

---

Unterschrift des Betreuers

# DIPLOMARBEIT

## Automatic Topic Detection in Song Lyrics

Ausgeführt am Institut für  
Computational Perception  
der Johannes Kepler Universität Linz

unter Anleitung von  
Univ.-Prof. Dipl.-Ing. Dr. Gerhard Widmer

durch

**Florian Kleedorfer**

Röbergasse 14b/9  
A-1090 Wien

---

Ort, Datum

---

Unterschrift (Student)

## Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

## Acknowledgements

Standing on the shoulders of giants isn't easy. It is hard to get there in the first place, and once up there helping hands keep you from falling off.

I am indebted to a number of people for the help they gave me in completing this work.

I want to thank Roman Cerny, Erik Fürst, Dominik Kovacs, David Mann and Seung-Bin Yim for volunteering to take part in the experiments, which were particularly tedious tasks – the promised amount of beer is probably not much of a recompense.

At our institute, Brigitte Krenn was always ready to contribute valuable pieces of her vast knowledge in computer linguistics. Thankfully, she was highly critical of my work from the beginning and remained so throughout my project, which made her an excellent guide and proofreader. I am also indebted to Erich Gstrein, who would always eagerly discuss my work, his own and their common aspects. Similarly I want to thank Peter Hlavac and Roman Cerny for ad-hoc debates about countless little ideas.

Needless to say I depended on my advisor Gerhard Widmer for directions, which he was able to provide with astonishing clairvoyance. The development of my work was scrutinized by Peter Knees, Tim Pohle, Markus Schedl and Klaus Seyerlehner, which I am utterly thankful for. I should pay special tribute to Peter Knees for developing and letting me use the source code of multiple lyrics alignment and to Tim Pohle for discussion and directions on non-negative matrix factorization. I also want to thank Robert Neumayer for valuable discussions on his own lyrics-related research project.

For proofreading and, maybe more importantly, for useful advice in the course of my research and writing, I express my sincere acknowledgements to Stefan Pomajbik, Karin Glaser and Roman Cerny. My mother Elisabeth Kleedorfer, whom I also want to thank for proofreading, takes the full responsibility for the quality of grammar and style of this work.

The work at hand was created at the Research Studio Smart Agent Technologies<sup>1</sup>. Funds came from the Austrian Federal Ministry of Economics and Labor, Impulsprogramm Kreativwirtschaft of Austria Wirtschaftsservice<sup>2</sup> and FIT-IT Semantic Systems project SEMPRES, which is funded by Österreichische Forschungsförderungsgesellschaft<sup>3</sup>.

---

<sup>1</sup><http://sat.researchstudio.at>

<sup>2</sup><http://www.awsg.at>

<sup>3</sup><http://www.ffg.at>

## Zusammenfassung

In der vorliegenden Arbeit wird ein Algorithmus zur automatischen Erkennung von Themen in Liedtexten vorgestellt. Dieser besteht hauptsächlich aus der Anwendung von Methoden des Textmining und dem anschließenden Einsatz von Clustering mittels *non-negative matrix factorization* (NMF). Die dabei entstehenden Cluster werden händisch benannt. Diese Benennung wird in einer kleinen Studie von Versuchspersonen vorgenommen. Die Studie belegt, dass die identifizierten Themen konsistent und erkennbar sind. Durch die Anwendung der präsentierten Methode auf eine Musiksammlung wird ein Information Retrieval System erstellt, das es erlaubt, die Sammlung nach Themen und Themenkombinationen zu durchsuchen.

## Abstract

We propose an algorithm for the automatic detection of topics in song lyrics. It mainly consists in the application of basic text mining techniques on a lyrics collection and clustering the terms found in the lyrics into topics by using *non-negative matrix factorization* (NMF). The resulting clusters are labeled by hand. A small-scale evaluation is used in order to create these labels. The study proves that the identified topics are coherent and recognizable. The result of applying our method to a collection of songs is an information retrieval system which can be queried for topics and topic combinations.

# Contents

1	Introduction	8
2	Related Work	10
2.1	Web Mining and MIR	10
2.2	The Beginning of Lyrics-Related MIR Research	11
2.3	Multi-Modality	12
2.4	NLP and statistical NLP	13
2.5	Lyrics Extraction from the Internet	14
2.6	Influences on Our Work	15
3	Methods	16
3.1	Text Mining	16
3.1.1	The Vector Space Model	16
3.1.2	Term Weighting Methods	16
3.1.3	Foundations of Semantic Analysis	18
3.2	Non-negative Matrix Factorization	20
3.2.1	Algorithms	21
3.3	Multiple Lyrics Alignment	24
3.3.1	Lyrics on the Web	24
3.3.2	Gathering and Preprocessing	25
3.3.3	Alignment	25
3.3.4	Producing the Result	26
3.3.5	Smoothing	26
3.3.6	Confidence Estimation	27
3.4	Lyrics Clipping	27
3.4.1	Algorithm	28
3.4.2	Evaluation	29
4	The Corpus	35
4.1	The Dataset	35
4.2	Crawling	35
4.3	Statistics	36
4.3.1	Genres	36
4.3.2	Languages	36
4.3.3	Length	36
4.3.4	Confidence	38

---

<b>5</b>	<b>Automatic Topic Detection Algorithm</b>	<b>41</b>
5.1	Introduction . . . . .	41
5.2	Overview . . . . .	42
5.3	Preprocessing . . . . .	43
5.4	Automatic Topic Detection . . . . .	43
5.4.1	Term Selection . . . . .	43
5.4.2	Term Weighting . . . . .	44
5.4.3	Clustering . . . . .	44
5.4.4	Labeling . . . . .	45
5.4.5	Calculating Topic Affiliation . . . . .	46
5.5	Query Formulation and Document Retrieval . . . . .	47
<b>6</b>	<b>Assessing the Quality of Identified Topics</b>	<b>48</b>
6.1	Viewing Term Weights . . . . .	48
6.2	Clustering Clusters . . . . .	50
6.3	Searching Songs . . . . .	52
<b>7</b>	<b>Parameter Optimization</b>	<b>55</b>
7.1	Informal Parameter Optimization . . . . .	55
7.1.1	Preprocessing . . . . .	55
7.1.2	Term Selection . . . . .	57
7.1.3	Term Weighting . . . . .	58
7.1.4	Clustering . . . . .	60
7.1.5	Document-Cluster Affiliation . . . . .	62
7.2	Parameter Optimization by Experiment . . . . .	64
7.2.1	Assessment of Individual Cluster Quality by Labeling . . . . .	64
7.2.2	Assessment of the Whole System's Quality by Retrieval . . . . .	68
7.2.3	Consequences . . . . .	68
<b>8</b>	<b>Experiments</b>	<b>70</b>
8.1	Setup . . . . .	70
8.2	Evaluation Measure . . . . .	71
8.3	Results and Discussion . . . . .	72
8.3.1	Relationships among topics . . . . .	74
8.3.2	Relationships among tags . . . . .	77
<b>9</b>	<b>Conclusion and Future Work</b>	<b>81</b>
<b>A</b>	<b>Stopwords</b>	<b>92</b>

CONTENTS	7
B Detailed Experimental Results	93

---

# Chapter 1

## Introduction

---

I remember an interesting coincidence that happened sometime in 2003. While listening to the song “*Common People*” by “*Pulp*”, a song that I had known for years and never liked particularly despite its catchy chorus, I started wondering what it was actually about, as I hadn’t paid attention to the words before. I went online for the lyrics and after reading them along with the music, I was awestruck by its beauty and bitterness – even more so as I had completely ignored this all the times that I had heard the song before. Some weeks later at a party, I recalled the incident and told that story – to see one of my friends start to gape at me and say that he had had the very same experience not long ago: eventually understanding “*Common People*” by reading the lyrics online and completely changing his mind about the song.

Online distribution of music has thrived during the past years and continues to do so. Many portals offer hundreds of thousands or even millions of songs to millions of customers worldwide. It has become increasingly difficult for users to overlook these music collections, which are normally organized in a classical, well-established manner – sorted by genres and popularity. Modern recommendation systems alleviate the difficulties of finding interesting music based upon observing user behaviour. More recent findings from the field of *music information retrieval* (MIR) have, to the best of our knowledge, not been introduced into large commercial systems. However, audio based similarity and recommendation techniques, which have reached a rather mature state, can be expected to play a more important role in commercial systems in the near future.

Off the paths of market-oriented music platforms, a large number of websites allow people to exchange their views about music, and a number of them deal with song lyrics or with discussions about the topics or plots of songs. All lyrics portals rely on users to provide the lyrics and there are at least several hundred thousand lyrics to be found on these platforms. These facts suggest that the number of people interested in the words of the songs they hear is huge. Quite in contrast to this analysis, the dimension of the semantic content of music is completely ignored in commercial platforms and most research prototypes. We believe that sophisticated tools that take the lyrics into account will have the potential to draw a lot of attention to themselves.

In our view, any music browsing or recommendation system is incomplete if it does not incorporate the dimension of the songs' semantic content. It is therefore our goal to create elements of such a system based on the analysis of song lyrics with the long-term objective of introducing it into an existing music browsing tool.

In the work at hand, we present the building blocks of a system that allows for searching a collection of lyrics by selecting from a set of automatically detected topics. We cover the whole process that such a collection has to undergo in order to be indexed by topic, from preprocessing over topic clustering to manual labeling of topics and the calculation of the degree of membership of songs in the detected topics. A small-scale user study emphasizes the recognizability of the topic clusters. Finally, we hint at possible applications of our method to music exploration systems.

The structure of this thesis is explained in the following: Chapter 2 gives an overview of the historical development of the fields of MIR that are concerned with lyrics and Web mining. In Chapter 3, we explain the central techniques we use in our algorithm, such as *non-negative matrix factorization* (NMF). We present our corpus in Chapter 4 with focus on providing statistics that help understanding design choices in later chapters. Chapter 5 is a highly condensed explanation of the whole algorithm, which is necessary in its superficiality for grasping the main idea of the work at hand. The tools we use for inspecting the results of the algorithm are presented in Chapter 6. In Chapter 7 we dilate on the process of optimization of a number of parameters. Experiments for determining the best values for a small parameter set, are accounted for in Chapter 8. Chapter 9 summarizes the thesis and gives an outlook on future work.

# Chapter 2

## Related Work

---

### Abstract

In the field of MIR, research concerning lyrics takes a minuscule space – we found only 16 articles (and 2 master’s theses) on the subject. In order to get an impression of the proportions, this number can be compared to the number of publications from the most important conference in the field, ISMIR<sup>1</sup>, which has produced 653 publications since the year 2000.

In this chapter, we shall give a brief overview of the evolution of this small field, separated by research focus and impact in the Sections 2.1–2.5. In Section 2.6, we summarize the influences of related publications on our work.

### 2.1 Web Mining and MIR

Today, the only way to obtain lyrics for a song in digital form is finding it on one of the many lyrics portals and extracting the relevant text from that Web page. This is probably the reason why lyrics related MIR research developed out of Web mining related MIR research.

To the best of our knowledge, the first MIR publication that uses text extracted from Web pages was published in 1999 (Cohen and Fan, 1999). In the following, the Internet is used as a data source by many researchers in the MIR community. These works can be divided into those which use *structured text* (e.g., cois Pachet et al., 2001; Cohen and Fan, 1999) and those which are based on unstructured sources, i.e., text from Web pages found using crawlers or search engines. In the latter case, the found documents are thought to be an artist’s or track’s cultural context, which is referred to as *cultural metadata* (Whitman and Smaragdis, 2002; Baumann and Hummel, 2003; Baumann and Halloran, 2004) or *community metadata* (Whitman and Lawrence, 2002; Ellis et al., 2002).

The primary use that community metadata is put to in the first years is the computation of similarity measures for artists with the goal of genre or style classification (Whitman and Smaragdis, 2002; Knees et al., 2004; Geleijnse and Korst, 2006b) or recommendation (Baumann and Hummel, 2003;

---

<sup>1</sup><http://www.ismir.net/>

Pohle et al., 2007), or both (Whitman and Lawrence, 2002). In more recent research, music browsing systems have been built based on these methods (e.g., Kleedorfer et al., 2007). Some of those systems are enriched with music-related words from an artist's (or song's) community metadata (Pampalk and Goto, 2007, 2006; Knees et al., 2006).

One of the most closely related publications to ours is not at all concerned with lyrics (Pohle et al., 2007). It explains the detection of artist style from cultural metadata. The downloaded Web content is represented in a *vector space model* (VSM) and then compressed into 8 semantic dimensions using NMF. The resulting information retrieval system can be queried using sliders for weighting each dimension. Results are displayed as the artist pages on the music community portal last.fm<sup>2</sup>.

## 2.2 The Beginning of Lyrics-Related MIR Research

The first instance of computational analysis of song lyrics (Scott and Matwin, 1998) is not concerned with music but with text classification based on WordNet<sup>3</sup>. One of the corpora used to demonstrate the approach just happens to be the DigiTrad database<sup>4</sup>, a database containing the lyrics of 6500 folk songs that are tagged with fixed keywords relating to the topics of the songs. Classification experiments are conducted for two sets of songs, each containing two classes; the attained accuracy for these problems is around 70%.

Until 2002, systems that in some way incorporated song lyrics only do so to add to the searchable content (MacLellan and Boehm, 2000; Frederico, 2002). In one of the most influential works in the field (Baumann and Klüter, 2002), a music retrieval system is presented combining audio based retrieval and *natural language processing* (NLP) methods that act on the lyrics of the songs. In addition to keyword search, the system offers retrieval of similar songs with respect to lyrics. This is done using the cosine similarity measure on a VSM of the lyrics collection. For future work, the authors hint at the possibility of organizing songs in an ontology or taxonomy of topics based on text clustering methods. In another publication (Klüter et al., 2002) this system is described in more detail, focusing on user interface issues, where lyrics are found to be important in a music retrieval context.

---

<sup>2</sup><http://www.last.fm/>

<sup>3</sup><http://wordnet.princeton.edu/>

<sup>4</sup><http://www.deltablues.com/folksearch.html>, as of 1999; cited again in 2007 (Wei et al., 2007). This link is outdated as of 04/2007 and we did not find a new location of the database.

## 2.3 Multi-Modality

The next stage in lyrics-related MIR is the computation of song similarity from a combination of lyrics based and audio based similarity measures. In the first article proposing such a method (Brochu and de Freitas, 2003), a *multi-modal mixture model* is trained in an *expectation maximization* (EM) algorithm. Music and text describing the songs are modeled jointly; in most cases the describing text is the song’s lyrics but the authors are not interested in the particularities of lyrics – they are rather interpreted as a very focused form of textual context of a song. The application scenario aimed for is retrieval, i.e., search for songs. In an extension of this method, the model is adapted to incorporate features extracted from the images of the respective album covers (Brochu et al., 2003). A similar idea is pursued in work aiming at artist style identification (Li and Ogihara, 2004b,a) based on music and lyrics. In this case, an EM algorithm optimizes a model consisting of two independent classifiers (one for audio features, one for text features) which are trained in a bootstrapping process. The results are validated with a set of 43 artists organized in three clusters based on “similar artist” relations obtained from the *all music guide*<sup>5</sup>.

In 2007, experiments on multi-modal genre classification are reported by Neumayer and Rauber (2007a). The applied method is classification using *support vector machines* (SVM) with audio and text features, the text features are the  $TF \times IDF$ <sup>6</sup> weights of the terms in a VSM. Songs are classified with respect to a system of 41 genres. In these experiments, classification accuracy is compared for different feature sets: pure audio features, pure lyrics features, and a combination of audio and lyrics features. Classification with lyrics features reaches an accuracy comparable to that reached on the combined feature set whereas classification with audio features is outperformed. The best accuracy attained is 48.4%, a tremendous advance compared to the 70% accuracy on two-class problems (Scott and Matwin, 1998) and even to 82% accuracy on five genres (Mahedero et al., 2005) (see also Section 2.4). The same authors also document work on multi-modal music clustering using *self-organizing maps* (SOM) (Neumayer and Rauber, 2007b). The songs are clustered separately in audio and text feature spaces. Visualizations of the genre distribution per cluster provide insight into their contiguity. As a quantification of that notion, a quality measure for parallel SOMs is introduced. The two resulting maps are visualized together in a browsable user interface using *smoothed data histograms* (SDH). Very closely related work

<sup>5</sup><http://www.allmusic.com>, as of 04/2007

<sup>6</sup>For an explanation of  $TF \times IDF$ , see Section 3.1.

is covered in greater detail by Neumayer (2007).

## 2.4 NLP and statistical NLP

The first publication that focuses solely on lyrics from a MIR point of view, and probably therefore one of the most frequently cited by other lyrics-related articles is published in 2004 (Logan et al., 2004). In this work, lyrics are analyzed using *probabilistic latent semantic analysis* (PLSA), a technique that represents each text document as a combination of topics (Hofmann, 1999). Similar songs are found using the L1 distance between these representation vectors. User-created artist similarity data (Ellis et al., 2002) is used as ground truth, which forces the authors to define an artist similarity measure grounded in lyrics. The results are compared to audio-based methods, which are found to perform better. However, the lyrics-based methods make different mistakes and it is therefore argued that the techniques can be combined profitably.

A quite influential publication from 2005 (Mahedero et al., 2005) shows the application of different NLP methods to lyrics. Four problems and methods for solving them are proposed: *Language Identification*, *Structure Extraction*, *Thematic Categorization* and *Similarity Search*. Language and nationality are shown to be important features by which people search for music (Bainbridge et al., 2003), so, it is argued, it makes sense to derive this kind of information from the lyrics. Results are quite promising (over 92% accuracy with 500 songs in 5 languages) thanks to the state of language identification technology. Structure extraction is concerned with automatically identifying the segments intro, verse, chorus, bridge and outro in the lyrics. This information can be used for bootstrapping audio segmentation algorithms. For thematic categorization, the Naive Bayes classifier was trained on 125 songs that were assigned to 5 topics by hand and achieved an accuracy of 82%. The authors state that the definition of the topics and the assignment of lyrics to them are problematic and at the same time crucial to categorization performance. Similarity search is done using the cosine measure in a VSM with  $TF \times IDF$  weighting and evaluated for the detection of cover songs and different versions of the same song.

In another publication from the same year, it is attempted to predict if a song is a hit song using SVM. The classifier is trained with audio features, lyrics features (derived via PLSA) and a combination of both. Comparison shows that the combination yields the best performance (assessed using *receiver operating characteristic* (ROC) analysis). In this article, problems concerning the transcription styles of lyrics are found on the Internet are

reported.

In two articles, the pragmatic aspects of music recommendation systems are analyzed (Baumann et al., 2004; Baumann and Halloran, 2004) in user studies. The similarity measure used for recommendations is again multi-modal, consisting of three components: an audio-based one, a similarity measure computed from cultural metadata and one derived from lyrics. The lyrics-based measure is computed as the cosine between the lyrics representations in a VSM with  $TF \times IDF$  weighting. Users can modify the relative weight of each measure to influence the recommendations they get. Their behaviour is recorded in different environments (in the lab and in “the wild”). Evaluations suggest that such multi-modality makes sense to accommodate varying moods. Recommendations based on lyrics are especially useful in situations where users want new and unexpected music.

At ISMIR 2007, Wei et al. (2007) demonstrate how keywords can be generated for songs based on lyrics. The proposed algorithm uses a complex combination of NLP methods, clustering and usage of WordNet. Its performance is compared to the results reported by Scott and Matwin (1998) (see also Section 2.2) and found to be superior.

Penaranda (2007) describes experiments with genre classification of lyrics using SVMs. The original contribution of this work is the blend of features on which the classifier is trained: terms, rhyme features, text-statistic features and *part-of-speech* (POS) features. The method is evaluated on a set of more than 7000 lyrics from 37 different genres, where accuracy of over 65% is reported.

