit the way I feel Whats real?	Punk rock, shell toes, Horns and halos Wicked white wings and And pointed tails Devil's eyes and nine inch nails Nocturnal renegades
Binary transcription	1 1 0 1 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0 1 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 1 0 1 0 0 0 1 0 0 1 1 1 1 1 1 1 1 0 1	1 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 0
Perceived pattern	0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 1 1	0 1 0 1 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 0 1 1 1 0 1 0 1 0 1 0 0 1 1
Standardised pattern identification	0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 1 1	0 1 0 1 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 0 1 1 1 0 1 0 1 0 1 0 0 1 1

Table 7.1: *Differences between CELEX binary transcription of lyrics and perceived stress patterns for the turquoise test song pair (Crazy Town and Faithless). The bottom row in red shows potentially relevant patterns for extraction.*

the CELEX dictionary they are never given this representation; it is always either a single on-beat (1) or a more complex pattern unlikely to be distinguished by a listener, e.g. ‘1 0 0 0’ for ‘can’. Training a statistical tagger using a corpus tagged with stress patterns may provide a more effective representation, and this is the obvious course for further research.

Nevertheless, it seems clear that if stress patterns provide any discriminating information, it is a very different aspect of language than previously considered. Results with acoustic and non-acoustic combined features suggest acoustic features should not be directly combined with information such as POS. On the other hand, studies in speech recognition suggest that POS can be effectively combined with prosody. More work is needed to understand the relationship between these aspects.

7.4 Combined model performance

Finally, considering the poor performance of the SOMs with combinations of diverse features and the eventual need for combination of any lyrical model with music audio features, some thought should be given as to how this might be achieved. Feature classes could be applied sequentially, first drawing broad distinctions with musical acoustic features and IS, then fine-tuning style boundaries with non-acoustic features. An alternative is a model in which the probability of a data point being assigned to a cluster is determined by each of several techniques whose votes are weighted to produce a final recommendation.

Probabilistic models might be used to simulate individual judgments about music. Every individual has a different criteria for judging lyrical similarity, and particularly when songs are very familiar, they may choose a dimension for judgement that is different to the dimensions which dominate a classifier response. In order to replicate subjective human judgments, weights determining the degree of influence features have upon classification could be tailored to an individual, perhaps using feedback from an online playlist generation system. In this way, music recommendation could be personalised to lyrical, as well as musical, taste.

At the same time, understanding and modeling the relationships of songs lyrics with respect to music as a whole might enable song recall based on descriptions such as ‘like x but more y’. Using a probabilistic model, this could be interpreted as ‘close to x in the most probable clustering, but close to y when clustering using attribute z, where z is the attribute of y with the greatest difference in value between y and the

majority of lyrics'. In other words, a song close to x, but shifted closer to y when projected onto the axis of that best identifies y.

Chapter 8

Conclusion

This thesis has explored the clustering of songs using lyrics features grouped into similar classes and heterogeneous combinations. Simple techniques were used to extract 140 features that were subsequently converted into song vectors and analysed with self-organising maps. Clustering was evaluated using visual analysis and objective measures with reference to eight hand-selected song pairs. Results suggest that for music in the hard-to-differentiate categories of pop, rock and related genres, a combination of features relating to language, grammar, sentiment and repetition improve on the clustering performance of IS with a more accurate analysis of song similarity and increased sensitivity to the nuances of song style. Although results are preliminary and need to be validated with further research on a larger data set, success in differentiating songs in these genres is a substantial achievement and to the knowledge of this author, has not yet been reported in the literature.

Combination of non-acoustic features related to the syntax, appearance and affect of words clustered lyrics better than the same feature classes assessed separately. This suggests the need to replace a modular view of language with a more comprehensive and interactive ideal. On the other hand, combination of more heterogeneous features deteriorated clustering performance, as seen with the combination of non-acoustic features and features relating to acoustic or conceptual properties of language. One theory regarding the cause of this observation is that there are different dimensions along which music similarity may be measured and that these dimensions should be modeled individually. The dimension of style may be better represented by lyrics whilst genre could more closely related to music, however this is conjecture.

It is still not clear how stress patterns of language fit in such a multi-dimensional model of music similarity. Stress patterns made very few errors in clustering lyrics, but

although this result is encouraging, it is clear that if stress is to be successfully applied to language classification then a more sophisticated approach is required. An obvious next step is to investigate what can be achieved with a statistical stress pattern tagger similar to taggers constructed for speech recognition systems. Research in speech processing indicates that a sufficiently accurate and robust stress-dependent language model could offer substantial gains in areas such as summarisation, machine translation and natural language generation.