## 2.5 Lyrics Extraction from the Internet

In 2005, the extraction of lyrics from the Internet is recognized as a MIR problem in its own right, worthy of academic research (Knees et al., 2005). The idea of the authors is to download multiple Web pages containing the lyrics to the same song and assembling a version of the lyrics that most sources agree upon, thus eliminating small spelling mistakes or errors caused by mishearing the words. A different approach to solving a similar problem was published shortly after (Geleijnse and Korst, 2006a). Its main contribution was the extraction of the lyrics from the rest of the content of the respective Web page, which was possible because the specific typical paragraph structure of lyrics was taken into account. The same authors conflated the two methods into a technique that can now be considered state of the art (Korst and Geleijnse, 2006).

## 2.6 Influences on Our Work

The governing idea behind the work at hand, providing the tools for the organization of songs by their topics, is not new at all. It is at least latent in two of the above mentioned publications (Logan et al., 2004; Dhanaraj and Logan, 2005)<sup>7</sup>, envisioned as future work in one (Baumann and Klüter, 2002) and actually tackled on a very small scale in one (Mahedero et al., 2005). The reason why there has not been any attempt at automatic topic detection on a song set of more realistic size is presumably the absence of adequate ground truth. We circumvent the problem by applying a method that produces topic models – and by evaluating their quality instead of the association of songs to topics.

As to the methods we apply, we were greatly influenced by the paper by Logan et al. (Logan et al., 2004), and as a consequence we experimented with *latent semantic analysis* (LSA) (Deerwester et al., 1990). We discovered through the publication by Pohle et al. (2007) that NMF suited our needs better due to its relatively small space complexity (which was the bottleneck of our LSA implementation).

We draw much upon the work on lyrics extraction by Knees et al. (Knees et al., 2005) for the preparation of our dataset, as we were in the position to use and adapt their software to suit our needs.

## Summary

We give an account of the evolution of lyrics-related MIR research in this chapter. We trace the roots of the first works actually mentioning lyrics back to the beginnings of Web mining-related research and maintain that there is a co-evolution of these two fields, which we exemplify citing a number of papers from both currents. We divide academic research into the two categories *multi-modality* and *NLP*, which in our view structures the field appropriately. We also dilate on a research topic that is (or is apt to be) fundamental to all research in the field, namely the extraction of high-quality lyrics from the Internet. Finally, we reveal our sources of inspiration for the methods and the research goals of the work at hand.

---

<sup>7</sup>Due to the use of PLSA for dimensionality reduction and semantic description, the authors actually mention *topics* found in the lyrics corpus

# Chapter 3

## Methods

---

### Abstract

This chapter covers the most important methods we use in the proposed algorithm. These include text mining in Section 3.1, matrix factorization in Section 3.2 and the extraction of lyrics from the Internet in Section 3.3. That last method is modified by us in order to improve results, and we explain and validate our modification in Section 3.4.

### 3.1 Text Mining

Text mining is a discipline of computer science that is concerned with extracting or manipulating information in text documents. In this field, methods have been developed for the solution of a wide variety of tasks, and the proper formalization of text depends on the task that is to be solved.

For our purposes, the most fundamental element of text is a *word form* or *term*. One or more such terms form a *sentence*, and a sequence of one or more sentences is a *document*. A collection of documents is called a *corpus*.

#### 3.1.1 The Vector Space Model

For many text mining tasks, the appropriate representation of a corpus is the *vector space model* (VSM) (Baeza-Yates and Ribeiro-Neto, 1999). In this model, the documents are represented as vectors containing the weight of each term. Consequently, the corpus takes the form of a large, sparse matrix containing one row per document and one column per term, the value  $t_{ij}$  is the weight of the  $j$ -th term in the  $i$ -th document. This matrix is called *term-document matrix* (TDM); in formal definitions it will be denoted  $T$ . Its values are non-negative integers, i.e.,  $t_{ij} \in \mathbb{N} \cup \{0\}$ . By default the term-document matrix  $T$  contains the term frequency of each term for each document.

#### 3.1.2 Term Weighting Methods

The weights of terms in a document description can be assigned following a number of methods, and though some methods are much more widely used

than others, the usefulness of the *weighting function* depends on the task at hand. Mathematically speaking, the weighting function is a matrix function

$$f : \mathbb{R}^{+,m \times n} \mapsto \mathbb{R}^{m \times n} \quad (3.1)$$

that maps a  $m \times n$  matrix to a matrix of the same dimensionality. In this work, we will mention three weighting schemes, which are briefly described here.

### Term Frequency Weighting

This weighting is simply the frequency of each term in each document. The function that performs this mapping is the identical function, i.e.,

$$f_{TF}(T) = T \quad (3.2)$$

Weighting terms this way incurs a bias toward long documents for many applications (for example, retrieval) simply because the cumulative weight of all terms of a document is equal to the word count of a document, and there is no normalization that downweights terms in long documents. For example, the only occurrence of the term “*heart*” in a document 100 words long gets the same weight as the only occurrence of that term in a document of 15000 words. In a retrieval scenario, a search for “*heart*”, both documents would be attributed equal relevance with respect to the query.

### Binary Weighting

Centrally to the *Boolean Model* (Baeza-Yates and Ribeiro-Neto, 1999) of an IR system but useful in other scenarios as well, this weighting function assigns a value of 1 to all terms occurring in a document and 0 to all others. This facilitates boolean queries in a retrieval use case, e.g., “Find all documents that contain “*bank*” and “*money*” and not “*river*””. The bias toward long documents, as described for term frequency weighting, is also brought about by the binary weighting method. The formal definition of this weighting function is

$$f_B(T) = \text{sgn}(T) \quad (3.3)$$

where  $\text{sgn}(A)$  is the signum function applied to each element of the matrix  $A$ . As in our case, the matrix  $A$  is non-negative (the TDM does not contain negative values), the result contains only 0’s and 1’s.

### TF×IDF

This weighting method is probably the most widely applied one in information retrieval systems. Its main idea is to normalize term frequencies in two ways, resulting in term weights which are not biased by document length and which capture the importance of terms in the whole corpus. This latter part is realized via multiplication of the term frequency (TF) with the inverse document frequency (IDF) of the term, i.e., the inverse of the number of documents the term occurs in. The exact definition of TF × IDF differs slightly from author to author; we apply the following function a term-document matrix  $T$  (Salton and Buckley, 1988; Xu et al., 2003).

$$f_{\text{TF} \times \text{IDF}}(T) = \left[ t_{ij} * \log \left( \frac{\sum_k \text{sgn}(t_{ik})}{\sum_k \text{sgn}(t_{kj})} \right) \right]_{ij} \quad (3.4)$$

### 3.1.3 Foundations of Semantic Analysis

The knowledge represented in a text can be extracted in an automated or semi-automated manner based on the structural relations of the words (terms), sentences and documents. Linguistic Structuralism is concerned with such relations between signs. This field of linguistics provides tools for automated text analysis. Two kinds of such relations, *syntagmatic* and *paradigmatic* relations, are explained and formally defined in the rest of this chapter.<sup>1</sup>

#### Syntagmatic Relations

Two word forms are in a syntagmatic relation if they occur together. If they occur together more often than they would by accident, they are in a statistic-syntagmatic relation.

Let the language  $L = (W, S)$ , where  $W$  is the set of different word forms and  $S$  is the set of sensible and correct sentences. In the following,  $w$  is an element of  $W$  and  $s$  an element of  $S$ .

$S$  is a multiset of sets  $s$  (the sentences), which consist of elements  $w_1, \dots, w_n$ , which are the different words in the sentence.  $s$  is unordered.

**Definition 1** *The local context  $K_s(w)$  of a word form  $w$  is the set of word forms that occur with  $w$  in a sentence  $s$ :*

$$K_s(w) = s \setminus \{w\}, w \in s \quad (3.5)$$

<sup>1</sup>This section is a summarization of (Heyer et al., 2006), Chapter 2.1 and 2.2.

**Definition 2** *Two word forms  $w_i, w_j$  in  $W$  are in syntagmatic relation iff there is at least one local context containing both word forms:*

$$\text{SYN}(w_i, w_j) \iff \exists s : w_j \in K_s(w_i) \quad (3.6)$$

The appearance of two word forms in one context is called co-occurrence.

**Definition 3** *Two word forms are in statistic-syntagmatic relation if they are in a syntagmatic relation and their co-occurrences are not random with respect to a suitable significance measure.*

The appearance of two word forms in a statistic-syntagmatic relation is called *significant co-occurrence*.

There are several groups of statistic-syntagmatic relations of special importance:

- neighbouring co-occurrences (right and left neighbours)
- co-occurrences in a fixed window of  $N$  word forms back and ahead.

### Paradigmatic Relations

Paradigmatic relations are defined by words that appear in similar contexts.

**Definition 4** *The global context  $K_G(w_i)$  of a word form  $w_i \in W$  is the set of all word forms  $w \in W$  with  $\text{SYNS}(w, w_i)$ :*

$$K_G(w_i) = \{w \mid \text{SYNS}(w, w_i)\} \quad (3.7)$$

In other words, the global context of a word form  $w$  contains all word forms that co-occur with  $w$  more frequently than at random.

Using a predicate for comparison, two global contexts can be compared.

**Definition 5** *If two global contexts are similar with respect to a threshold value  $t$ , this is expressed as*

$$\text{SIM}_t(K_G(w_i), K_G(w_j)) \quad (3.8)$$

The more similar the global contexts of two word forms are, the more similar is their use.

**Definition 6** *Two word forms  $w_i, w_j \in W$  are in paradigmatic relation  $\text{PARA}(w_i, w_j)$ , iff their global contexts are similar with respect to a similarity measure with fixed threshold.*

$$\text{PARA}(w_i, w_j) \iff \text{SIM}_t(K_G(w_i), K_G(w_j)) \quad (3.9)$$

## 3.2 Non-negative Matrix Factorization

NMF is a method that allows for clustering objects that are represented in a matrix. This is done by calculating an approximation of the matrix by the product of two smaller matrices. The factor matrices contain the representation of parts of the objects together with the information of how they are combined to form the whole objects (which are described by the original matrix). Clustering can be done by grouping the objects with similar parts together (Paatero and Tapper, 1994; Lee and Seung, 1999).

Formally speaking, let  $V_{m \times n}$  be the original matrix and  $W_{m \times k}$  and  $H_{k \times n}$  be the factor matrices, values of the latter two are found such that

$$V \approx WH \quad (3.10)$$

and

$$\forall i \in \{0, \dots, m\}, j \in \{0, \dots, k\} : w_{ij} \geq 0 \quad (3.11)$$

$$\forall i \in \{0, \dots, k\}, j \in \{0, \dots, n\} : h_{ij} \geq 0 \quad (3.12)$$

The *inner dimension*  $k$  is the crucial parameter for this method. It determines the width of  $W$  and the height of  $H$  and thus limits the number of different “parts of objects” (semantic dimensions, object features, topics, or whatever term may fit depending on the application scenario) that can be found by the algorithm.

Because of the non-negativity constraint on  $W$  and  $H$  (see Equation 3.11), the objects in the original matrix are represented as additive combinations of common features. This quality separates NMF from the matrix factorization methods *vector quantization* (VQ) and *principal components analysis* (PCA), which both learn holistic, not parts-based representations. From the perspective of cognitive science, this is an interesting feature because evidence exists that there are parts-based representations in the human brain (Lee and Seung, 1999). As a result, the object features found by NMF are quite intelligible for humans. For an illustration of that point let’s assume that NMF is used to process a database of pictures of human faces. Each face is then represented as a combination of face parts, which may remind one of the way phantom pictures are produced. This *additive* method is actually the way humans tend to describe such pictures, e.g., “Moustache, dark eyes, bald head, big ears, ...”. In contrast to this way of describing faces, it is quite counter-intuitive to do the same in a subtractive manner.

### 3.2.1 Algorithms

For the work at hand, we had to implement NMF as no such code could be found for our development environment, R (R Development Core Team, 2006). We confined ourselves to quite basic realizations of two different algorithms (Berry et al., 2007), *multiplicative update* (MU) and *alternating least squares* (ALS). With both methods we calculate the factor matrices  $W$  and  $H$  with the goal of minimizing the distance between  $V$  and  $WH$  in the Frobenius norm, i.e.,

$$\frac{1}{2}\|V - WH\|_F^2 \rightarrow \min \quad (3.13)$$

Both algorithms take the parameters  $V, k$  and *maxiter*, where  $V$  is the term-document matrix,  $k$  is the inner dimension of the factor matrices and *maxiter* is the number of iterations after which the algorithms stop.

In recent years, many refinements to these methods have been proposed, none of which we implemented as the observed performance of ALS was satisfying for our application. We acknowledge, however, that further work in this field toward the state of the art may yet improve the quality of our topic clusters.

#### Multiplicative Update

This method<sup>2</sup>, shown in Algorithm 1<sup>3</sup>, computes NMF by updating the elements in the randomly initialized matrices  $W$  and  $H$  proportionally to the difference of the respective elements in  $V$  and  $WH$  ppa:nmfAlgorithmsProof. From our experience, we can confirm that this algorithm is quite slow and does not always converge, as discussed by Berry et al. (2007).

```

Data:  $V_{m \times n}$ ,  $k$ , maxiter
Result:  $W, H$ 
 $W \leftarrow \text{rand}(m, k)$  ; % initialize W with positive random numbers
 $H \leftarrow \text{rand}(k, n)$  ; % initialize H with positive random numbers
for  $i \leftarrow 1$  to maxiter do
  |  $H \leftarrow H \cdot (W^T V) ./ (W^T W H + 10^{-9})$  ;
  |  $W \leftarrow W \cdot (V H^T) ./ (W H H^T + 10^{-9})$  ;
end

```

**Algorithm 1:** Multiplicative update algorithm computing NMF

<sup>2</sup>cp. Berry et al. (2007), Section 3.1.

<sup>3</sup>In this listing we use the operators  $\cdot$  and  $./$  for element-wise matrix multiplication and division, respectively.

### Alternating Least Squares

This method<sup>4</sup>, shown in Algorithm 2<sup>5</sup>, starts out with random initialization of the matrix  $W$ . Then the equation  $V = WH$  is solved alternately for  $H$  and  $W$ , each time using a least squares method, solving a convex optimization problem. To that end  $V = WH$  is transformed into a matrix equation of the form  $AX = B$  that can be solved for  $X$  by existing algorithms if  $A$  is square and  $B$  is compatible with  $A$  in that it has the same number of rows. The transformation requires all three matrices to be invertible. For computing  $H$ , it is fairly simple:

$$\begin{aligned} WH &= V && ./W^T. \\ W^TWH &= W^TV \end{aligned}$$

For optimizing  $W$ , the transformation is more complicated:

$$\begin{aligned} V &= WH && ./V^T \\ VV^T &= WHV^T \\ V(WH)^T &= WHV^T \\ VH^TW^T &= WHV^T \\ W(HH^TW^T) &= W(HV^T) \\ HH^TW^T &= HV^T \end{aligned}$$

The process is iterated a specified number of times.

The ALS algorithm proved highly useful as it was fast and certain to converge.

---

<sup>4</sup>cp. Berry et al. (2007), Section 3.3.

<sup>5</sup>In this listing we use the operators `.*` and `./` for element-wise matrix multiplication and division, respectively.

```
Data:  $V_{m \times n}$ ,  $k$ , maxiter  
Result:  $W$ ,  $H$   
 $W \leftarrow \text{rand}(m, k)$  ; % initialize  $W$  with positive random numbers  
for  $i \leftarrow 1$  to maxiter do  
    Solve for  $H$  in matrix equation  $W^T W H = W^T V$  ;  
    Set all negative elements in  $H$  to 0 ;  
    Solve for  $W$  in matrix equation  $H H^T W^T = H V^T$  ;  
    Set all negative elements in  $W$  to 0 ;  
end
```

**Algorithm 2:** Alternating least squares algorithm computing NMF

### 3.3 Multiple Lyrics Alignment

At the beginning of our research we were concerned with building a corpus of lyrics based upon a collection of songs of which we had the music and metadata. Consequently, the arising challenge consisted in extracting specific lyrics from the web. This non-trivial task can be performed using different approaches, all of which include an online search for lyrics, the extraction of the relevant text from the Web pages, and handling of different versions of the lyrics to the same song. Thanks to the cooperation between our research group and the Institute of Computational Perception of the Johannes Kepler Universität Linz, we were in the position to use Peter Knees' implementation of *multiple lyrics alignment* (MLA), explained in the work by Knees et al. (2005); Knees (2008). This technique handles Web search and production of the most probable version of the lyrics to a specific song and computes a *confidence estimation* that indicates the quality of the result. It is a combination of Web crawling, rule-based lyrics preprocessing and, at the very core of the approach, alignment of the text found in the downloaded Web pages. The latter is achieved using an algorithm for Multiple Sequence Alignment, a technique that originated from bioinformatics, where it is used to align DNA and protein sequences.

#### 3.3.1 Lyrics on the Web

If the ultimate goal is obtaining a plausible version of the lyrics for a given song based on different versions obtained from the Internet, it is of great interest to know in which ways lyrics tend to differ from each other. Several such dimensions are listed by Knees et al. (2005, Section 3):

- *Different spellings* of words.
- *Differences in the semantic content* mostly by misheard words.
- *Different versions* of songs.
- *Annotation of background voices, spoken text and sounds.*
- *Annotation of performing artist*, especially in duets.
- *Annotation of chorus or verses* to avoid duplication.
- *References and abbreviations of repetitions.*
- *Inconsistent structuring.*

In addition to these differences between versions of lyrics found on the Internet, some lyrics portals use watermarking techniques. This suggests

that portals use crawling to retrieve lyrics from each other, a hypothesis which is backed by our observation that in many cases, lyrics from different sources match exactly, including annotation style. Some portals put a visible link to their website between two stanzas<sup>6</sup>, others interrupt the lyrics for advertisements<sup>7</sup>, yet others insert a line in nearly invisible font size (but not colour, interestingly), stating the name of the portal<sup>8</sup>. In one portal<sup>9</sup>, most lines of the lyrics contained invisible additional text that was obviously inserted at random and in most cases read *[song title] [artist name] lyrics*. Sometimes one or two of the elements of that text were missing.

### 3.3.2 Gathering and Preprocessing

Web search engines are used to obtain Web pages containing the lyrics. For Web search, queries of the form “*artist name*” “*track name*” *lyrics* are used. Web pages are stripped of HTML markup and converted to lower case.

Next, downloaded pages are transformed by a rule-based algorithm that replaces references to the chorus with the actual text of the chorus. This step is referred to as *expansion*. There are different styles of writing the chorus such as writing it fully each time as opposed to annotating the chorus with “*chorus*” the first time it appears and afterwards referring to it (e.g., using “*chorus x2*”). As such differences pose a problem for the subsequent *alignment* step, they are eliminated beforehand by expanding them to the actual chorus.

### 3.3.3 Alignment

An alignment of two sequences is essentially created by placing them next to each other and inserting gaps in order to maximize the number of identical elements at the same position. The *Needleman-Wunsch Algorithm* (Needleman and Wunsch, 1970) is a technique commonly used to solve this problem. It produces a globally optimal alignment for two input sequences; while the principle of the algorithm is extensible to any number of sequences, its computational complexity hinders its applicability in practice. To overcome this problem, alignments are computed between pairs of texts, then the results

---

<sup>6</sup>e.g., <http://www.metrolyrics.com>, as of 2007/04/17

<sup>7</sup>e.g., <http://www.lyricsandsongs.com>, <http://www.lyricsdepot.com>, as of 2007/04/17

<sup>8</sup>e.g., <http://www.completealbumlyrics.com/>, as of 2007/04/17

<sup>9</sup>Unfortunately, we must admit that we found out about that watermarking technique a while after we did the crawling, and we could not find out which portal the pages were downloaded from.

are aligned pairwise again, and this is done recursively until all texts have been combined hierarchically (Corpet, 1988).