According to gold standard human-authored playlists, judgments of song similarity are based strongly on music, however this observation may be limited to playlists and is not necessarily extensible to music in the wider domain. In particular, since test song pairs could only be effectively coupled when they were from the same genre, analysis of the correspondence between lyrics and expert human judgments of genre and style may be more fruitful than comparison with similarities observed in playlists. In this regard, the *All Music Guide* used in earlier studies, and also in this research, offers a more stable and flexible standard for comparison of results.

SOM analysis suggests that a few well-chosen attributes may be as good as, if not better than, deep analysis using many features. The specific features used in this study offer no fixed solution, but do indicate promising areas for further research. Results in sentiment analysis using word polarity along the positive/negative and strong/weak axes were not as successful as expected, suggesting that Osgood's third dimension of semantic differentiation is vital to sentiment detection; to differentiate between, for example, anger and depression, the active/passive dimension must be present. On the other hand, results with phrasal repetition are inconclusive. It may be that repetition is best used to differentiate between diverse music genres since most pop is highly repetitive and therefore unsuitable for analysis in this manner.

There are also areas for further development that this research could not address due to the limitations of time and complexity. The distribution of repetition throughout a lyric may be pertinent, including whether repetition is scattered or clustered in the chorus, and how the density of primary and secondary stress changes as a song progresses. In order to model such phenomena, it would be necessary to accurately extract song structure, especially information about the chorus and verse. This might be achieved using an ngram-count approach like the one outlined in this research.

Finally, although this research did not compare clustering achieved with lyrics and published results on similarity obtained using acoustic data, this is a open avenue for future research. There is much work to be done but the possibilities are exciting.

Bibliography

Abney, S. (1996). Part-of-speech tagging and partial parsing. In Church, K., Young, S., and Bloothooft, G., editors, *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers.

Anderson, C. W. and McMaster, G. E. (1982). Computer assisted modeling of affective tone in written documents. *Computers and the Humanities*, 16(1):1–9.

Argamon, S., Bloom, K., Esuli, A., and Sebastiani, F. (2007). Automatically determining attitude type and force for sentiment analysis. In *Proceedings of the 3rd Language and Technology Conference (LTC'07)*.

Arnfield, S. (1996). Word class driven synthesis of prosodic annotations. In *International Conference on Spoken Language, ICSLP 96*, volume 3, pages 1978–1980.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Baumann, S. and Hummel, O. (2005). Enhancing music recommendation algorithms using cultural metadata. *Journal of New Music Research*, 34(2).

Berenzweig, A., Ellis, D. P. W., and Lawrence, S. (2003). Anchor space for classification and similarity measurement of music. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 2 (ICME '03)*, pages 29–32, Washington, DC, USA. IEEE Computer Society.

Berenzweig, A., Logan, B., Ellis, D. P. W., and Whitman, B. P. W. (2004). A large-scale evaluation of acoustic and subjective music-similarity measures. *Comput. Music J.*, 28(2):63–76.

Besson, M., Fata, F., Peretz, I., Bonnel, A.-M., and Requin, J. (1998). Singing in the brain: Independence of lyrics and tunes. *Psychological Science*, 6(9):494–498.

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., and Jurafsky, D. (2004). Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*.

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Boiy, E., Hens, P., Deschacht, K., and Moens, M.-F. (2007). Automatic sentiment analysis in on-line text. In *Proceedings of the 11th International Conference on Electronic Publishing*.

Brochu, E. and de Freitas, N. (2003). "Name that song!": A probabilistic approach to querying on music and text. In *Advances in Neural Information Processing Systems 15*.

Cano, P., Celma, O., Koppenberger, M., and Buld, J. M. (2006). The topology of music recommendation networks. *Chaos*, 16:013107.

Cano, P., Kaltenbrunner, M., Gouyon, F., and Battle, E. (2002). On the use of fastmap for audio retrieval and browsing. In *Proceedings of the International Symposium on Music Information Retrieval*, Paris, France.

Chen, K. and Hasegawa-Johnson, M. (2003). Improving the robustness of prosody dependent language modeling based on prosody syntax dependence. In *Automatic Speech Recognition and Understanding, 2003 (ASRU '03) IEEE Workshop on*, pages 435–440.

Chisholm, E. and Kolda, T. G. (1999). New term weighting formulas for the vector space method in information retrieval. Technical Report ORNL-TM-13756, Oak Ridge National Laboratory, Oak Ridge, TN.

Columbia (2007). Music similarity raw data and statistics. `http://www.ee.columbia.edu/~dpwe/research/musicsim/`.

Curran and Clark (2007). C&C Tagger. `http://svn.ask.it.usyd.edu.au/trac/candc/wiki/MaxEntTaggers`.