The *Needleman-Wunsch Algorithm* is parametrized by a gap penalty and a similarity matrix for the alphabet that the sequences use. MLA uses a gap penalty of  $-1$ , and a similarity matrix is defined by attributing a value of 10 for aligning two exactly matching words and 0 if two aligned words differ. Thus, the algorithm tends to align mismatching words rather than to insert gaps. Table 3.1<sup>10</sup> shows such an alignment.

### 3.3.4 Producing the Result

For each column in the alignment, the most common word  $w$  is chosen for the result. In order to make sure that  $w$  is correct with some certainty,  $w$  is only accepted if the ratio of its occurrence count to the number of aligned rows is larger than the threshold parameter  $t$ .

Pages not containing the lyrics at all are filtered from the final result by computation of a preliminary alignment using  $t=0.3$ . All sequences that have less than 33% accordance with the preliminary alignment are removed. The remaining sequences are used to produce the final alignment. The impact of  $t$  is shown in Table 3.1.

... it's showtime	-	for dry climes and bedlam is dreaming of rain when the hills	...
... it's show time		for dry climes and bedlam is dreaming of rain when the hills	...
... it's showtime	-	for dry climes and bedlam is dreaming of rain when the hills	...
... it's showtime	-	for drag lines and bedlam is dreaming of rain when the hills	...
... it's showtime	-	for dry climes and bedlam is dreaming of rain when the hills	...

Table 3.1: Section from the alignment of the song “Los Angeles is Burning” by “Bad Religion”. The four rows on the top are word sequences extracted from the web, the row at the bottom is the result obtained with any threshold value  $t$  below 0.75.

### 3.3.5 Smoothing

The results produced using the algorithm as it has been described so far often contains words that do not belong to the lyrics at the beginning and at the end. This is due to the fact that in most cases, the lyrics are preceded by the name of the artist and the title of the song; similarly, lyrics are commonly followed by copyright notices of some form. While some of those words coincide and are thus aligned, most do not, which is why there tends to be a number

<sup>10</sup>reproduction of Figure 1 by Knees et al. (2005)

of gaps around the aligned terms in the final alignment. Consequently, the resulting alignment is post-processed analyzing the coherence of the output sequence:

*For every word in the output all rows agreeing with the output on this word are consulted. The number of agreements of the five preceding words and of the five subsequent words with the output is summed up. If this sum is below 35% of the number of totally examined words, the word is removed from the output. Although this procedure removes unrelated words in most cases, it is also at risk to remove words at the end or the beginning of coherent sequences, especially if agreement among the different pages is low.<sup>11</sup>*

### 3.3.6 Confidence Estimation

For the final result, which is available at this stage of the process, a *confidence estimation* value is calculated, giving information on the quality of the output. *This value is a heuristic that incorporates both the certainty of decision for the single words and the coherence of the output string.*<sup>12</sup> It can be used as a filter criterion in subsequent processing steps with the effect of ensuring a minimal quality of the lyrics. The mathematical definition of the certainty value is:

$$\text{conf}(l) = \frac{\text{len}(\tilde{l})}{\text{len}(l)} \cdot \frac{1}{\text{len}(l)} \sum_{i=1}^{\text{len}(l)} \text{cert}(l_i) \quad (3.14)$$

where  $l$  is an alignment,  $\tilde{l}$  its smoothed version,  $l_i$  the  $i^{\text{th}}$  word in  $l$ ,  $\text{len}(x)$  the length of  $x$ ,

$$\text{cert}(x) = \frac{\text{maxword}(x)}{\text{depth}(x)}, \quad (3.15)$$

$\text{depth}(x)$  the number of rows in the alignment  $x$  and  $\text{maxword}(x)$  the number of occurrences of the most frequent word in  $x$ .

## 3.4 Lyrics Clipping

We modified MLA in order to overcome a shortcoming of the approach. The quality of the lyrics obtained through MLA was sometimes lowered by

<sup>11</sup>cp. Knees et al. (2005, Section 4.3.1)

<sup>12</sup>cp. Knees (2008, Section 2)

artifacts like the artist’s name, the song title or the word “*lyrics*” at the beginning of the text as well as due to trailing text containing copyright notices or remarks like “*lyrics added by . . .*”. We found that these problems could be mitigated by combining MLA with a different approach to lyrics extraction from the Web (Geleijnse and Korst, 2006a). Here, regular expressions are used for determining which part of a Web page contains the lyrics. The algorithm they describe for clipping the lyrics was added to MLA as an additional preprocessing stage.

One of the shortcomings of the technique of Multiple Lyrics Alignment is the presence of artifacts from the webpage containing the lyrics, which are found at the beginning and at the end of the final output of MLA. Smoothing is applied in order to remove these spurious words, which works in many cases. However, a side effect of this method is the risk of removing other possibly important words.

By providing another means for identifying the lyrics portion inside the Web pages, a preprocessing stage that performs this task can be prepended to MLA, which allows for leaving out the smoothing step of MLA, thus avoiding its side effects. We call this additional preprocessing step *Lyrics Clipping* (LC).

### 3.4.1 Algorithm

The technique we use for solving this problem has been developed by Geleijnse and Korst (2006a). They propose a simple heuristic approach for finding exclusively the lyrics in a Web page. The central idea of the algorithm is to make use of the structure of lyrics and to identify a coherent part of a Web page that has this structure. Lyrics consist of stanzas separated by blank lines, each stanza consists of one or more lines. The authors state the crucial observation that lyrics are void of html markup other than “`<br>`” (or “`<br />`”), which allows for a projection of the lines of a Web page (containing lyrics) to a string of the form  $(l|b|n)^+$  according to the following scheme:

- b** if the substring is empty or the string only consists of white space characters (blank).
- l** if the substring does not contain any html-tags and contains between 3 and 80 characters (lyrics line).
- n** otherwise (non lyrics).

A regular expression can then be used to extract a long enough portion of the projection that contains only “*l*” and “*b*” characters. The corresponding substring in the Web page’s html code is extracted as the song’s lyrics.

During exploration of the Web content we found the observation stated by Geleijnse and Korst (2006a) concerning the uniform layout of lyrics on Web pages to be correct in most cases. In other words, the most common form of Web lyrics is plain text with “`<br>`” tags at the end of each line. However, a considerable amount of pages does not fit into that scheme. Some actually use html markup within lyrics for annotations (e.g. “`<i>[Chorus]</i>`”) or for visually emphasizing the whole chorus. On other pages, the “`<br>`” tags are at the beginning of each source code line, not at the end. Yet on other pages, the entire lyrics are placed inside the body of one “`<pre>`” element, so there are no “`<br>`” tags at all. Sometimes, watermarking techniques use html markup, for example, a line containing the name of the lyrics portal in a very small font is inserted into the lyrics.<sup>13</sup>

Taking our findings into account, the algorithm for lyrics identification was slightly adapted: As a preprocessing step, the html source code is stripped of opening and closing tags that only format text.<sup>14</sup> Next, we insert a newline (“`\n`”) character before and after each html tag. Any line of the resulting html source code contains either zero or more whitespace characters, exclusively plain text, or exactly one html tag.

The projection of the html source code on the **bnl**-representation has to be changed slightly: the character *b* is inserted into the string for each line containing a “`<br>`” tag along with an arbitrary amount of white space.

The **bnl**-representation is matched against the regular expression

$$R = l \cdot (b|l)^* \cdot l \quad (3.16)$$

and the longest match is interpreted to represent the lyrics. The corresponding portion of the html source code is extracted.

### 3.4.2 Evaluation

#### Measurement

The quality of LC is measured by comparing automatically clipped lyrics to manually extracted ones. The quality measure used here is the one developed by Knees (2008). Precision and recall values are defined for a text with respect to the *correct* version of the same text using the alignment of the

<sup>13</sup>e.g., see <http://www.completealbumlyrics.com>, as of 2007/04/02

<sup>14</sup>We removed “`<b>`”, “`<p>`”, “`<i>`”, “`<u>`”, “`<pre>`”, “`<font>`”, “`<span>`”, “`<emph>`”, “`<strong>`” and “`<bold>`” tags.

two<sup>15</sup> that is computed with very high penalty values for aligning different words at the same position and very low penalty values for aligning a word with a gap. This parametrization leads to alignments in which columns contain either two matching words or a word and a gap.

Precision measures how well the algorithm performs in not adding non relevant words; recall measures how well it performs in adding the relevant words. Both values can be derived from the number of gaps in the aligned result as follows:

$$Prec = 1 - \frac{|\text{gaps}_{\text{manual}}|}{\text{len}} \quad (3.17)$$

$$Rec = 1 - \frac{|\text{gaps}_{\text{automatic}}|}{\text{len}} \quad (3.18)$$

where  $\text{gaps}_{\text{manual}}$  denotes the gaps in the manually clipped lyrics,  $\text{gaps}_{\text{automatic}}$  denotes the gaps in the automatically clipped lyrics and  $\text{len}$  is the length of the alignment.

### Test Data

The ground truth for our evaluation was created by crawling for ten different songs that were randomly chosen from our track database and extracting the lyrics manually from the downloaded pages. In total, 100 lyrics pages were analyzed<sup>16</sup>. Table 3.2 shows which songs were used.

Song Title	Artist
All I Do Is Think of You	Troop
Everything Will be Alright	the Killers
Futoreal	Iron Maiden
Good Morning Sunshine	Aqua
Izabella	Jimi Hendrix
My Fault	Eminem
Satisfaction	The Rolling Stones
Stay Gold	Stevie Wonder
This Will Be My Year	Semisonic
Violet	Hole

Table 3.2: Songs used for evaluation of lyrics clipping

<sup>15</sup>see Section 3.3.3 or Knees (2008, Section 5).

<sup>16</sup>This number was reached by chance. It is neither possible to use a fixed number of pages per song since the number of obtainable pages varies, nor did we stop as soon as 100 pages had been collected. However, we started out aiming to use approximately 100 lyrics for our evaluation.

Pages that did not contain lyrics were not used. Any text obviously not part of or relating to the song lyrics' content, such as the title, the artist, copyright disclaimers, watermarks and the like were not included. On the other hand, annotations<sup>17</sup> were included, even if they were found at the end of the lyrics.

### Results and Discussion

	precision	recall	processing time
mean	0.983	0.961	23.9 ms
standard deviation	0.031	0.123	19.5 ms

Table 3.3: Mean and standard deviation for precision, recall and processing time of Lyrics Clipping

Table 3.3 shows mean and standard deviation for precision, recall and processing time<sup>18</sup>. Precision and recall are plotted for every single test case in Figure 3.1 and boxplots<sup>19</sup> are displayed in Figure 3.2.

Low precision is caused by a text fragment that does not belong to the lyrics but is found in the same environment, such as the artist's name, the title of the song, copyright disclaimers and watermarks. The more of these artifacts are present, the lower the yielded precision value will be.

Low recall is mostly due to the fact that the lyrics section of the downloaded page contains html markup that cannot be removed by the proposed algorithm because it is code affecting page layout, not text style. In 8 out of a total of 11 cases where recall was  $< 1$ , the lyrics were deliberately interrupted by advertisements or watermarking code. Only two of the lyrics portals that we got results from present lyrics this way<sup>20</sup>. In the other three cases, considerable deviation from the html standard made recognition of the relevant part of the page impossible.

For 59% of the test sample, LC reaches a precision of 1. In only 2 of these cases, LC shows a recall smaller than 1. In contrast to that, from the 41% of all test cases for which LC has a precision smaller than 1, LC has non-perfect

<sup>17</sup>Such as *[Chorus]*, *~ad lib til fade~*, and the like.

<sup>18</sup>Measured on a 3GHz Intel Pentium4 processor with 1 gigabyte memory.

<sup>19</sup>We use the standard implementation of boxplots in R (R Development Core Team, 2006). A box is drawn around the region containing the upper and the lower quartile, whiskers extend to maximally 1.5 times the interquartile range in both directions, any data points beyond these limits are displayed as outliers.

<sup>20</sup><http://www.lyricsandsongs.com> and <http://www.lyricsdepot.com>, as of 2007/04/06

Figure 3.1: Plot of precision and recall of Lyrics Clipping for every single test case.

Figure 3.2: Boxplots showing the distribution of precision and recall values of Lyrics Clipping.

recall in 16 cases. This correlation can be explained by the algorithm being more susceptible to precision problems than to recall-related ones: Spurious words are included in the result rather easily; important words are only dropped if the Web page does not fit the patterns at all, and when this is the case, there is often non-relevant text mixed up with the actual lyrics in the clipping result.

Both measures reach quite high values considering that only simple pattern matching is performed for clipping the lyrics from the Web pages; on average, 98.3% of all extracted words actually belong to the lyrics and 96.1% of all words in the lyrics are found in the clipped version. From 88% of all pages, the complete lyrics were clipped successfully without missing a single word. In 59% of the clipped lyrics, not a single word was wrongly included.

### Conclusion

The experimental results show that Lyrics Clipping is useful for extracting lyrics from the Internet. Yet, as the quality of resulting lyrics is subject to variations, direct use of these results bears a certain risk. Several properties of LC suggest using it as a preprocessing step in MLA, though:

- As has been stated in the motivation for LC, due to LC's high precision, the smoothing step of MLA can be left out, which reduces the risk of removing relevant words within the lyrics.
- LC is fast, even for large pages, and its results are normally considerably smaller than the original Web pages. As MLA has more problematic space and time complexity, using the output of LC as input is a considerable improvement.
- The algorithm's performance with respect to recall is crucial if text mining is to be performed on the resulting lyrics as it is highly undesirable to lose potentially important words. The average recall of 96.1% may be alarming under this aspect, but it has to be noted that severe recall problems arise only with single lyrics portals. As MLA combines lyrics mined from different sources, we expect it to be capable of alleviating these shortcomings in most cases.

Due to time constraints we did not perform experiments comparing the performances of MLA with and without LC. We rather decided in favour of more thorough investigation of the text mining tasks. The latest developments have probably made it unnecessary to perform such an evaluation, as a combination of clipping and alignment approaches were proposed since we have developed our implementation (Korst and Geleijnse, 2006).

## Summary

This chapter gives an overview of the methods we apply in the implementation of our algorithm. We introduce some basics of text mining and semantic analysis as well as the clustering technique that we use for the automatic detection of topics. We explain how extraction of lyrics from the Internet works and we give an account of our effort to improve the described technique. At one point, we show multiple approaches for the solution of the same problem (see Section 3.2) in order to illustrate that the most effective application of complex algorithms, in this case NMF, is not always straightforward. Furthermore, we explain the foundations of semantic analysis (see Section 3.1.3), not so much because we applied these methods in our algorithm but because they conceptually underlie any topic detection method and therefore give the reader some insight into the kind of relationships between terms that are found by NMF.

# Chapter 4

## The Corpus

---

### Abstract

In this chapter, we describe the creation and important properties of our lyrics corpus. We describe the set of songs our corpus is based upon very briefly in Section 4.1, the process of gathering the lyrics in Section 4.2 and relevant quantifiable properties of our lyrics corpus in Section 4.3. The information we provide is helpful for understanding decisions we make in the process of parameter optimization covered in Chapter 7.

### 4.1 The Dataset

We work with a music archive that has been developed by the Research Studio Smart Agent Technologies<sup>1</sup> in cooperation with Verisign Austria<sup>2</sup>. This set comprises approximately 60.000 highly popular audio tracks<sup>3</sup> by some 6.000 artists.

### 4.2 Crawling

Through our cooperation with the Institute of Computational Perception of the Johannes Kepler Universität Linz<sup>4</sup>, we had access to the source code of MLA, including the crawler. This java based application was adapted for parallelized crawling of multiple lyrics, thus enhancing throughput. We used MLA with 0.25 for the threshold parameter  $t$ . The crawler was configured to download the first 20 search results and use the first 10 pages that are downloaded successfully.

The lyrics to the songs in our database were extracted from the Internet at a speed of 28 seconds per track on average; this worked for roughly two thirds of the songs. The reason for this rather low rate of success is that the method we employed is not suitable for some classes of tracks. Such classes

---

<sup>1</sup><http://sat.researchstudio.at/>

<sup>2</sup><http://www.verisign.at/>

<sup>3</sup>i.e., the most frequently accessed tracks in Verisign's Content Download Platform within a certain time span for a specific music portal.

<sup>4</sup><http://www.cp.jku.at/>

comprise of course the purely instrumental songs, but also works that bear a composite *artist* attribute, e.g., “*Mint Condition/feat. Charlie Wilson of the Gap Band*”, which are hard to find with Internet search engines without customized rules. Moreover, the lyrics to older or only locally known music are often not found by MLA.

After removal of duplicates, which are also present in our dataset, our corpus comprises 33863 lyrics.

## 4.3 Statistics

In this section, we give a summary of some properties of the lyrics corpus. These properties are referred to in later chapters as the basis of certain design decisions, hence at this point we do not draw conclusions nor do we propose hypotheses about the data.

### 4.3.1 Genres

The tracks are affiliated with one or more of 31 genres. The most important genres are “*Alternative*” (7880 songs), “*Pop*” (7753 songs), “*Hip-Hop*” (3527 songs), “*Rock*” (3309 songs) and “*Country*” (3164 songs). Figure 4.1 shows the frequency of all genres in the lyrics corpus. The frequency relates to the *main genre* of each track. It is evident that the distribution is highly uneven; the largest six genres account for more than 80% of the tracks. The genres are actually not a flat list but a taxonomy (e.g., there is “*Hip-Hop*” and the sub-genres “*West Coast*” and “*East Coast*”). For our purposes, however, genre is not of special importance, which is why this relationship between genres is not reflected in the figures.

### 4.3.2 Languages

The lyrics are in 15 different languages, though the vast majority is in English. We found about 300 Spanish songs, roughly 30 in Italian, French, Gaelic and Latin; all other languages are even less frequent.

### 4.3.3 Length

In our corpus, the average song’s lyrics are about 1370 characters long. The lengths vary between six and 28538; both extremes are due to errors of MLA. Figure 4.2 shows a histogram of the length of the lyrics. In addition to that, length is grouped by genre and contrasted with the “size” of each genre in

Figure 4.1: Distribution of genres in the corpus

Figure 4.2: Distribution of the document lengths in the corpus. The 224 Documents that are longer than 5000 characters are not shown.

Figure 4.3. Among the more important genres, “*Hip-Hop*” has the highest average length, followed by “*Dance/Electronic*” and “*Country*”.

#### 4.3.4 Confidence

The confidence value assigned by MLA is an indication for the quality of the extracted lyrics. The average confidence of our corpus is 0.936; Figure 4.4 shows a histogram of the confidence value. In Figure 4.5, confidence is plotted over length to give an impression of the relationship between the two values.

Figure 4.3: Average length of lyrics grouped by genre. The number of songs in each genre is shown in the left column.

Figure 4.4: Distribution of the confidence value in the corpus. The 340 Documents with a confidence equal to or below 0.6 are not shown.

Figure 4.5: Joint Distribution of confidence and length

# Chapter 5

## Automatic Topic Detection Algorithm

---

### Abstract

The structure of the algorithm at the very core of our work is accounted for in this chapter. Having explained the details of the applied methods earlier<sup>1</sup>, we are now in the position to give a rather concise view of our system. The account spans all steps necessary for processing a collection of lyrics and creating an information retrieval system that uses automatically detected topics. In a later chapter we shall dilate on the decisions regarding the various parameter values; here, however, we use the values we finally chose. We explicitly accept this temporal inconsistency in order to have the most important information available at one spot.

### 5.1 Introduction

The solution to any interesting problem in computer science is accomplished in a two-sided process.