Davies, D. L. and Bouldin, W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1(2):224–227.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Dhanaraj, R. and Logan, B. (2005). Automatic prediction of hit songs. In *ISMIR*, pages 488–491.

Dittenbach, M., Rauber, A., and Merkl, D. (2002). Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing*, 48(1-4):199–216.

Ekman, P. (1972). Universal and cultural differences in facial expression of emotion. In Cole, J., editor, *Nebraska Symposium on Motivation*, pages 207–282. Lincoln: University of Nebraska Press.

Ellis, D. P. W., Whitman, B., Berenzweig, A., and Lawrence, S. (2002). The quest for ground truth in musical artist similarity. In *ISMIR 2002*.

Esuli, A. and Sebastiani, F. (2006). SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of LREC-06, 5th Conference on Language Resources and Evaluation*, pages 417–422, Genova, IT.

Fritzke, B. (1995). A growing neural gas network learns topologies. In Tesauro, G., Touretzky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA.

Godbole, N., Srinivasaiah, M., and Skiena, S. (2007). Large-scale sentiment analysis for news and blogs. In *International Conference on Weblogs and Social Media*.

Grover, C. and Tobin, R. (2006). Rule-based chunking and reusability. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference of the European chapter of the Association for Computational Linguistics*, pages 174–181, Morristown, NJ, USA. Association for Computational Linguistics.

Heylighen, F. and Dewaele, J.-M. (2002). Variation in the contextuality of language: An empirical measure. *Foundations of Science*, 7:293–340.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm.

Infomap (2007a). Algorithm description. downloaded from `http://infomap-nlp.sourceforge.net/doc/algorithm.html`.

Infomap (2007b). Natural language processing software. downloaded from `http://infomap-nlp.sourceforge.net/`.

Inquirer (2007). The general inquirer. `http://www.wjh.harvard.edu/~inquirer/3JMoreInfo.html`.

Kamps, J. and Marx, M. (2002). Words with attitude. In *1st International WordNet Conference*, pages 332–341.

Kamps, J., Marx, M., Mokken, R. J., and de Rijke, M. (2004). Using wordnet to measure semantic orientation of adjectives. In *International Conference on Language Resources and Evaluation (LREC)*, volume IV, pages 1115–1118.

Knees, P., Pampalk, E., and Widmer, G. (2004). Artist classification with web-based data. In *ISMIR*.

Knees, P., Pampalk, E., and Widmer, G. (2005a). Automatic classification of musical artists based on web-data. *GAI Journal*, 24(1):16–25.

Knees, P., Pohle, T., Schedl, M., and Widmer, G. (2006a). Automatically describing music on a map. In *Proceedings of the 1st Workshop on Learning the Semantics of Audio Signals (LSAS 2006), 1st International Conference on Semantics and Digital Media Technology (SAMT 2006)*.

Knees, P., Pohle, T., Schedl, M., and Widmer, G. (2006b). Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Multimedia Information Retrieval*, pages 147–154.

Knees, P., Schedl, M., Pohle, T., and Widmer, G. (2006c). An innovative three-dimensional user interface for exploring music collections enriched. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 17–24, New York, NY, USA. ACM Press.

Knees, P., Schedl, M., and Widmer, G. (2005b). Multiple lyrics alignment: Automatic retrieval of song lyrics. In *ISMIR*, pages 564–569.

Kohonen, T. (1982). Self-organized formation of topographically correct feature maps. *Biological Cybernetics*, 43:59–69.

Kurimo, M. (1999). Indexing audio documents by using latent semantic analysis and SOM. In Oja, E. & Kaski, S., editor, *Kohonen Maps*, pages 363–374. Elsevier, Amsterdam.

Landauer, T. and Dumais, S. (1997). A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Li, T. and Ogihara, M. (2004). Music artist style identification by semi-supervised learning from both lyrics and content. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 364–367, New York, NY, USA. ACM Press.

Logan, B. (2004). Music recommendation from song sets. In *ISMIR*.

Logan, B., Kositsky, A., and Moreno, P. (2004). Semantic analysis of song lyrics. In *ICME*, pages 827–830.

Mahedero, J. P. G., Álvaro MartÍnez, Cano, P., Koppenberger, M., and Gouyon, F. (2005). Natural language processing of lyrics. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 475–478, New York, NY, USA. ACM Press.

Martin, J. R. and White, P. R. R. (2005). *The language of evaluation: Appraisal in English*. London, UK: Palgrave.

Neumayer, R., Dittenbach, M., and Rauber, A. (2005). 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. Queen Mary, University of London.

Newby, G. B. (2000). Information space based on html structure. In *TREC 2000*.

Osgood, C. E., Succi, G. J., and Tannenbaum, P. H. (1957). *The Measurement of Meaning*. University of Illinois Press.

Pachet, F., Westermann, G., and Laigre, D. (2001). Musical data mining for electronic music distribution. *Proceedings of 1st International Conference on Web Delivering of Music*.

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Pampalk, E. (2001). Islands of music - analysis, organization, and visualization of music archives. Master's thesis, Vienna University of Technology.

Pampalk, E., Dixon, S., and Widmer, G. (2003). Exploring music collections by browsing different views. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR'03)*, pages 201–208, Baltimore, MD. John Hopkins University.

Pienimaki, A. (2006). Organised evaluation in (music) information retrieval: TREC and MIREX. Downloaded from `www.cs.helsinki.fi/u/linden/teaching/irr06/drafts/anna_pienimaki_mirex.pdf`.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Powers, W. G., Jordan, W. J., and Street, R. L. (1979). Language indices in the measurement of cognitive complexity: Is complexity loquacity? *Human Communication Research*, 6(1):69–73.

Rauber, A. and Fruhwirth, M. (2001). Automatically analyzing and organizing music archives. *Lecture Notes in Computer Science*, 2163:402.

Roiger, A. (2007). Analyzing, labeling, and interacting with soms for knowledge management. Master's thesis, Department of Software Technology and Interactive Systems, Vienna University of Technology.

Rousset, P., Guinot, C., and Maillet, B. (2006). Understanding and reducing variability of som neighbourhood structure. *Neural Networks*, 19(6-7):838–846.

Samsonova, E. V., Kok, J. N., and IJzerman, A. P. (2006). Treesom: Cluster analysis in the self-organizing map. *Neural Networks*, 19(6-7):935–949.

Samuelsson, C. and Voutilainen, A. (1997). Comparing a linguistic and a stochastic tagger. In Cohen, P. R. and Wahlster, W., editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 246–253, Somerset, New Jersey. Association for Computational Linguistics.

Schedl, M., Pohle, T., Knees, P., and Widmer, G. (2006). Assigning and visualizing music genres by web-based co-occurrence analysis. In *ISMIR*, pages 260–265.

Schutze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Scott, S. and Matwin, S. (1998). Text classification using WordNet hypernyms. In Harabagiu, S., editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 38–44. Association for Computational Linguistics, Somerset, New Jersey.

SentimentAI (2007). Yahoo! group. http://tech.groups.yahoo.com/group/SentimentAI/.

Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217.

Shriberg, E., Bates, R., Taylor, P., Stolcke, A., Jurafsky, D., Ries, K., Coccaro, N., Martin, R., Meteer, M., and Van Ess-Dykema, C. (1998). Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3-4):443–492.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

Taboada, M. and Grieve, J. (2004). Analyzing appraisal automatically. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 158–161, Stanford, US.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to data mining*, chapter Cluster Analysis: Basic Concepts and Algorithms, pages 487–559. Addison-Wesley.

Tang, C., Dwarkadas, S., and Xu, Z. (2004). On scaling latent semantic indexing for large peer-to-peer systems. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 112–121, New York, NY, USA. ACM Press.

Taylor, P. (2005). Hidden Markov models for grapheme to phoneme conversion. In *NTERSPEECH-2005*, pages 1973–1976.

Taylor, P. A., Shimodaira, H., Isard, S., King, S., and Kowtko, J. (1996). Using prosodic information to constrain language models for spoken dialogue. In *ICSLP '96*.

Tjong Kim Sang, E. F. and Buchholz, S. (2000). Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132. Lisbon, Portugal.

Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *Computational Linguistics*, 40:417.

Typke, R., Wiering, F., and Veltkamp, R. C. (2005). A survey of music information retrieval systems. In *ISMIR*, pages 153–160.

Vembu, S. and Baumann, S. (2004). A self-organizing map based knowledge discovery for music recommendation systems. In *CMMR*, pages 119–129.

Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. (1999). Self-organizing map in matlab: the som toolbox. In *Proc. of Matlab DSP Conference 1999, Espoo, Finland, November 16–17*, pages 35–40.

Whitman, B. (2005). *Learning the Meaning of Music*. PhD thesis, Massachusetts Institute of Technology, MA, USA.

Whitman, B. and Smaragdis, P. (2002). Combining musical and cultural features for intelligent style detection. In *ISMIR*.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hoffmann, P. (2007). Recognizing contextual polarity: An exploration of features for phrase-level. *Computational Linguistics*, to appear.

Zeimpekis, D. and Gallopoulos, E. (2006). TMG : A MATLAB toolbox for generating term-document matrices from text collections. In Kogan, J., Nicholas, C., and Teboulle, M., editors, *Grouping Multidimensional Data: Recent Advances in Clustering*. Springer Berlin Heidelberg.