One side deals with finding the necessary functions and the mode of their combination to reach the envisioned goal. When solving that side of the problem, the aspect of interest is *form*: the form of the input, how it is modified by the functions and the form of the desired output.

The other side of the process is *content*-oriented. On that side of the process, the goal is to find suitable parameters for the chosen functions in order to produce an output value that lies within a margin of error around the desired output. This is done by informed guessing, trial and error or by conducting experiments.

As long as the process continues, these sides can hardly be viewed separately because they interfere with each other constantly. Ex post, however, this separation is possible and greatly simplifies explaining the solution.

In this work, the side concerned with form and functions is treated in the remainder of this chapter. The process of finding suitable values for the function parameters are covered in Chapter 7 and Chapter 8.

---

<sup>1</sup>cp. Chapter 3.

## 5.2 Overview

We propose a system that allows for searching a collection of lyrics by selecting from a set of topics. In this chapter, we describe the formal details of the procedure that we propose for making a collection of text files browsable by topic. This process has been developed with a specific collection of lyrics and is therefore biased toward it. However, none of the system's elements are applicable solely to the lyrics domain. Although it may have to be changed in some ways, the program can be used to index and search any kind of text by topic.<sup>2</sup>

The main contribution of this work, automatic topic detection, is essentially a transformation of the text collection into a vector space model in which each document is represented by a vector of topic affiliation values. The whole system we propose consists of the following logical building blocks:

1. *Preprocessing*  
Reading text from text files and creating a *term-document matrix* (TDM).
2. *Automatic Topic Detection*
  - (a) *Term Selection*  
Shrinking the TDM by dropping terms and documents.
  - (b) *Term Weighting*  
Changing the values for the terms in the TDM.
  - (c) *Clustering*  
Dividing the TDM into a set of *topics*.
  - (d) *Labeling*  
Manually assigning labels to the topic clusters.
  - (e) *Calculating Topic Affiliation*  
Automatically assigning topics to documents.
3. *Query Definition and Retrieval*  
Manually defining a query vector and producing matching documents.

The rest of this chapter explains these elements of the algorithm in detail. For further discussion of the actual parameters the reader is referred to Chapter 7 and Chapter 8.

---

<sup>2</sup>This is why we use the rather general terminology of text mining. Throughout the chapter, the word *document* actually refers to a *lyrics document*.

## 5.3 Preprocessing

This stage starts the whole preparation procedure. Text files are read from the file system and represented as a *vector space model* (VSM), in other words, a TDM is created. Each text file is interpreted as a *document*, the text is tokenized, the resulting tokens form the set of *terms* that represents the document. Stemming can be used to sum up all different forms of a word and to reduce the number of different terms. We use the R package *tm* (R Development Core Team, 2006; Feinerer, 2007) for this step.

## 5.4 Automatic Topic Detection

### 5.4.1 Term Selection

Two approaches are used to cut down on the number of terms. First, stopword lists for a number of languages help to remove the most frequent words, which are not considered to be helpful for topic clustering. We used stopwords<sup>3</sup> for English, Spanish, French and German and a custom stopword list for lyrics<sup>4</sup>. Second, terms and documents are deleted if they do not meet conditions defined by upper and lower thresholds for the document frequency of a term ( $f_{\max}$ ,  $f_{\min}$ ) and by a minimal term count for documents ( $t_{\min}$ ). The pseudocode for this operation is shown in Algorithm 3.

<p><b>Data:</b> TDM, <math>f_{\max}</math>, <math>f_{\min}</math>, <math>t_{\min}</math>  <b>Result:</b> reduced TDM  Remove terms with document frequency <math>&gt; f_{\max}</math> ;  Remove documents with less than <math>t_{\min}</math> terms ;  <math>t_r \leftarrow 1</math> ;  <b>repeat</b>        Remove terms with document frequency <math>&lt; f_{\min}</math> ;        <math>t_r \leftarrow</math> number of terms removed in last step ;        Remove documents with less than <math>t_{\min}</math> terms ;  <b>until</b> <math>t_r = 0</math> ;</p>
---

**Algorithm 3:** Shrinking the TDM

The effect of shrinking the TDM is illustrated in Figure 5.1.

<sup>3</sup>These lists come with the R package *tm* and are part of the snowball stemmer (<http://snowball.tartarus.org/>).

<sup>4</sup>This list contains mainly exclamations like “*uuh*” and non-lyrics terms such as “*song-lyrics*” or “*inurl*”. The full list can be found in Appendix A.

Figure 5.1: Effects of term selection on the TDM. Both rows and columns may be dropped.

Figure 5.2: Conversion of the term weighting from *term frequency* to *binary*.

### 5.4.2 Term Weighting

The term *term weighting* refers to the values terms take in documents as a part of the TDM. In our process, the TDM is created using just the term frequencies, the occurrence count of each term in each document. In current information retrieval systems, a number of different weighting schemes are used depending on the problem at hand (Baeza-Yates and Ribeiro-Neto, 1999). We found the binary weighting to be the most useful at this point. As explained in Section 3.1.2, binary weighting assigns a value of 1 to a term if it occurs in a document and 0 otherwise. Figure 5.2 shows how the TDM is changed in the weighting step.

### 5.4.3 Clustering

The TDM is clustered using NMF (Xu et al., 2003). This technique is described in detail in Chapter 3. For our current interests, suffice it to explain that using this method, the TDM is approximated by the matrix product of two matrices of appropriate dimensionality. NMF is parametrized most

Figure 5.3: Non-negative factorization of the TDM. The TDM is approximated by the product  $WH$ .

prominently by the number of clusters that it shall produce,  $k$ . More formally, let  $T$  be the TDM,  $W$  and  $H$  the factor matrices, and

$$T_{\text{documents} \times \text{terms}} = W_{\text{documents} \times k} H_{k \times \text{terms}} \quad (5.1)$$

The parameter  $k$  is the *inner dimension* of the factorization, i.e., the number of dimensions that both factor matrices share. For our corpus of lyrics, best results were achieved using  $k = 60$ . The approximation of the TDM by the NMF is depicted in Figure 5.3. The more important of the two factor matrices for our purposes is  $H$ , which contains the weights of each term in each cluster.

#### 5.4.4 Labeling

In order to make the clustering useful to an end user, the clusters need some kind of identifiers associated with them which hint at the nature of each cluster's content. Labeling a cluster was done by reading its most important terms and assigning one or more words (tags) that best summarized those terms. This stage is illustrated in Figure 5.4. The exact procedure followed to label the clusters and its results are described in Chapter 8.

Figure 5.4: Manual labeling of the NMF clusters in the factor matrix  $H$ . The manually assigned labels in this example are 'love', 'party' and 'loss'.

Figure 5.5: Computation of the documents' affiliation strength to the clusters

### 5.4.5 Calculating Topic Affiliation

The degree of membership of a document in each cluster is computed using the original TDM weighted with term frequencies. This step is shown in Figure 5.5. First, only the columns (i.e., terms) that were used in the NMF are selected ( a ). The resulting matrix is multiplied by the transposed factor matrix  $H$  from the NMF that contains the term weights per cluster ( b ). Hence, for each cluster and each document, each term belonging to the document is multiplied by the weight of the term in the cluster and the sum over these products is regarded as the weight of the cluster for the document. After the calculation of this document affiliation matrix, its rows are normalized to a length of 1 in the euclidean norm.

This manipulation concludes the transformation process, resulting in an information retrieval system in which the documents (lyrics) are represented by cluster affiliations in a vector space model.

Figure 5.6: Query definition and document retrieval

## 5.5 Query Formulation and Document Retrieval

Retrieval of documents requires defining a query specifying which clusters the documents should belong to. Resulting documents are ranked with respect to that query. Figure 5.6 illustrates this process. A query vector is defined, assigning a weight to each cluster ( a ). The cosine similarity values between the rows of the document affiliation matrix and the normalized query vector define the rank of all documents in the query result ( b ). The result is sorted in descending order; the first results are the most relevant ones ( c ).<sup>5</sup>

### Summary

We present the algorithm for automatic topic detection and its application for creating an information retrieval system. The focus lies on explicitness and legibility of the explanations, which is why we illustrate each step instead of giving formal definitions. Our algorithm consists of preprocessing, term selection and weighting, automatic clustering, manual labeling, calculation of topic affiliation for each document, and of a method for defining queries for documents.

---

<sup>5</sup>Note that the computation is shown as a matrix multiplication. This is actually correct because the rows of the matrix and the query vector are normalized to have a length of 1 in the euclidean vector norm – in which case the dot product yields the cosine.

# Chapter 6

## Assessing the Quality of Identified Topics

---

### Abstract

In this section, we briefly explain which methods we used for assessing the quality of clustering results. In Section 6.1, a tool displaying information about single clusters is described. It is meant to give an impression of the topic that the cluster describes, and in particular, how recognizable that topic is. Section 6.2 deals with a method that allows for viewing the similarity relations between clusters, giving an overview of the whole cluster system. The pragmatic value of a whole clustering result, i.e., how useful it is for searching songs, is assessed by methods described in Section 6.3.

### 6.1 Viewing Term Weights

In the clustering algorithm that we used, NMF, the weights of the terms are contained in the factor matrix  $H$ . In order to find out about the topics represented by a cluster, we look at the cluster's most important terms. For this purpose, we wrote evaluations that were capable of producing these term lists along with a graph showing the weights of the terms. Example 6.1.1 and Example 6.1.2 give an example of a cluster inspection. The distribution of weights in the example is actually quite representative: In clusters, the term weight decreases exponentially, differences only concern the rate of decrease and the absolute maximum.

```
Cluster # 30
1  hold hand    tight  arms  close  touch
7  kiss strong  forever break  till   breath
13 help hands   hope   told  feeling feels
19 wait near    onto   fear  soon   understand
25 gon  holding watch  stand lips  faith
```

Example 6.1.1: Most important terms of a cluster

Example 6.1.2: Plot of term weights. The dotted section of the plot has not been chosen to represent the cluster.

Taking into account what has been said about the typical weight distribution, it should be clear that it is not trivial to decide how many terms should be used when a cluster's most important terms are to be displayed.

If a fixed number of terms is used, i.e., the top  $N$  terms, this can lead to quite different qualities in the result. A cluster that has an extremely narrow term weight distribution is then represented by terms which are relatively irrelevant. Symmetrically, in clusters with a broad term weight distribution by far too few of the really important terms are shown.

Any other method for selecting the number of terms to show must calculate this number separately for each cluster. In the cases just discussed, this has the tendency to lead to some clusters being described by just a few words whereas for others, more than one hundred are displayed.

We eventually decided to apply a combination of the two ideas: At most 30 tags are shown per cluster (i.e.,  $N = 30$  was used), but all of these must have weights larger than the mean of all weights plus the standard deviation of all weights, or more formally, let  $w_{c,t} \in W_c$  be the weights of the terms  $t \in T$  with respect to a cluster  $c$ , then the set  $T_c$  of terms to display for the cluster is defined as follows.

$$top_N(X) = \begin{cases} \text{highest } N \text{ values} & \text{if } |X| > N \\ X & \text{else} \end{cases} \quad (6.1)$$

$$T_c = top_N(\{t \in T | w_{c,t} \geq \text{mean}(W_c) + \text{stddev}(W_c)\}) \quad (6.2)$$

While this method for determining the term count proved quite useful, it turned out later that it would have been even better to use a lower bound for the term count as well.<sup>1</sup>

## 6.2 Clustering Clusters

Viewing term weights in clusters, as described above, gives a good impression of a single cluster. However, it is also of great interest to see the overall similarity structure of all clusters. This functionality is realized in an additional evaluation tool, which calculates similarities between all clusters and depicts them in a dendrogram.

The inter-cluster similarity is calculated by first computing the affiliation value for each document in the TDM and each cluster as explained in Section 5.4.5. The resulting document-cluster affiliations are the rows in the resulting matrix, the columns of which are interpreted as cluster representations, and their mutual cosine value is calculated.

---

<sup>1</sup>One cluster got overly bad ratings in the experiments because it was described only by the five terms “*re*”, “*coming*”, “*feeling*”, “*looking*”, “*else*”, none of which were linkable to any topic. Consult the experimental results for Cluster #52 in Appendix B for details.

### Example 6.2.1: Dendrogram of clusters

Example 6.2.1 shows such a plot. The height at which nodes are connected represents their similarity; the further to the right this combination is shown, the more similar they are. Line thickness and shading indicate the number of items (documents) in the leaf (cluster) or branch (combination of clusters). In the example, the leaf labels are composed of the cluster index in square brackets, one or more tags and a number indicating perceived quality in brackets<sup>2</sup> and the strongest three terms. The numbers in square brackets are the clusters' ids.

---

<sup>2</sup>Tags and quality were attributed during experiments, see Section 7.2 for more information.

### 6.3 Searching Songs

The ultimate purpose of our system is indexing and retrieval of songs. The evaluation methods we have covered so far, however, are concerned with the assessment of the quality of some of its parts. In this section, we describe an evaluation tool that is used to analyze the usefulness of our whole system with respect to its purpose: searching songs.

Retrieval of song lyrics is done essentially by specifying a query vector<sup>3</sup>, assigning a value  $w \in [-1, 1]$  to each cluster of interest (all other clusters get weight 0). With this parameter, the search algorithm described in Section 5.5 on page 47 is executed, producing a list of songs sorted by query match score. This list can be viewed by the user. For each result, a bar plot is displayed, showing the cluster affiliation values of the current song.

For example, the query vector ('love forever' = 1, 'loss past' = 1) may yield the lyrics shown in Example 6.3.1, with the cluster affiliation values given in Figure 6.1.

---

<sup>3</sup>As described in Section 5.5 on page 47.

<p>i should have known you'd be this  way  serves me right, to fall in love  you brought me love then you took  it back  we had it all and that's a fact  love is a game that we all play  there's one thing i've got to say  i gave you my heart and i gave you  my world  i spent those lonley nights right here  in the cold  everytime you touch me my body  starts to quiver  all i want to do is love you all night  long  chorus:  can't let you go my love  it's you that i need  you keep me wanting you  you push my love aside  repeat  i can't help but think of how you  made me feel  everytime i think of the two of us  can't realize  can't realize  can't seem to buy it  now that you're gone  now that you're gone  chorus: can't let you go my love  it's you that i need  you keep me wanting you  you push my love aside  repeat (1x)  chorus: can't let you go my love  it's you that i need  you keep me wanting you  you push my love aside  repeat (1x)  can't let you go my love</p>	<p>you fit me like a glove  'cause when you hold me tight  the feelings oh so right  just can't foget those times  when you were mine, all mine  this love i have for you is true  so come back to the one who loves  you  can't let you go'  can't let you go'  everytime you me my body starts  quiver  all i want do is love you all night  long'  i should have known that you'd be  this way  serves me right to fall in love  i can't help but think how you left  me feeling  everytime i think about the two of  us  can't realize  can't seem to buy it  now that you're gone  now that you're gone  chorus: can't let you go my love  it's you that i need  you keep me wanting you  you push my love aside  repeat (1x)  chorus: can't let you go my love  it's you that i need  you keep me wanting you  you push my love aside  repeat (1x)  chorus: can't let you go my love  it's you that i need  you keep me wanting you  you push my love aside  repeat (1x)  i can't let you go</p>
--	--

“Can't Let You Go” by *Coro*

Example 6.3.1: Result of query for a song from the clusters tagged “*love forever*” and “*loss past*”. The cluster affiliation values are shown in Figure 6.1 on page 54

Figure 6.1: Cluster affiliation of the song from Example 6.3.1

# Chapter 7

## Parameter Optimization

---

### Abstract

In this chapter, we give an account of the efforts made for optimizing the result of the proposed algorithm. Depending on the nature of the specific problems that arose, we could either carry out this optimization by trial and error (Section 7.1), or we were obliged to design experiments in order to find the best values. These experiments are described in the Section 7.2.

## 7.1 Informal Parameter Optimization

### 7.1.1 Preprocessing

#### Improving the Quality of the Corpus

From the beginning, our research was aimed at being applicable to a song set of moderate but realistic size. As explained in Chapter 4, the lyrics to 33863 songs had been downloaded from the Internet, which we considered by far enough. It turned out, however, that we needed such numbers because a number of those documents had to be deleted due to flaws of our lyrics harvesting program.

A closer look into our corpus reveals a correlation between the length of the text and its quality: the shorter it is, the more unlikely it is to contain the complete lyrics, but rather only a small part of it or, in some cases, only snippets of html code or text on a Web page (see examples in Example 7.1.1). A viable solution for improving the quality of the corpus is therefore to filter short lyrics. However, some songs actually do have very short lyrics (see Example 7.1.2); moreover, this is more common in some genres than in others<sup>1</sup>, so filtering introduces a genre bias into the corpus. We decided to accept this trade-off both because our research is not focused on genres and because lyrics that short do not convey much semantic content to analyze, at least not in textual form.<sup>2</sup> Our corpus was therefore created using a lower

---

<sup>1</sup>cp. the genre-specific lengths displayed in Figure 4.3 on page 39

<sup>2</sup>Note that brevity of lyrics may, however, be a valuable piece of information in a general music recommendation scenario.

threshold of 200 characters for the length of lyrics. The number of songs in the collection was thus reduced by approximately 3% to 32831.<sup>3</sup>

There are 1 Runnin' the Game lyrics found.  
Showing Runnin' the Game song lyrics page 1 of 1

---

Lyrics temporarily unavailable.  
Please check back later

---

(Software Version 4. 6 11. 02. 2007)  
Datenbank Hosting by Powerplant New Media

Example 7.1.1: Three samples of text erroneously identified as lyrics.

Oh oh oh whoah  
Oh oh oh whoah  
Oh oh oh whoah

“Switch 625” by *Def Leppard*

---

Se a cabo  
Se a cabo  
Se a cabo  
Se a cabo

“Se a cabo” by *Santana*

---

That's right  
Have more rhythm  
Woooo!  
More rhythm

“Cherry Twist” by *The Chrystal Method*

Example 7.1.2: Three songs with very short lyrics.

---

<sup>3</sup>cp. the distribution of the lyrics' lengths in Figure 4.2 on page 38.

Our lyrics harvesting system assigns a confidence value to each extracted text (see (Knees et al., 2005)), which is used to identify lyrics to be deleted from the corpus. The confidence, a real value  $\in [0, 1]$ , is an estimate of the quality of the text, i.e., how close it is to the real lyrics of the song in question. Informal analysis showed that this value indicates poor quality quite reliably below 0.7. Higher confidence values did not seem as strongly correlated with the observed quality of the lyrics. These findings led us to set the lower threshold to 0.7, which further reduced the number of songs by 1.5% to 32323.<sup>4</sup>

### Stemming

We considered the use of stemming<sup>5</sup> at this point of the process but eventually decided against it. The strongest arguments for the use of stemming are a) the document description becomes less noisy and b) the term-document matrix becomes smaller. The downside of this method is that stemming algorithms are not perfect<sup>6</sup> and thus introduce a new kind of noise. Moreover, we believed stemming to be less reliable for text collections containing, as ours, different dialects or sociolects in quite peculiar non-standardized styles of transcription.<sup>7</sup> The stemmed terms are also less readable than the original terms, making subsequent research tedious unless each stem was projected back to the most frequent original term, which in turn entails additional programming effort. The most important argument against the use of stemming, however, was that we did not observe a strong effect of stemming during informal research.

### 7.1.2 Term Selection

Reducing the number of terms is a powerful measure to influence the outcome of the subsequent clustering stage. On the one hand, this decreases the size of the TDM, making any subsequent computation faster and thus allowing more thorough experimentation. On the other hand, experiments showed that term selection can have tremendous influence on the kind of clusters that are produced.

Deletion of stopwords mainly reduces noise, thus improving clustering

---

<sup>4</sup>cp. the distribution of the confidence values in the corpus in Figure 4.4 on page 40.

<sup>5</sup>Different forms of a word are projected to one form, the stem; e.g., “*running*”, “*runs*” → “*run*”.

<sup>6</sup>Forms of different words are projected to the same stem and forms of the same word are projected to different stems.

<sup>7</sup>e.g., “*feelin*”, “*gettin*”, “*boyz*”, ...

performance. We did not evaluate the effect of stopword removal as it seemed clear from skimming through these lists that these words do not convey any information relevant for topic clustering tasks. As a number of different languages are present in the corpus<sup>8</sup>, most prominently English, Spanish, French and German, we applied stopword removal<sup>9</sup> for these. In addition to these lists, we created a specific stopword list for lyrics<sup>10</sup>.

Terms with especially high or low document frequency<sup>11</sup> are natural candidates for removal from the TDM. This is due to reasons derived from theoretical considerations concerning clustering, which is a technique used to find groups in document sets; documents belonging to such a group are more similar to each other than to members of other groups. Terms occurring in only one document can hardly be useful for clustering, as they only add dimensions to the document representation in which they are equally dissimilar from all other items. We therefore chose to eliminate all terms with a document frequency of 1, which in our case reduced the memory consumed by the TDM by about 3%.

Terms with a very high document frequency account for dimensions of the document representation in which a great amount of documents are similar. With respect to clustering, these dimensions may be regarded as noise that makes the real clusters more difficult to discern. This rationale suggests that the clustering result improves as frequent terms are removed. Moreover, removing the frequent terms greatly reduces the space consumed by the TDM: in our case, deleting terms with a document frequency  $> 500$  reduced the memory consumption by 42%. However, the downside of this strategy may be that clusters that are mainly defined by frequent terms are lost. Informal experiments indicated that the use of an upper limit for the document frequency makes the resulting clusters more diverse, but it was impossible to decide for one of the limits (including no limit) based on these observations, so this parameter was chosen for systematic evaluation (see Section 7.2).

### 7.1.3 Term Weighting

In general text retrieval applications, it is important to apply the right weighting to terms representing documents. The term weighting function

---

<sup>8</sup>cp. Section 4.3.2 on page 36

<sup>9</sup>These lists come with the R package *tm* and are part of the snowball stemmer (<http://snowball.tartarus.org/>)

<sup>10</sup>This list contains mainly exclamations like “*uuh*” and non-lyrics terms such as “*song-lyrics*” or “*inurl*”. See Appendix A for the whole list.

<sup>11</sup>As explained in Chapter 5, a term’s document frequency is defined as the number of documents it occurs in.

is a transformation of the TDM that produces a matrix of the same dimension with (not necessarily) different values. In text retrieval applications, the purpose of the weighting function is to amplify the weights of the terms that are most typical for a document and to lower the weights of the other terms. This is because the main use case of the system is *retrieval*, i.e., finding documents that match a user-defined query. In this use case, it is desirable to have those documents match for which the terms in the query are most important. In our algorithm, the use case to optimize for is not retrieval but *clustering*. During our experiments, we learned that this requires different properties of the weighting function. We evaluated the usefulness of three different weighting schemes in our context. In the following, we describe the consequences of using different weighting functions before clustering. The weighting functions examined here are described in more detail in Section 3.1.2 on page 16.

- *Term Frequency*. This is the most simple weighting function possible – no change to the TDM at all. The weight of a term is its frequency in the document. This function is biased with respect to document length: Longer documents have a higher sum of term weights and are therefore more important in the TDM than shorter documents. Clustering using weighting produced term clusters in which the frequent terms were clearly overrepresented.
- $TF \times IDF$ . This weighting scheme, possibly the most frequently used one in text retrieval, attributes strong weight to the typical terms and low weight to the terms that are present in many other documents. Moreover, a normalization with respect to document length (term count) is applied. When used before clustering the TDM, the terms with low document frequency are too important in the resulting clusters.
- *Binary*. This method is defined as replacing all nonzero entries in the TDM by 1. Any term frequency information is removed from the document representation, which is a two edged property: on the one hand, frequent and infrequent terms in a document become equally important in its description. On the other hand, this makes the document representation more robust against certain problems of the lyrics harvesting process, which result in overly frequent terms<sup>12</sup>. It may also be interesting to note that binary weighting incurs the same bias with

---

<sup>12</sup>cp. Section 3.3.1 on page 24

respect to document length as the term frequency weighting. Notwithstanding this shortcoming, we found that binary weighting led to the best clustering results of the three approaches we tried: the problems brought about by the other two were not observed here and as a second positive effect the distribution of songs into clusters turned out to be much less skewed with binary weighting.

### 7.1.4 Clustering

Like most clustering algorithms, NMF is parametrized with the number of clusters to produce,  $k$ . As it is an iterative algorithm, its behaviour can be further influenced by the number of iterations.

#### Number of Iterations

We implemented NMF such that the approximation error<sup>13</sup> was recorded in each iteration. This allowed us to find the minimal number of iterations necessary for producing stable results, which was 30. This number has to be taken with a grain of salt, though, because we could not verify it for all numbers of clusters that we tried in our experiments. This is due to the fact that the computation of the approximation error necessitates the explicit computation of the product  $WH$ , which is a non-sparse matrix of the size  $terms \times documents$  and obviously needs a considerable amount of memory. The factor matrices  $W_{documents \times k}$  and  $H_{k \times terms}$  themselves are also non-sparse, and the amount that they occupy in main memory depends linearly on the number of clusters,  $k$ . When we experimented with values of  $k = 60$  or  $k = 100$ , the computation of the approximation error was no longer possible due to memory shortage. As no systematic experiments were conducted to study the influence of the number of clusters on the approximation error, we had to rely on our feeling that 50 iterations be enough in these cases.

#### Number of Clusters

The number  $k$  of clusters to use in NMF is one of the crucial parameters of the whole system. Ideally, this number is equal to the number of discernable topics in the documents without overfitting. However, no two humans would

---

<sup>13</sup>i.e., the summed squared difference between the product of the factor matrices and the TDM.

agree on the same number when asked to do so<sup>14</sup>, and so we can safely maintain that there is no absolutely correct value for  $k$ .

It is interesting to note that, unlike one may intuitively assume, low  $k$  does not naturally lead to more general and high  $k$  to more specific topic clusters, and that the former conjecture is much less true than the latter.

A low value for  $k$  causes NMF to produce clusters that are actually mixtures of multiple topics, which may be related hierarchically, but this is not necessarily the case. For instance, one cluster may describe the topic “*love*”, and on a closer look, the sub-topics “*loss*”, “*happy*” and “*family*” are recognizable, while another cluster could at first glance contain only the “*gangsta*” topic but at the same time be the strongest cluster for all spanish songs. In the first case, the clustering result is acceptable – none of the songs that fall into the cluster would really be misclassified; the only valid criticism is the lack of exactness. In the latter case, a portion of the lyrics that fall into that cluster are clearly misclassified because the cluster combines multiple different “real” topics.

When using high values for  $k$ , NMF tends to produce more specific clusters, most of which are quite interesting and useful. One of the disadvantages of this setting, however, is a tendency of NMF to find the same topics multiple times. Another noticeable side effect of high  $k$  values is that the important terms tend to be co-occurents<sup>15</sup> of the first term in the cluster. In Example 7.1.3, the top 30 terms of such a cluster (from a clustering with the parameters:  $f_{min} = 2, f_{max} = 500, t_{min} = 1, k = 60$ ) are presented. The terms are sorted by weight, highest first, from left to right and top to bottom. The strongest term, “*lie*”, is sided by many strong co-occurents, which can be grouped by the meaning of “*lie*” they refer to: “*awake*”, “*bed*”, “*tired*” and more terms indicate the meaning “to stay or rest in a horizontal position”. The terms “*promise*”, “*lied*” and “*lying*”, suggest the meaning “to make an untrue statement with intent to deceive”. The terms “*goodbye*”, “*eye*”, “*guy*” and “*bye*” are rhyme words of “*lie*” and that is probably the reason for their being so highly weighted in the cluster.

---

<sup>14</sup>Assuming that we could, in principle, find two people who were willing to perform this task on some 30.000 lyrics.

<sup>15</sup>see Section 3.1.3 for the definition of significant co-occurrences.

lie	goodbye	thinking	lies	eye	awake
doesn	promise	bed	tired	loving	walked
giving	wouldn	leaving	bye	perfect	guy
lay	sad	lover	lied	lying	tear
learned	skin	tears	magic	hoping	oooh

Example 7.1.3: Top 30 terms of a cluster which contains mostly strong co-occurents of the strongest term, “*lie*”

We did not manage to clarify how useful or detrimental the effects of a large number of clusters is to the overall purpose of our system, finding topics in lyrics. In many cases, the strongest term in a cluster is ambiguous, and the co-occurring terms reveal this fact, so the cluster itself tends to be more ambiguous than one that is less dependent on the strongest term. While this may seem like a drawback, it may as well be very useful if the first term or terms were used as the cluster label. In this case, disambiguation is left to the end-user, who would naturally understand the label it in the presence of other labels. For example, consider two songs strongly affiliated with the cluster from Example 7.1.3 (called “*lie*”), and one of them is also in a cluster “*trust*”, whereas the other is a cluster labelled “*night*”. Although the cluster name “*lie*” is ambiguous in this example, it is not a problem in such an information retrieval scenario at all.

For rhyme words, though, the question of utility is quite easily answered: they do not help to define a topic cluster at all.

The above observations strongly motivate, in our view, the use of a high value for the number of clusters in NMF. Notwithstanding this conviction, we decided to conduct experiments in order to determine the best value for  $k$ . These experiments are described in detail in Section 7.2 on page 64 et seqq.

### 7.1.5 Document-Cluster Affiliation

Once the cluster representations have been computed by NMF, the documents’ affiliation to each of the clusters must be determined. The result of this step is a matrix  $A_{\text{documents} \times k}$ , assigning an affiliation value to each document-cluster combination. Thus, the documents in the corpus are represented in a modified vector space model, represented by clusters instead of terms.

Non-negative matrix factorization clusters not only the terms, but also the documents. This clustering is defined by the factor matrix  $W_{\text{documents} \times k}$  and it is a viable way to use  $W$  directly as the document-to-cluster affiliation matrix. The advantage of this approach is that  $W$  is a part of the clustering result itself and can therefore be expected to be more accurate than any

affiliation calculated based on  $W$ . In contrast to this, a major disadvantage is that no new, unseen documents can be integrated sensibly into such a system. In a realistic information retrieval scenario this is not an option, so we decided against using  $W$  as the affiliation matrix.

For the calculation of cluster affiliation values for all documents in the corpus, all that is needed is a TDM and a term-cluster matrix ( $H$ ). In Section 7.1.3 on page 58, we explained that different weighting schemes can be applied to a TDM, and that the best choice for the weighting depends on the envisioned use. Our current problem is different from the classic information retrieval problem and from clustering, so the best weighting has again to be determined experimentally. We made some effort to do so by searching songs<sup>16</sup>, using different weightings before computing the document-cluster affiliation. As there were no reliably discernable differences in the results, we accepted to use the initial TDM, i.e., the term frequency weighting, for these purposes. We readily admit, however, that this decision is not very well-founded and that systematic and formal research may well lead to different results. The main difficulty arising when comparing the different approaches here is that there is no way of evaluating their effects apart from using the whole system for retrieving songs. Without a decent user interface, such an evaluation is quite work-intensive, so in this case, we decided against a more formal approach.

After the calculation of the matrix  $A$ , we add a normalization of its rows to a length of 1 in the euclidean norm.

---

<sup>16</sup>See Section 6.3 on page 52

## 7.2 Parameter Optimization by Experiment

After performing the optimization steps explained in the previous section, some parameters were still undecided for. In the following, we give an account of the formal evaluation of different values for these parameters.

The parameters we still had to decide about were the number of clusters  $k$  to use for NMF and the upper limit for the document frequency of terms. The candidate values for  $k$  were 5,10,30 and 60. The effect of using an upper limit for the document frequency was tested with a limit of 500 and without any limit. All in all, we computed NMF clustering results for 8 combinations of parameter values. Those NMF results were combined in what we termed the *evaluation set*,  $E$ . The clustering results are in the following referred to as  $R_{k,f_{max}}$  as shown in Table 7.1.

		$k$			
		5	10	30	60
$f_{max}$	500	$R_{5,500}$	$R_{10,500}$	$R_{30,500}$	$R_{60,500}$
	$\infty$	$R_{5,\infty}$	$R_{10,\infty}$	$R_{30,\infty}$	$R_{60,\infty}$

Table 7.1: The evaluation set used in systematic optimization

The assumption governing the choice of these values – formed during the process of “non-systematic” optimization – was that the best clusters were produced with a high number for  $k$  and a document frequency limit of 500. The subsequent evaluation was designed to decide whether this assumption would hold.

There are two main aspects of quality that are observable in our system. First, the quality of an NMF cluster can be assessed by looking at the important terms in that cluster and deciding whether they go well with each other and define a discernable topic. This aspect is assessed for the said parameter combinations in Section 7.2.1. Second, the whole system can be evaluated with respect to its usefulness to the overall purpose of the system by using it to search songs for different topics. A comparison of all  $R_{k,f_{max}}$  with respect to this second quality measure is given in Section 7.2.2.

### 7.2.1 Assessment of Individual Cluster Quality by Labeling

#### Setup

The assessment of the NMF clusters’ quality was done using the inspection tool described in Section 6.1 on page 48 that shows the most important terms in a cluster. A primitive user interface was built that first displays a cluster’s

terms and then prompts the user for some tags and a numeric quality value that indicates how well these tags describe the cluster. An example of such an interaction is shown in Example 7.2.1.

```

1 cry      tears      pain      makes    goodbye  rain    lie
8 laugh    sometimes remember hurt     sad      bad     bye
15 wonder  sky         smile     friend   crying   cried   hope
22 child   sleep      lonely    dont     river    kiss    eye
29 tear    mother

```

Please take some time to read the terms. Continue with [ENTER]

Please enter tags for this cluster:

How well do these tags describe the cluster?

(1=well, 2=not well, not badly,3=badly):

Example 7.2.1: Tagging and quality assessment of an NMF cluster. User input is boxed.

The whole evaluation set  $E$ , consisting of eight clustering results, was split in two subsets,  $E_1$  with  $f_{max} = 500$  and  $E_2$  with  $f_{max} = \infty$ .<sup>17</sup> Both subsets thus contained 4 NMF clustering results.

The tagging and quality assessment procedure was conducted separately on  $E_1$  and  $E_2$ . During the assessment, no information was available as to the nmf result that the clusters belonged to. This was important so as not to be biased toward any value of  $k$ . Table 7.2 shows a summary of the evaluation. For each clustering result, mean and standard deviation of the quality measure are displayed (integer  $\in \{1, 2, 3\}$ ; the lower, the better). These values were weighted by the number of songs in each cluster such that large clusters, containing many songs, had a higher impact on the result than small ones.

The statistics in Table 7.2, separated into  $E_1$ (left) and  $E_2$ (right), show the rank of each clustering result within its evaluation subset. The rightmost column, *avg. rank* contains the average rank for specific  $k$  over the different settings of  $f_{max}$ .

<sup>17</sup>This fact is hard to explain logically as the reasons for it are rather historical. Our initial intention was to decide on a value for  $k$  using the method explained above. Actually seeing (unexpected) quantified results raised the question how they depend on  $f_{max}$ , the maximum document frequency, which in the first round of evaluation was  $f_{max} = 500$ . Consequently, a second experiment was set up with  $f_{max} = \infty$ .

## Results and Discussion

As a precondition to a scientifically correct interpretation of the results, we have to state that strictly speaking results in  $E_1$  and  $E_2$  are not directly comparable. This is due to the fact that they were evaluated by the same person in different sessions, which can introduce various kinds of bias. In the following, we assume that these effects are limited to a uniform, general influence on the quality values. For example, while assessing the quality of  $E_1$  the test person may have been in a better mood and rated all clusters quite positively, whereas he may have been in a bad mood while evaluating  $E_2$  and gave rather negative feedback in general. This assumption doesn't allow us to draw conclusions concerning the choice of the  $f_{\max}$  value isolated from  $k$ . What it does permit, however, is studying the effect of constant values of  $k$  and different  $f_{\max}$  on the clustering result.

The results show that a very low  $k$  leads to clusters of low quality. In both  $E_1$  and  $E_2$ , the NMF result obtained with  $k = 5$  was ranked third out of four, so this property can be said to be independent of  $f_{\max}$ . The NMF clusterings with  $k = 10$  were ranked first in  $E_1$  and last in  $E_2$  – in contrast to the above case, there is an obvious inverse correlation between cluster quality and  $f_{\max}$ . In the case of large values of  $k$ , a general connection between  $f_{\max}$  and the quality is arguable: the higher  $f_{\max}$ , the better the quality of individual clusters.

Comparing the mean qualities of clusterings in  $E_1$  and  $E_2$  relative to their respective average<sup>18</sup> actually confirms the observations stated above. Column “rel. mean diff.” in Table 7.3 shows that the clusterings with low  $k$  stagnate or deteriorate in terms of quality with higher  $f_{\max}$  whereas those with high  $k$  improve.

As to the “winner” of this evaluation, there is to say that average rank as well as mean and standard deviation of qualities strongly indicate that best results are reached with  $k = 30$ . Moreover, under the assumptions made above, we can draw the conclusion that  $f_{\max} = \infty$  is the better choice for  $f_{\max}$  in this case.

---

<sup>18</sup>i.e., the average quality in all of  $E_1$  and  $E_2$ , respectively

clustering	$E_1$			clustering	$E_2$			avg rank <sup>a</sup>
	mean	stddev	rank		mean	stddev	rank	
$R_{5,500}$	1.51	0.55	3	$R_{5,\infty}$	1.72	0.68	3	3.0
$R_{10,500}$	1.10	0.38	1	$R_{10,\infty}$	1.74	0.52	4	2.5
$R_{30,500}$	1.45	0.68	2	$R_{30,\infty}$	1.26	0.44	1	1.5
$R_{60,500}$	1.61	0.76	4	$R_{60,\infty}$	1.52	0.59	2	3.0
mean	1.42				1.56			

Table 7.2: Results of quality evaluation

<sup>a</sup>This column shows the average rank for each value of  $k$ .

clustering	$E_1$		clustering	$E_2$		rel. mean diff. <sup>b</sup>
	mean	rel.mean <sup>a</sup>		mean	rel. mean <sup>a</sup>	
$R_{5,500}$	1.51	1.07	$R_{5,\infty}$	1.72	1.10	-0.04
$R_{10,500}$	1.10	0.78	$R_{10,\infty}$	1.74	1.12	-0.34
$R_{30,500}$	1.45	1.02	$R_{30,\infty}$	1.26	0.81	0.22
$R_{60,500}$	1.61	1.14	$R_{60,\infty}$	1.52	0.97	0.16

Table 7.3: Results of quality evaluation: comparison of relative means, illustrating the effect of  $f_{\max}$ 

<sup>a</sup>This column shows the mean quality relative to the overall mean quality in the evaluation subset

<sup>b</sup>This column shows the differences of the “relative means”

## 7.2.2 Assessment of the Whole System's Quality by Retrieval

In this section we describe our efforts to evaluate how useful the different parameter settings (i.e., clustering results)  $R_{k,f_{\max}}$  are to a prospective end user. This can only be evaluated by simulating as closely as possible an IR application that uses our method for music browsing. Our simulation consists in the manual definition of a query vector, using these in the querying procedure and displaying the lyrics to the resulting songs and their topic affiliation values, as described in Section 6.3 on page 52.

### Setup

For our evaluation we defined IR tasks that had to be performed with each setting  $R_{k,f_{\max}}$ . These tasks consisted in finding songs for a list of topics, which was “romantic”, “party”, “life”, “religious”, “hiphop&crime” and “loss”. If the results for the first three tasks had been unsatisfactory, the evaluation of a specific setting was aborted. The usefulness of the system for performing a task was graded on a 1-5 scale (1 being the best grade).

### Results and Discussion

The results of this evaluation are presented in Table 7.4. It is evident that five and ten clusters are too few to get satisfactory results, so they do not receive an overall grade. The other clusterings can be ranked by their grades, the best one is  $R_{60,\infty}$ , mainly because none of the tasks yields a particularly bad result.

## 7.2.3 Consequences

We decided to decide on the values for  $f_{\max}$  and  $k$  on the basis of the experiment described in Section 7.2.2 because the aspect of quality covered is the more important one with respect to the envisioned application scenario. Hence, we use the parameter values  $f_{\max} = 500$  and  $k = 60$  for all subsequent work.

clustering	“romantic”	“party”	“life”	“religious”	“love”	“anime”	
$R_{5,\infty}$	–	–	–	–	–	–	–
$R_{10,\infty}$	–	–	–	–	–	–	–
$R_{30,\infty}$	1	2	2	5	3	2	2.5
$R_{60,\infty}$	2	3	3	1	2	2	2.16
$R_{5,500}$	–	–	–	–	–	–	–
$R_{10,500}$	4	2	5	–	–	–	–
$R_{30,500}$	3	1	4	4	2	2	2.66
$R_{60,500}$	1	3	5	5	2	1	2.84

Table 7.4: Assessment of the IR system’s performance for searching songs from six different topics

# Chapter 8

## Experiments

---

### Abstract

This chapter describes the experiments we performed using the parameter set developed earlier. We explain the experimental setup and the evaluation measure we had to develop specifically for this application. We conclude by discussing the results and showing how the space of our songs is structured according to our methodology.

### 8.1 Setup

Stages 1-4 of the described algorithm (see Chapter 5) were applied to our lyrics database with the settings  $k = 60$  and  $f_{\max} = \infty$ , derived from the experiments in Section 7.2. The resulting 60 clusters were used as the basis for a two-phased experiment that was inspired by the delphi method (Dalkey and Helmer, 1963) which is a common method for obtaining opinion consensus from a group of experts.

In the first phase, test subjects were shown the most important terms (as explained in Chapter 6) of each cluster and were asked to provide tags that summarise the terms.

In the second phase, the same word lists were shown to the same test subjects, but this time the task was to choose the best tags from those collected during the first phase, and not more than two. These tests were carried out with 6 subjects, all male between 20 and 32 years of age with strong background in computer science and little in music.

```

1  god    lord    soul    heaven  thank  pray
7  jesus  help    holy    earth   hands  hope
13 damn   king    glory   praise  peace  bless
19 dead   child   father  born    black  stand
25 word   grace   death   friend  power  lift
Please take some time to read the terms. Continue with [ENTER]

```

```

1 god
2 gospel
3 prayer
4 religion
Please choose the best tag(s) for the cluster, at most 2, best
first, separated by whitespace.
If no tag fits the cluster, leave empty. 

```

```

How well do these tags describe the cluster?
(1=well, 2=not well, not badly,3=badly): 

```

Example 8.1.1: Choice of tags and quality assessment of an NMF cluster.  
User input is boxed.

## 8.2 Evaluation Measure

The strength of agreement among test subjects cannot be measured after the first phase because they are completely free in their production of tags, so it is very unlikely that identical tags be used. In phase 2, when the subjects have to choose from the tags produced during phase 1, this is possible because all perform identical tasks. For estimating the significance of the labeling outcome, we compute the probability of the actual result being attained by completely random behaviour on behalf of the subjects. The rationale is similar to that of methods for assessing inter-coder agreement: The lower this probability, the more evidence there is that the result is due to intelligible features of the data.

During phase 2, there was a given number of tags ( $m$ ) associated with a given cluster<sup>1</sup>. If a person chose a tag at first position, we assigned a grade of 1 to that tag. If the person chose it for the second position, the grade was 2, all other tags were assigned grade 3. Thus, In the whole session, a ( $n \times m$ )

<sup>1</sup>There were at least 2 tags for each cluster, at most 10; mean tag count per cluster was 6.25.

grading matrix was created containing the grades for all  $m$  tags created by all  $n$  test subjects.

The behaviour of the subjects was modeled as follows: For a given cluster, i.e., for a given  $m$ , a subject could choose one of the tags as the best tag with a probability of  $P(\textit{first}) = p_1$  and chose none with probability  $P(\textit{none}) = 1 - p_1$ . If a tag was chosen best, each tag was equally probable to be chosen with a probability of  $1/m$ . Then, the person could pick another tag as second best with probability  $P(\textit{second}|\textit{first}) = p_2$  and no second best tag with probability  $P(\textit{nosecond}|\textit{first}) = 1 - p_2$ . If a tag was chosen as second best, again, all tags were equally probable for this choice with probability  $1/(m - 1)$ .

The model parameters  $p_1$  and  $p_2$  are computed based on the behaviour of the test subjects.  $p_1$  is defined as the percentage of cases in which at least one tag was chosen,  $p_2$  as the percentage of the former cases in which also a second tag was picked.

Consequently, given  $m$ ,  $p_1$  and  $p_2$ , the probability  $p(g)$  for a tag to get grade  $g \in \{1, 2, 3\}$  is

$$p(1) = \frac{p_1 p_2}{m} + \frac{p_1(1-p_2)}{m} \quad (8.1)$$

$$p(2) = \frac{p_1 p_2}{m} \quad (8.2)$$

$$p(3) = \frac{p_1 p_2(m-2)}{m} + \frac{p_1(1-p_2)(m-1)}{m} + 1 - p_1 \quad (8.3)$$

Now, as the result of phase 2, we get a grading matrix for each cluster. The strength of agreement is assessed by computing how probable the column means of such a matrix are a priori. As each value in each column takes the values 1,2 or 3 with the probabilities explained above, the probability for a column to contain  $g_1$  times the grade 1,  $g_2$  times 2 and  $g_3$  times 3 is

$$\frac{n!}{g_1!g_2!g_3!} p(1)^{g_1} p(2)^{g_2} p(3)^{g_3} \quad (8.4)$$

The likelihood of reaching a given column mean (i.e., average grade for a tag) is the sum of the probabilities of all grade combinations that result in the same or a better mean.

### 8.3 Results and Discussion

The result of the assesment procedure is a number of tags for each cluster. The tags are associated with a grade and a value which indicates the likelihood for the grade to result from chance agreement among subjects.

Table 8.1 shows the best-graded tag of each cluster, provided that the said likelihood is at most 10%. Due to this selection criterion, only 41 out of 60 tags are shown. This table should provide an overview of the tags that can be used to index the song collection. It is far from covering all the actual topics of the songs, but it shows a certain diversity. Figure 8.1 shows the levels of significance for the most popular tag of each cluster in 5% steps. In 31 out of 60 clusters, the best tag was agreed on with a less than 5% likelihood of chance agreement. For another 10 clusters, the significance level was between 5% and 10%. These results suggest that a reasonable portion of the clusters describes discernable topics and that they are reliably tagged.

appearance	boys_and_girls	boys_and_girls	broken_hearted
clubbing	conflict	crime	dance
dance	depression	dream	dream
emotion	emotion	family	feelings
future	gangsta	gangsta	gangsta
gangsta	going_out	gospel	hard_times
hiphop	home	leave	listen
loneliness	loss	love	love
love	music	music	nature
party	sorrow	talk	weather
world			

Table 8.1: Winning tags at a significance level of 10%

Figure 8.1: Number of clusters per significance level of the winning tag

The presented findings are an adequate assessment of the quality of individual topics. They fail, however, to give an impression of the big picture, the effects of applying our algorithm to the whole corpus. This is why we try to shed some light on the structure our method imposes on the song collection in the following two sections.

### 8.3.1 Relationships among topics

In order to explore the relationships among the topic clusters, we visualize them in a dendrogram (as described in cp. Section 6.2). In this case, we use a significance level of 30% so as not to have too many clusters without any tag.<sup>2</sup> It is important to keep in mind that the topic's similarities are computed based on the membership vectors of all the songs in the corpus, not on the term weights for each of the topics. Hence, similarity here reflects similar weighting of the the topics in the songs' topic membership vectors. The dendrogram is depicted in Figure 8.2. The cluster ids set in square brackets relate to the cluster ids in Appendix B.

Several things can be noted about our method, judging from the dendrogram:

- *Plausibility.* Many of the clusters that are combined early (i.e., far to the right) have similar tags. See, e.g., clusters 19 (“*gangsta*” “*aggressive*” “*hiphop*”) and 15 (“*hiphop*”) or clusters 25 (“*music*” “*dance*”) and 12 (“*music*” “*clubbing*” “*party*”). This suggests that the tags are meaningful with respect to the whole song collection, and not only with respect to the topic they are assigned to as suggested by the evaluation measure<sup>3</sup>.
- *Hierarchy.* In those cases where clusters have similar tags the dendrogram provides a means for combining similar clusters hierarchically. For future development, doing so could greatly simplify a user interface with respect to usability.
- *Related Topics.* In many cases, the dendrogram shows which distinct topics are related in the songs. For example, clusters 29 (“*leaving*”) and 3 (“*feelings*” “*sorrow*”) are combined with 56 (“*home*”), which seems rather plausible. Clusters 32 (“*gangsta*”) and 5 (“*conflict*”) are combined with 55 (“*talk*” “*friends*” “*trust*”), 17 (“*party*” “*clubbing*”),

---

<sup>2</sup>It is actually not necessary to raise the significance level as the clusters that do not get any tags can simply be left out.

<sup>3</sup>Note that the evaluation measure is local, i.e., only allows drawing conclusion about the tags of a single topic cluster

49 and 1 (both “*love*”). This example, too, is tractable for someone who knows some gangsta hiphop songs, which mostly talk about crime, cars, money, trust among friends or the gang, partying and women (maybe the order should be reversed in this list, though).

Figure 8.2: Dendrogram of the final clusters with user-provided labels

### 8.3.2 Relationships among tags

Most of the clusters are labeled with more than one tag, and some tags appear in more than one label. It is therefore sensible to distinguish between clusters and tags, allowing for a separate analysis of the tag space. This view of our system is especially useful with respect to possible future developments of a user interface that allows for searching songs. Such a program would be more usable if it allowed for manipulating requests on the tag level instead of the cluster level.

We visualize the tags in a graph in which each node represents a tag. Similar tags are connected, the font size of the tag reflects its importance in the corpus. This is shown in Figure 8.3. For creating this visualization, we first compute the topic affiliation vector for each song, creating a matrix  $A_{k \times \text{documents}}$ ,  $k$  being the number of topic clusters. Next, the column vectors of  $A$  are normalized to a length of 1, so that each song can be interpreted as a combination of topics, which yields another matrix  $\bar{A}$ . Based on  $\bar{A}$ , a second matrix  $B_{t \times \text{documents}}$  is created,  $t$  being the number of distinct tags in the cluster labels, assuming a significance threshold of 30%. Each row in  $B$ , describing a tag, is now defined as the average over the rows in  $\bar{A}$  the topic labels of which contain the respective tag. This procedure yields a description of all tags by the membership strength of all documents in the topics labeled with the tags. We compute the complete cosine similarity matrix  $S_{t \times t}$  and filter the similarity values such that only the four most similar tags are used. Based on  $S$ , we use the R package *igraph* (Csardi and Nepusz, 2006) to create the graph visualization.<sup>4</sup> The font sizes of the tags are assigned based on the ordering of the row sums of  $B$ , i.e., the tag size only reflects ordering, not proportionality.

The first thing to note about this method is that the tags that co-occur in topic labels have very similar, if not identical representations in the matrix  $B$ . Therefore they reach high similarity values and are thus placed next to each other in the graph. For example, the tags “*talk*”, “*friends*” and “*trust*” co-occur in the label of cluster 55 and they are grouped together in the graph. This shows that there is a certain bias to the analysis that should not be forgotten. However, a number of the relations depicted in the graph do not derive from co-occurrence or transitive co-occurrence of tags in labels but from the topic memberships of the songs. For example, “*weekend*” and “*going\_out*” form the label of cluster 14 and do not co-occur with any other tags. They are nevertheless placed near “*party*” and “*dance*” in the graph view.

From the graph view, the following observations can be made:

---

<sup>4</sup>The layout function we apply is `layout.graphopt(niter=10000, charge=0.05)`

- *Tag Importance.* The relative importance of the tags is shown by the font size of the tags. Though the length of the tag plays an important role in the perceived size, the important tags can easily be distinguished from the less important ones. From looking at the graph, it is instantly clear that a lot of songs are either “*hiphop*” (or “*aggressive*” “*gangsta*” “*hiphop*”), about “*love*”, “*relations*”, “*party*”, “*religion*” or “*loss*”. The less important tags are often modifiers of the important ones, e.g., for “*love*”, the related tags are “*broken\_harted*”, “*hope*”, *topicromance*, “*party*”.
- *Tag Connectivity.* One may expect important tags to have more connections than less important ones. While this is generally the case, counter-examples can be found. For example, “*love*”, the most important tag, has only five relations to other tags, whereas “*depression*”, much less important, has many. This may be an effect of our test subjects being more precise about depressive topics than about romantic ones. However, a relatively low node degree with respect to importance indicates that the tag is very important for many songs compared to other tags. Consequently, one may regard these tags as having a high descriptive value compared to other tags with similar importance and higher node degree.
- *Regions.* The graph visualization can be partitioned into regions containing coherent tags. On the right hand side, we find tags concerning relations, friendship, love and uncertainty. The lower and left part of the graph is about negative feelings. Positively connotated tags can be found in the upper region, where also tags expressing elation and aggressiveness are positioned.
- *Topical Bias.* As a result of seeing the tags in regions, it is possible to detect bias in the topics identified by our algorithm. It is quite obvious, for example, that there is a strong leaning toward negative emotions as opposed to positive ones.

Figure 8.3: Graph visualization of the tags. Similar tags are connected. The size of a tag reflects its importance

## Summary

In this chapter, we present a small evaluation which on the one hand indicates that our method works and on the other hand produces labels for the identified topics. Using the results of the evaluation, we show the structure that our method imposes onto the song collection by two visualizations, a cluster dendrogram and a graph visualization.

# Chapter 9

## Conclusion and Future Work

---

The work at hand explains the structure and parametrization of an algorithm for the application of NMF to song lyrics. The focus is on showing the distinct stages of the algorithm and the considerations concerning the choice of parameter values for each stage. The most interesting choices for parameter values, in our view, are a) the high value for  $k$  (60 may still not be high enough) and b) the use of binary weighting prior to NMF clustering.

We also present an assessment of the clustering outcome indicating that most of the topic clusters resulting from our algorithm are useful for indexing our music collection. The procedure used for assessment is at the same time an integral part of the algorithm, the labeling stage, which has the convenient property that a statistically interpretable confidence value is calculated for each cluster so that it can be rejected or accepted for use in the subsequent stages.

For future work concerned with the application of our method to a real world music browsing system, the Section 8.3.1 and especially Section 8.3.2 provide a good starting point. The graph visualization presented in the latter section is well suited as a blueprint for creating a tag cloud-like user interface for querying a song database. In this context it is most convenient that the “tag cloud” can be used for searching by weighting tags as well as for displaying the tag affiliation of the resulting songs.

It should be noted that many elements of the proposed method can be optimized or replaced. For example, a more effective combination of stop-word detection and chunking may allow for filtering terms such as “*won*”, “*wouldn*”, “*didn*” etc., which are currently not removed from the corpus. Another possible adaptation may be to use weighting more efficiently: it could make sense to use  $TF \times IDF$  weighting on the document-to-topic affiliation matrix in order to downweight the common topics. Yet another way of improving the results could consist in finer-grained clustering, i.e., the generation of more topics. This may be achieved just by raising the value for  $k$  in the NMF clustering step but it is more probable that hierarchical clustering would have to be performed in order to find less widely spread topics. In addition to that, the question arises how similar clusters can be combined or hierarchically arranged for display to end users.

Taking the whole approach one step further, using NMF may turn out to be only one approach among many for creating weighted word lists (i.e., clus-

ters) for indexing song collections. We consider the use of term co-occurrences most promising for the semi-manual definition of such lists for topics that are known to be contained in the archive but are not found by clustering.

# List of Figures

3.1	Plot of precision and recall of Lyrics Clipping for every single test case. . . . .	32
3.2	Boxplots showing the distribution of precision and recall values of Lyrics Clipping. . . . .	32
4.1	Distribution of genres in the corpus . . . . .	37
4.2	Distribution of the document lengths in the corpus . . . . .	38
4.3	Average length of lyrics grouped by genre . . . . .	39
4.4	Distribution of the confidence value in the corpus . . . . .	40
4.5	Joint Distribution of confidence and length . . . . .	40
5.1	Term selection in the TDM . . . . .	44
5.2	Changing the weighting of the TDM . . . . .	44
5.3	Factorization of the TDM . . . . .	45
5.4	Labeling of the NMF clusters . . . . .	46
5.5	Computation of the documents' affiliation strength to the clusters . . . . .	46
5.6	Query definition and document retrieval . . . . .	47
6.1	Cluster affiliation of a song . . . . .	54
8.1	Number of clusters per significance level of the winning tag . .	73
8.2	Dendrogram of the final clusters with user-provided labels . .	76
8.3	Graph visualization of the tags . . . . .	79

# List of Tables

3.1	Section of a lyrics alignment . . . . .	26
3.2	Songs used for evaluation of lyrics clipping . . . . .	30
3.3	Mean and standard deviation for precision, recall and processing time of Lyrics Clipping . . . . .	31
7.1	The evaluation set used in systematic optimization . . . . .	64
7.2	Results of quality evaluation . . . . .	67
7.3	Results of quality evaluation: effect of $f_{\max}$ . . . . .	67
7.4	Assessment of the IR system's performance for searching songs from six different topics . . . . .	69
8.1	Winning tags at a significance level of 10% . . . . .	73
A.1	List of lyrics-specific stopwords . . . . .	92

# List of Examples

6.1.1 Most important terms of a cluster . . . . .	48
6.1.2 Plot of term weights . . . . .	49
6.2.1 Dendrogram of clusters . . . . .	51
6.3.1 Lyrics found for “love” and “loss” . . . . .	53
7.1.1 Text erroneously identified as lyrics . . . . .	56
7.1.2 Very short lyrics . . . . .	56
7.1.3 Top 30 terms of a cluster which contains mostly strong co- occurents of the strongest term, “lie” . . . . .	62
7.2.1 Tagging and quality assessment of an NMF cluster . . . . .	65
8.1.1 Choice of tags and quality assessment of an NMF cluster . . . . .	71

# Bibliography

- Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley.
- Bainbridge, D., Cunningham, S. J., and Downie, J. S. (2003). How people describe their music information needs: A grounded theory analysis of music queries. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR '03)*, Baltimore, Maryland , USA.
- Baumann, S. and Halloran, J. (2004). An ecological approach to multimodal subjective music similarity perception. In *Proceedings of the 1st Conference on Interdisciplinary Musicology (CIM '04)*, Graz, Austria.
- Baumann, S. and Hummel, O. (2003). Using cultural metadata for artist recommendations. In *Proceedings of the 3rd International Conference on Web Delivering of Music (WEDELMUSIC '03)*, pages 138–141, Leeds, UK.
- Baumann, S. and Klüter, A. (2002). Super convenience for non-musicians: Querying mp3 and the semantic web. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR '02)*, pages 297–298, Paris, France.
- Baumann, S., Pohle, T., and Vembu, S. (2004). Towards a socio-cultural compatibility of mir systems. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR '04)*, Barcelona, Spain.
- Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173.
- Brochu, E. and de Freitas, N. (2003). “name that song!” a probabilistic approach to querying on music and text. In S. Becker, S. T. and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 1505–1512, Cambridge, MA. MIT Press.
- Brochu, E., de Freitas, N., and Bao, K. (2003). The sound of an album cover: Probabilistic multimedia and IR. In Bishop, C. M. and Frey, B. J., editors, *Proceedings of 9th International Workshop on Artificial Intelligence and Statistics*, Key West, USA.

- Cohen, W. W. and Fan, W. (1999). Web-collaborative filtering: recommending music by crawling the Web. *Computer Networks*, 33(1–6):685–698.
- cois Pachet, F., Westermann, G., and Laigre, D. (2001). Musical data mining for electronic music distribution. In *WEDELMUSIC '01: Proceedings of the First International Conference on WEB Delivering of Music (WEDELMUSIC'01)*, page 101, Washington, DC, USA. IEEE Computer Society.
- Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering. *Nucl. Acids Res.*, 16(22):10881–10890.
- Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.
- Dalkey, N. and Helmer, O. (1963). An experimental application of the delphi method to the use of experts. *Management Science*, 9(3):458–467.
- 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 *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 488–491, London, UK.
- Ellis, D. P. W., Whitman, B., Berenzweig, A., and Lawrence, S. (2002). The quest for ground truth in musical artist similarity. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR '02)*, pages 170–177, Paris, France.
- Feinerer, I. (2007). *tm: Text Mining Package*. R package version 0.2-3.7.
- Frederico, G. C. S. (2002). Actos: a peer-to-peer application for the retrieval of encoded music. In *Proceedings of the 1st International Conference on Musical Application Using XML (MAX '02)*, Milan, Italy.
- Geleijnse, G. and Korst, J. (2006a). Efficient lyrics extraction from the web. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR '06)*, pages 371 – 372, Victoria, Canada.
- Geleijnse, G. and Korst, J. (2006b). Web-based artist categorization. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, pages 266 – 271, Victoria, Canada.

- Heyer, G., Quasthoff, U., and Wittig, T. (2006). *Text Mining: Wissensrohstoff Text*. W3L-Verlag, Herdecke, Bochum.
- Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pages 289–296, Stockholm, Sweden.
- Kleedorfer, F., Harr, U., and Krenn, B. (2007). Making large music collections accessible using enhanced metadata and lightweight visualizations. In *Proceedings of the 3rd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS '07)*, pages 138–144, Barcelona, Spain.
- Klüter, A., Baumann, S., and Norlien, M. (2002). Using natural language input and audio analysis for a human-oriented mir system. In *Proceedings of the First International Symposium on Cyber Worlds (CW'02)*, page 74, Washington, DC, USA. IEEE Computer Society.
- Knees, P. (2008). Addendum to “Multiple Lyrics Alignment: Automatic Retrieval of Song Lyrics”. Technical Report CPJKU-TR-2008-MLA, Dept. of Computational Perception, Johannes Kepler University, Linz, Austria.
- Knees, P., Pampalk, E., and Widmer, G. (2004). Artist classification with web-based data. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR '04)*, pages 517–524, Barcelona, Spain.
- Knees, P., Schedl, M., Pohle, T., and Widmer, G. (2006). An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web. In *Proceedings of the ACM Multimedia 2006 (MM'06)*, pages 17–24, Santa Barbara, California, USA.
- Knees, P., Schedl, M., and Widmer, G. (2005). Multiple lyrics alignment: Automatic retrieval of song lyrics. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 564–569, London, UK.
- Korst, J. and Geleijnse, G. (2006). Efficient lyrics retrieval and alignment. In Verhaegh, W., Aarts, E., ten Kate, W., Korst, J., and Pauws, S., editors, *Proceedings Third Philips Symposium on Intelligent Algorithms (SOIA '06)*, pages 205 – 218, Eindhoven, the Netherlands.
- Lee, D. D. and Seung, S. H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

- Li, T. and Ogihara, M. (2004a). Music artist style identification by semi-supervised learning from both lyrics and content. In *Proceedings of the 12th ACM International Conference on Multimedia (MULTIMEDIA '04)*, pages 364–367, New York, NY, USA. ACM Press.
- Li, T. and Ogihara, M. (2004b). Semi-supervised learning for music artists style identification. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM '04)*, pages 152–153, New York, NY, USA. ACM.
- Logan, B., Kositsky, A., and Moreno, P. (2004). Semantic analysis of song lyrics. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo*, volume 2, pages 827–830, Baltimore, Maryland, USA.
- MacLellan, D. and Boehm, C. (2000). Mutated'll: A system for music information retrieval of encoded music. In *Proceedings of the 1st International Conference on Music Information Retrieval (ISMIR 2000)*, Plymouth, Massachusetts, USA.
- Mahedero, J. P. G., Martínez, A., Cano, P., Koppenberger, M., and Gouyon, F. (2005). Natural language processing of lyrics. In *Proceedings of the 13th ACM International Conference on Multimedia (MULTIMEDIA '05)*, pages 475–478, New York, NY, USA. ACM Press.
- Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.
- Neumayer, R. (2007). ATLANTIS or towards a multi-modal approach to music information retrieval and its visualisation. Master's thesis, Vienna University of Technology, Department of Software Technology and Interactive Systems.
- Neumayer, R. and Rauber, A. (2007a). Integration of text and audio features for genre classification in music information retrieval. In *Proceedings of the 29th European Conference on Information Retrieval (ECIR'07)*, pages 724–727, Rome, Italy.
- Neumayer, R. and Rauber, A. (2007b). Multi-modal music information retrieval - visualisation and evaluation of clusterings by both audio and lyrics. In *Proceedings of the 8th Conference Recherche d'Information Assistée par Ordinateur (RIAO '07)*, Pittsburgh, PA, USA. ACM.

- Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.
- Pampalk, E. and Goto, M. (2006). Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR '06)*, pages 367–370, Victoria, Canada.
- Pampalk, E. and Goto, M. (2007). Musicsun: A new approach to artist recommendation. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR '07)*, pages 101–104, Vienna, Austria.
- Penaranda, J. (2007). Text mining von songtexten. Master's thesis, Vienna University of Technology, Department of Software Technology and Interactive Systems.
- Pohle, T., Knees, P., Schedl, M., and Widmer, G. (2007). Meaningfully Browsing Music Services. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR '07)*, pages 115–116, Vienna, Austria.
- R Development Core Team (2006). R: A language and environment for statistical computing. ISBN 3-900051-07-0.
- Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.
- Scott, S. and Matwin, S. (1998). Text classification using wordnet hypernyms. In *Workshop on usage of WordNet in NLP Systems (COLING-ACL '98)*, pages 45–51.
- Wei, B., Zhang, C., and Ogihara, M. (2007). Keyword generation for lyrics. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR '07)*, pages 121–122, Vienna, Austria.
- Whitman, B. and Lawrence, S. (2002). Inferring descriptions and similarity for music from community metadata. In *Proceedings of the International Computer Music Conference (ICMC)*, Göteborg, Sweden.
- Whitman, B. and Smaragdis, P. (2002). Combining musical and cultural features for intelligent style detection. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR '02)*, pages 47–52, Paris, France.

- Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '03)*, pages 267–273, New York, NY, USA. ACM.

# Appendix A

## Stopwords

---

During the development of our algorithm, we collected a number of words we deemed useless for our purposes. They were summarized in a lyrics-specific stopwords list and removed from the TDM in the term selection stage. Table A.1 shows all those stopwords.

aah	dup	na	uhmm
ad	fi	net	uhn
adclicker	google	nuh	uhun
adcock	hah	oh	um
addlink	hee	ohh	uma
ah	heh	online	umi
ahh	hey	oo	umm
ahh	ho	ooh	ummm
array	hoo	ooo	ummmmmm
aw	hosting	ooowe	ummmmmm
aww	hotmail	password	un
awww	html	script	upload
awwww	huh	songtext	url
awwww	inurl	songtexte	uuh
awww	javascript	text	verse
awww	la	trojan	wee
cgi	liedertext	uggh	whoo
chorus	liedertexte	uggh	wop
com	lla	ugh	ya
da	login	ughhh	ye
de	ltd	ugy	yeah
dee	lyric	uh	yo
di	lyric	uh	yoo
dll	lyrics	uhh	yuh
do	mp3	uhhh	
download	mpg	uhhhh	
dub	myspace	uhhuh	

Table A.1: List of lyrics-specific stopwords

# Appendix B

## Detailed Experimental Results

---

This appendix contains the raw data collected during our experiments. The data is organized by topic cluster. For each cluster, we list the most important terms<sup>1</sup>, the data collected during the first round of the experiment and the data collected during the second round<sup>2</sup>. These data are arranged in tabular format. At the bottom of each page, the tags describing the cluster are shown for the significance levels used in the experiments. In the following, we briefly describe the gathered material.

### Most Important Terms

The terms are in tabular format which has to be read line by line; the most important term is the at the top left position.

### Data from the First Round

The test subjects were asked to provide tags to describe the cluster along with a quality measure  $\in 1, 2, 3$ , 1 being the best grade. These data are provided in no particular order.

### Data from the Second Round

At this stage of the experiment, the test subjects were asked to select an ordered set of the best tags from those provided by all subjects in the first round. The maximum number of tags they could chose was 2. We attributed the grade 1 to the tag selected first, 2 for second place, and 3 for all others. For each cluster, a table is constructed containing the grades thus assigned by each test subjects the columns labeled “1”-”5”. Next to the grading data, the average grade for each tag and the prior probability with which this average is attained are shown.<sup>3</sup>

---

<sup>1</sup>The method for selecting the most important terms is described in Section 6.1.

<sup>2</sup>The structure of our experiments is explained in Chapter 8.

<sup>3</sup>See Chapter 8 for an explanation how this probability is computed.

## Tags for Clusters by Significance Level

For both significance levels used in the experiments, 10% and 30%, the respective tags are shown. A tag passes for a given significance level if the prior probability of its average rating is below the respective percentage.

## Cluster #1

baby	crazy	babe	tags	quality
lovin	boy	sweet	HipHop woman	3
honey	body	touch	sex	1
repeat	lady	hot	lovers	1
shake	woman	pretty	love relations sex feel-	1
mama	whoa	goin	ings	
ooh	feelin	till	gnagnagna	1
bout	loving	slow	love dance hot	1
roll	money	sugar		
daddy	move	mmm		

Table B.2: Tags and quality provided in first round

Table B.1: Most important terms in the cluster

	1	2	3	4	5	means	probability
dance	3	3	2	3	3	2.80	0.579
feelings	3	3	3	3	3	3.00	1.000
gnagnagna	3	3	3	3	3	3.00	1.000
hiphop	1	3	3	3	3	2.60	0.420
hot	3	3	3	2	3	2.80	0.579
love	3	1	1	3	2	2.00	0.021
lovers	3	3	3	3	3	3.00	1.000
relations	3	3	3	3	1	2.60	0.420
sex	3	3	3	1	3	2.60	0.420
woman	3	3	3	3	3	3.00	1.000

Table B.3: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: love

Tags for significance level 30%: love

## Cluster #2

maybe	change	hope	tags	quality
crazy	pretty	didn		3
someday	chance	waiting	resurrection	3
mean	understand	sometimes	hope change future	1
wish	thinking	fly	unsure doubt	2
save	trying	share	soft mind	1
looking	goodbye	sleep	maybe	2
lady	fight	ass		
standing	wait	days		
tomorrow	lonely	living		

Table B.5: Tags and quality provided in first round

Table B.4: Most important terms in the cluster

	1	2	3	4	5	means	probability
change	2	3	3	3	1	2.40	0.188
doubt	3	2	3	1	3	2.40	0.188
future	1	3	2	3	2	2.20	0.102
hope	3	1	1	3	3	2.20	0.102
maybe	3	3	3	3	3	3.00	1.000
mind	3	3	3	3	3	3.00	1.000
resurrection	3	3	3	3	3	3.00	1.000
soft	3	3	3	3	3	3.00	1.000
unsure	3	3	3	3	3	3.00	1.000

Table B.6: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: future hope change doubt

## Cluster #3

please	help	comes	tags	quality
knees	lord	remember		2
door	forgive	bring	everyday_life	2
sorry	understand	honey	waiting_for_help	2
wait	hate	trying	feelings pain mistaken	1
mean	daddy	promise	sorrow relations	
darling	clothes	begging		2
bathroom	bad	friends	sorry	1
tired	hurt	fun		
closet	floor	round		

Table B.8: Tags and quality provided in first round

Table B.7: Most important terms in the cluster

	1	2	3	4	5	means	probability
everyday_life	3	3	3	3	1	2.60	0.506
feelings	2	1	3	3	2	2.20	0.127
mistaken	3	3	3	3	3	3.00	1.000
pain	1	3	3	3	3	2.60	0.506
relations	3	3	3	3	3	3.00	1.000
sorrow	3	2	3	1	3	2.40	0.228
sorry	3	3	1	3	3	2.60	0.506
waiting_for_help	3	3	3	3	3	3.00	1.000

Table B.9: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: feelings sorrow

## Cluster #4

inside	deep	hide	tags	quality
feeling	outside	ride	feelings	3
pain	burn	skin	red-hot	1
close	burning	waiting	inside_outside feelings	2
watch	sky	seen	action outside love	2
fly	alive	matter	soul	1
lie	lonely	heaven	inside	2
alright	voice	makes		
fire	touch	soul		
feels	step	hurt		

Table B.11: Tags and quality provided in first round

Table B.10: Most important terms in the cluster

	1	2	3	4	5	means	probability
action	3	3	3	3	3	3.00	1.000
feelings	1	1	1	3	3	1.80	0.016
inside	3	3	3	3	3	3.00	1.000
inside_outside	3	3	3	1	1	2.20	0.127
love	3	3	3	2	3	2.80	0.670
outside	3	3	3	3	3	3.00	1.000
red-hot	3	3	3	3	3	3.00	1.000
soul	2	2	3	3	3	2.60	0.506

Table B.12: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: feelings

Tags for significance level 30%: feelings inside\_outside

## Cluster #5

wrong	bad	strong	tags	quality
told	mean	fight	feelings	3
trying	doesn	matter	brawl	1
help	feeling	belong		3
break	thinking	late	pain cheat feelings	2
didn	sorry	stand	dark	1
lie	wait	wish	conflict	2
song	white	friend		
hurt	dont	thats		
change	aint	care		

Table B.14: Tags and quality provided in first round

Table B.13: Most important terms in the cluster

	1	2	3	4	5	means	probability
brawl	3	3	3	3	3	3.00	1.000
cheat	3	3	3	1	3	2.60	0.631
conflict	1	1	1	2	1	1.20	0.000
dark	3	3	3	3	3	3.00	1.000
feelings	2	3	2	3	2	2.40	0.354
pain	3	2	3	3	3	2.80	0.785

Table B.15: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: conflict

Tags for significance level 30%: conflict

## Cluster #6

hear	sound	listen	tags	quality
calling	voice	music		3
heard	words	hope	open_air_concert	2
near	radio	touch	music	1
sayin	saying	hand	music listen sound	1
voices	fear	body		2
sky	sounds	loud	sound	1
speak	whisper	wind		
talking	bells	coming		
help	falling	mother		

Table B.17: Tags and quality provided in first round

Table B.16: Most important terms in the cluster

	1	2	3	4	5	means	probability
listen	1	1	3	1	1	1.40	0.016
music	2	3	2	3	2	2.40	0.603
open_air_concert	3	3	3	3	3	3.00	1.000
sound	3	3	1	2	3	2.40	0.603

Table B.18: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: listen

Tags for significance level 30%: listen

## Cluster #7

cuz	dont	aint	tags	quality
cant	wit	boy	hipHop	2
hook	thats	bout	gangsta	1
tha	gettin	yall		3
body	niggas	repeat	hiphop	2
lil	goin	yea	chaos	1
hands	ready	bitch	hiphop	1
touch	mean	ur		
dat	wont	crazy		
tryna	nothin	damn		

Table B.20: Tags and quality provided in first round

Table B.19: Most important terms in the cluster

	1	2	3	4	5	means	probability
chaos	3	3	3	3	3	3.00	1.000
gangsta	1	3	1	3	3	2.20	0.640
hiphop	2	1	2	1	3	1.80	0.264

Table B.21: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: *(None)*

Tags for significance level 30%: hiphop

## Cluster #8

ve	seen	found	tags	quality
times	waiting	tried	feelings	3
heard	thinking	lose	broken_love	1
bad	friends	coming		3
learned	feeling	lot	relations	3
guess	words	broken	search	1
looking	loved		loss	2

Table B.22: Most important terms in the cluster

Table B.23: Tags and quality provided in first round

	1	2	3	4	5	means	probability
broken_love	3	3	3	2	3	2.80	0.852
feelings	3	3	3	3	3	3.00	1.000
loss	1	3	1	1	3	1.80	0.064
relations	3	1	3	3	1	2.20	0.293
search	3	2	3	3	3	2.80	0.852

Table B.24: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: loss

Tags for significance level 30%: loss relations

## Cluster #9

look	lookin	looking	tags	quality
eye	roll	smile	hipHop	3
top	looks	ride	party	1
money	hair	black	face appearance	1
door	girls	town	crush looks superficial	1
alive	friend	hook	sense	1
mama	sound	pretty	looks	1
front	watch	beautiful		
gettin	woman	looked		
aint	cos	half		

Table B.26: Tags and quality provided in first round

Table B.25: Most important terms in the cluster

	1	2	3	4	5	means	probability
appearance	1	3	1	2	1	1.60	0.003
crush	3	3	3	3	3	3.00	1.000
face	2	3	3	3	3	2.80	0.670
hiphop	3	3	3	3	3	3.00	1.000
looks	3	2	3	1	3	2.40	0.228
party	3	3	3	3	3	3.00	1.000
sense	3	1	3	3	3	2.60	0.506
superficial	3	3	3	3	3	3.00	1.000

Table B.27: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: appearance

Tags for significance level 30%: appearance looks

## Cluster #10

time	line	remember
wait	waste	change
times	goodbye	goes
waiting	days	move
wasting	rhyme	

Table B.28: Most important terms in the cluster

tags	quality
	3
rap	2
time	1
time past_and_future	1
ongoing change devel-	
oping	
memory	2
past	1

Table B.29: Tags and quality provided in first round

	1	2	3	4	5	means	probability
change	1	3	3	3	3	2.60	0.506
developing	3	3	3	1	3	2.60	0.506
memory	3	1	3	3	3	2.60	0.506
ongoing	3	3	3	2	3	2.80	0.670
past	2	3	3	3	3	2.80	0.670
past_and_future	3	3	3	3	1	2.60	0.506
rap	3	3	3	3	3	3.00	1.000
time	3	3	1	3	2	2.40	0.228

Table B.30: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: time

## Cluster #11

cause	dont	goin
aint	alright	bout
mic	ready	makes
getting	cant	rhymes
girls		

Table B.31: Most important terms in the cluster

tags	quality
hipHop	3
battle	2
	3
	3
sing	1
sing	2

Table B.32: Tags and quality provided in first round

	1	2	3	4	5	means	probability
battle	3	3	3	3	3	3.00	1.000
hiphop	1	3	1	1	3	1.80	0.264
sing	3	1	3	3	3	2.60	0.925

Table B.33: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: hiphop

## Cluster #12

play	game	rock	tags	quality
music	boy	dance	party electronic	2
games	guitar	hit	clubbing	1
roll	break	watch	music fun party	1
cool	pay	money	music dance sound	1
ready	dj	girls	square	1
beat	streets	radio	rock_n_roll party	1
gettin	fool	wit		
lookin	band	sound		
rap	tryin	played		

Table B.35: Tags and quality provided in first round

Table B.34: Most important terms in the cluster

	1	2	3	4	5	means	probability
clubbing	3	1	3	3	1	2.20	0.102
dance	3	3	3	3	3	3.00	1.000
electronic	3	3	3	3	3	3.00	1.000
fun	3	3	3	3	2	2.80	0.622
music	2	2	1	1	3	1.80	0.012
party	1	3	2	3	3	2.40	0.188
rock_n_roll	3	3	3	3	3	3.00	1.000
sound	3	3	3	2	3	2.80	0.622
square	3	3	3	3	3	3.00	1.000

Table B.36: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: music

Tags for significance level 30%: music clubbing party

## Cluster #13

hard	times	trying	tags	quality
help	sometimes	getting	hipHop	3
easy	money	boy	car accident	2
days	understand	living	hard_times help	1
tried	cold	tryin	hard_times mischance	2
thats	bad	goodbye	depression	2
break	til	hate	hard	2
dont	hurt	ride		
boys	change	cant		
start	goin	harder		

Table B.38: Tags and quality provided in first round

Table B.37: Most important terms in the cluster

	1	2	3	4	5	means	probability
accident	3	3	3	3	3	3.00	1.000
car	3	3	3	3	3	3.00	1.000
depression	3	3	3	3	3	3.00	1.000
hard	3	3	3	3	3	3.00	1.000
hard_times	1	3	3	1	1	1.80	0.016
help	3	1	3	3	2	2.40	0.228
hiphop	3	3	3	3	3	3.00	1.000
mischance	3	3	3	2	3	2.80	0.670

Table B.39: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: hard\_times

Tags for significance level 30%: hard\_times help

## Cluster #14

night	sleep	tight	tags	quality
late	dance	morning		3
comes	alright	body	disco	1
dream	music	saturday	sleep	1
lights	dark	fight	party going_out week-	1
party	town	bed	end fun action	
stars	moon	till	saturday_night	2
sight	middle	spend	party night	1
woman	rock	lonely		
dreams	close	floor		

Table B.41: Tags and quality provided in first round

Table B.40: Most important terms in the cluster

	1	2	3	4	5	means	probability
action	3	3	3	3	3	3.00	1.000
disco	3	3	3	3	1	2.60	0.459
fun	3	2	3	3	3	2.80	0.622
going_out	3	3	2	2	2	2.40	0.188
night	3	3	1	3	3	2.60	0.459
party	3	1	3	3	3	2.60	0.459
saturday_night	3	3	3	3	3	3.00	1.000
sleep	3	3	3	3	3	3.00	1.000
weekend	1	3	3	1	3	2.20	0.102

Table B.42: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: weekend going\_out

## Cluster #15

em	hit	wit	tags	quality
black	ride	roll	hipHop	1
nigga	girls	move	gangsta	1
beat	hands	gon	party clubbing	1
gettin	money	throw	hiphop party dance	2
club	boys	pop	street	
niggaz	party	block	gangsta	2
rock	lookin	game	gangsta party	1
dance	bout	watch		
floor	hook	hot		

Table B.44: Tags and quality provided in first round

Table B.43: Most important terms in the cluster

	1	2	3	4	5	means	probability
clubbing	3	3	3	3	3	3.00	1.000
dance	3	3	3	3	3	3.00	1.000
gangsta	1	3	3	3	3	2.60	0.631
hiphop	2	1	2	1	1	1.40	0.003
party	3	3	1	2	3	2.40	0.354
street	3	2	3	3	3	2.80	0.785

Table B.45: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: hiphop

Tags for significance level 30%: hiphop

## Cluster #16

cry	tears	pain	tags	quality
makes	goodbye	rain	problem	3
lie	laugh	sometimes	depression	1
remember	hurt	sad	sorrow	1
bad	bye	wonder	sad hurt love_pain	1
sky	smile	friend	family goodbye	
crying	cried	hope	sad	1
child	sleep	lonely	cry	1
dont	river	kiss		
eye	tear	mother		

Table B.47: Tags and quality provided in first round

Table B.46: Most important terms in the cluster

	1	2	3	4	5	means	probability
cry	3	3	3	3	3	3.00	1.000
depression	1	1	3	3	1	1.80	0.012
family	3	3	3	3	3	3.00	1.000
goodbye	3	3	2	3	3	2.80	0.622
hurt	2	2	3	1	3	2.20	0.102
love_pain	3	3	3	3	3	3.00	1.000
problem	3	3	3	3	3	3.00	1.000
sad	3	3	1	2	3	2.40	0.188
sorrow	3	3	3	3	3	3.00	1.000

Table B.48: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: depression

Tags for significance level 30%: depression hurt sad

## Cluster #17

wanna	ride	body	tags	quality
talk	repeat	dont		3
hook	dance	move	disco	1
free	wit	hot	party clubbing	1
hit	feelin	watch	hiphop	3
party	hate	scream	fleeing	1
club	babe	bad	party dance hot	2
goin	girls	shorty		
throw	bout			

Table B.50: Tags and quality provided in first round

Table B.49: Most important terms in the cluster

	1	2	3	4	5	means	probability
clubbing	2	3	3	1	1	2.00	0.056
dance	3	3	2	3	3	2.80	0.724
disco	3	3	3	3	3	3.00	1.000
fleeing	3	3	3	3	3	3.00	1.000
hiphop	3	3	3	3	3	3.00	1.000
hot	3	2	3	3	3	2.80	0.724
party	1	1	1	2	3	1.60	0.005

Table B.51: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: party clubbing

Tags for significance level 30%: party clubbing

## Cluster #18

walk	talk	hand	tags	quality
door	walking	dance		3
road	looking	stand	freedom	2
street	walked	streets		3
anymore	line	smile	street move walk	2
told	water	hide	moving	2
gold	chance	words	walk	2
understand	body	shoes		
floor	strong	wit		
free	days	dark		

Table B.53: Tags and quality provided in first round

Table B.52: Most important terms in the cluster

	1	2	3	4	5	means	probability
freedom	1	3	3	3	3	2.60	0.715
move	3	1	3	3	3	2.60	0.715
moving	3	3	3	3	3	3.00	1.000
street	3	3	3	2	3	2.80	0.852
walk	3	3	1	1	3	2.20	0.293

Table B.54: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: walk

## Cluster #19

			tags	quality
shit	nigga	fuck	hipHop gangster	1
bitch	ass	niggaz	gangsta	1
niggas	fuckin	wit	gangsta motherfucker	1
hit	bitches	money	hiphop sex	1
gon	hoes	dick	dirt	2
bout	talkin	game	gangsta aggressive	1
aint	lil	motherfuckin		
motherfucker	hoe	gettin		
hood	watch	damn		
pussy	club	tryin		

Table B.56: Tags and quality provided in first round

Table B.55: Most important terms in the cluster

	1	2	3	4	5	means	probability
aggressive	1	3	1	3	3	2.20	0.162
dirt	3	3	3	3	3	3.00	1.000
gangsta	2	1	3	2	1	1.80	0.024
gangster	3	3	3	3	3	3.00	1.000
hiphop	3	3	3	1	2	2.40	0.281
motherfucker	3	3	3	3	3	3.00	1.000
sex	3	3	2	3	3	2.80	0.724

Table B.57: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: gangsta

Tags for significance level 30%: gangsta aggressive hiphop

## Cluster #20

ain	nothin	gon	tags	quality
money	niggaz	tryin	hipHop	2
bout	game	nigga	gangsta	1
gettin	seen	hell	hiphop rap	1
wit	comin	goin	hiphop money street	2
cuz	lord	livin	crime	2
daddy	easy	hit	gangsta money	1
woman	country	talkin		
hook	top	trying		
hood	thinkin	house		

Table B.59: Tags and quality provided in first round

Table B.58: Most important terms in the cluster

	1	2	3	4	5	means	probability
crime	2	3	3	3	3	2.80	0.785
gangsta	1	2	1	3	1	1.60	0.010
hiphop	3	1	2	1	3	2.00	0.084
money	3	3	3	3	3	3.00	1.000
rap	3	3	3	3	3	3.00	1.000
street	3	3	3	2	3	2.80	0.785

Table B.60: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: gangsta hiphop

Tags for significance level 30%: gangsta hiphop

## Cluster #21

believe	reason	easy	tags	quality
truth	free	faith	love	3
didn	found	start	gospel	3
feeling	understand	told	feelings	some- 2
lie	chance	trying	thing_to_believe_in	
alright	repeat	breathe	promise cheat rela-	2
else	holding	meant	tions	
coming	friend	lies	reflection	1
guess	believed	happened	trust	1
knowing	pass	giving		

Table B.61: Most important terms in the cluster

Table B.62: Tags and quality provided in first round

	1	2	3	4	5	means	probability
cheat	3	3	3	1	3	2.60	0.459
feelings	3	3	3	3	1	2.60	0.459
gospel	1	1	3	3	3	2.20	0.102
love	3	3	3	3	3	3.00	1.000
promise	3	3	2	3	3	2.80	0.622
reflection	3	3	3	3	3	3.00	1.000
relations	3	3	3	2	3	2.80	0.622
something_to_believe_in	3	3	3	3	3	3.00	1.000
trust	3	2	1	3	3	2.40	0.188

Table B.63: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: gospel trust

## Cluster #22

gone	days	dead	tags	quality
strong	miss	money		3
carry	remember	forget	drugs	3
goin	blue	aint	dying	los- 3
niggas	wish	lights	ing_something	
til	hope	yesterday	past sad	2
living	late	cant	past	2
coming	summer	tomorrow	loss	1
window	song	anymore		
wake	round	train		

Table B.65: Tags and quality provided in first round

Table B.64: Most important terms in the cluster

	1	2	3	4	5	means	probability
drugs	3	3	3	3	3	3.00	1.000
dying	1	3	3	3	3	2.60	0.631
losing_something	3	3	3	3	1	2.60	0.631
loss	3	1	1	3	3	2.20	0.214
past	3	2	2	1	3	2.20	0.214
sad	3	3	3	2	3	2.80	0.785

Table B.66: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: loss past

## Cluster #23

fall	rain	stand	tags	quality
watch	break	sky	love	2
apart	falling	catch	nature	3
ground	waiting	asleep	things_that_can_fall	1
rise	tears	coming	rise_and_fall	2
knees	wall	start	wet	1
dream	arms	stars	rain	2
fell	fly	dont		
days	pieces	remember		
black	wake	hands		

Table B.68: Tags and quality provided in first round

Table B.67: Most important terms in the cluster

	1	2	3	4	5	means	probability
love	1	3	3	1	3	2.20	0.214
nature	3	1	3	2	3	2.40	0.354
rain	3	3	3	3	3	3.00	1.000
rise_and_fall	3	2	3	3	3	2.80	0.785
things_that_can_fall	3	3	1	3	3	2.60	0.631
wet	3	3	3	3	3	3.00	1.000

Table B.69: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: love

## Cluster #24

god	lord	soul	tags	quality
heaven	thank	pray	gospel	1
jesus	help	holy	gospel	1
earth	hands	hope	religion gospel god	1
damn	king	glory	religion	1
praise	peace	bles	prayer	1
dead	child	father	god	1
born	black	stand		
word	grace	death		
friend	power	lift		

Table B.71: Tags and quality provided in first round

Table B.70: Most important terms in the cluster

	1	2	3	4	5	means	probability
god	1	3	2	3	1	2.00	0.231
gospel	3	1	3	3	2	2.40	0.603
prayer	3	2	3	3	3	2.80	0.921
religion	2	3	1	1	3	2.00	0.231

Table B.72: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: god religion

## Cluster #25

song	sing	music	tags	quality
dance	rock	singing	dance	3
sweet	file	heard	concert	1
songs	move	tabs	music songs	1
bring	words	roll	music singing instru-	1
soul	radio	sound	ments sound	
floor	sad	shake	dancefloor	1
guitar	hands	blue	music	1
makes	king	praise		
loud	lord	goes		

Table B.74: Tags and quality provided in first round

Table B.73: Most important terms in the cluster

	1	2	3	4	5	means	probability
concert	3	3	3	3	3	3.00	1.000
dance	1	3	3	3	2	2.40	0.228
dancefloor	3	3	3	3	3	3.00	1.000
instruments	3	3	3	3	3	3.00	1.000
music	3	1	1	2	1	1.60	0.003
singing	2	3	2	3	3	2.60	0.506
songs	3	2	3	3	3	2.80	0.670
sound	3	3	3	1	3	2.60	0.506

Table B.75: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: music

Tags for significance level 30%: music dance

## Cluster #26

stop	rock	break	tags	quality
move	drop	top	party dance fun	1
hot	body	pop	dancing	2
beat	start	til	rock music rock_n_roll	2
dance	music	till	party dance hiphop	1
party	ready	slow	shock	1
shake	check	listen	party dance	1
floor	flow	feelin		
mic	hit	roll		
mc	coming	whoa		

Table B.77: Tags and quality provided in first round

Table B.76: Most important terms in the cluster

	1	2	3	4	5	means	probability
dance	3	2	2	3	3	2.60	0.459
dancing	3	3	3	1	3	2.60	0.459
fun	3	3	3	3	3	3.00	1.000
hiphop	3	3	3	3	3	3.00	1.000
music	3	3	3	3	1	2.60	0.459
party	1	3	3	2	3	2.40	0.188
rock	3	1	1	3	3	2.20	0.102
rock_n_roll	3	3	3	3	2	2.80	0.622
shock	3	3	3	3	3	3.00	1.000

Table B.78: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: rock party

## Cluster #27

light	sun	sky
morning	shine	rain
comes	fire	moon
dark	red	black
stars	white	blue
bright	door	wind
cold	fly	shining
burning	burn	summer
hand	darkness	star
hope	till	bring

tags	quality
weather	2
rainbow	1
weather seasons	1
weather outside	2
nature	1
hope	1

Table B.80: Tags and quality provided in first round

Table B.79: Most important terms in the cluster

	1	2	3	4	5	means	probability
hope	3	3	2	3	3	2.80	0.785
nature	3	2	1	3	1	2.00	0.084
outside	3	1	3	2	3	2.40	0.354
rainbow	3	3	3	3	3	3.00	1.000
seasons	3	3	3	3	3	3.00	1.000
weather	1	3	3	1	2	2.00	0.084

Table B.81: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: nature weather

Tags for significance level 30%: nature weather

## Cluster #28

don	care	anymore
mean	worry	understand
forget	else	

Table B.82: Most important terms in the cluster

tags	quality
problem	1
broken_love	1
	3
understanding	in- 2
sightfully	
insecure	2
worry	1

Table B.83: Tags and quality provided in first round

	1	2	3	4	5	means	probability
broken_love	3	3	3	1	3	2.60	0.631
insecure	3	3	3	3	3	3.00	1.000
insightfully	3	3	3	3	3	3.00	1.000
problem	1	3	3	3	3	2.60	0.631
understanding	3	3	3	3	3	3.00	1.000
worry	3	3	1	3	3	2.60	0.631

Table B.84: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: *(None)*

Tags for significance level 30%: *(None)*

## Cluster #29

leave	fly	easy	tags	quality
dont	goodbye	dead	loss	2
break	waiting	cold	murder_for_hire	3
set	tried	dream	leaving	1
empty	lonely	breathe	leave	2
leavin	leaving	phone	dream	2
block	scream	smile	loss goodbye	1
past	couldn	lot		
catch	dark	free		
cant	door	niggaz		

Table B.86: Tags and quality provided in first round

Table B.85: Most important terms in the cluster

	1	2	3	4	5	means	probability
dream	3	1	3	3	3	2.60	0.631
goodbye	3	3	2	3	1	2.40	0.354
leave	3	3	1	3	2	2.40	0.354
leaving	1	3	3	1	3	2.20	0.214
loss	3	3	3	2	3	2.80	0.785
murder_for_hire	3	3	3	3	3	3.00	1.000

Table B.87: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: leaving

## Cluster #30

hold	hand	tight	tags	quality
arms	close	touch	love	2
kiss	strong	forever	believe	3
break	till	breath	love	1
help	hands	hope	together love feelings	1
told	feeling	feels	relations	
wait	near	onto	romance	1
fear	soon	understand	romance future	1
gon	holding	watch		
stand	lips	faith		

Table B.89: Tags and quality provided in first round

Table B.88: Most important terms in the cluster

	1	2	3	4	5	means	probability
believe	3	3	3	3	3	3.00	1.000
feelings	3	3	3	3	3	3.00	1.000
future	3	3	3	3	3	3.00	1.000
love	1	1	2	3	1	1.60	0.005
relations	2	3	3	1	3	2.40	0.281
romance	3	3	1	2	2	2.20	0.162
together	3	3	3	3	3	3.00	1.000

Table B.90: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: love

Tags for significance level 30%: love romance relations

## Cluster #31

name	didn	told	tags	quality
mean	game	heard	3	3
remember	wasn	couldn	sport challenge	3
pain	care	rain		3
seen	started	called	sorrow	1
tried	rock	walked	relation	2
wouldn	blame	talk		3
school	king	shame		
calling	word	lord		
girls	fame	break		

Table B.92: Tags and quality provided in first round

Table B.91: Most important terms in the cluster

	1	2	3	4	5	means	probability
3	3	3	3	3	3	3.00	1.000
challenge	3	3	3	3	3	3.00	1.000
relation	1	3	3	2	3	2.40	0.457
sorrow	3	3	3	1	3	2.60	0.715
sport	3	1	3	3	3	2.60	0.715

Table B.93: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: (*None*)

## Cluster #32

stay	forever	change	tags	quality
reason	dont	nigga	Hiphop love	2
lay	alive	fly	hip-hop	1
loving	game	whatever		3
strong	gon	flow	hiphop	3
niggaz	morning	wit	love	1
found	goodbye	town	gangsta	2
break	remember	awhile		
bed	listen	sky		
block	lose	pray		

Table B.95: Tags and quality provided in first round

Table B.94: Most important terms in the cluster

	1	2	3	4	5	means	probability
gangsta	1	3	1	3	1	1.80	0.121
hip-hop	3	1	3	3	2	2.40	0.603
hiphop	3	3	3	1	3	2.60	0.816
love	3	3	3	2	3	2.80	0.921

Table B.96: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: gangsta

## Cluster #33

heart	apart	start	tags	quality
break	soul	broken	lovesickness	1
arms	pain	lonely	illness	2
forever	dark	breaking	broken_hearted	2
words	broke	beating	love_pain love tears	1
deep	tear	tears	think_&_feel	2
feeling	loved	beat	love loss	1
falling	chance	lord		
care	hurt	fool		
close	loving	couldn		

Table B.98: Tags and quality provided in first round

Table B.97: Most important terms in the cluster

	1	2	3	4	5	means	probability
broken_hearted	3	1	1	2	1	1.60	0.003
illness	3	3	3	3	3	3.00	1.000
loss	3	2	3	3	2	2.60	0.506
love	1	3	3	3	3	2.60	0.506
love_pain	3	3	3	1	3	2.60	0.506
lovesickness	2	3	2	3	3	2.60	0.506
tears	3	3	3	3	3	3.00	1.000
think_&_feel	3	3	3	3	3	3.00	1.000

Table B.99: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: broken\_hearted

Tags for significance level 30%: broken\_hearted

## Cluster #34

ll	forever	hand
wait	someday	soon
promise		

Table B.100: Most important terms in the cluster

tags	quality
	3
faith	2
	3
promise future	1
future	2
love future	2

Table B.101: Tags and quality provided in first round

	1	2	3	4	5	means	probability
faith	3	1	3	3	1	2.20	0.423
future	1	3	1	3	3	2.20	0.423
love	3	2	3	3	3	2.80	0.921
promise	3	3	2	1	2	2.20	0.423

Table B.102: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: *(None)*

Tags for significance level 30%: *(None)*

## Cluster #35

life	living	pain
rest	livin	death
change	wife	spend
days	game	soul
dream	bring	hope
found	goes	worth
alive	thug	thank
road	dreams	moment

Table B.103: Most important terms in the cluster

tags	quality
life	3
killling	2
living_and_dying life	1
prison	2
classic	2
life	2

Table B.104: Tags and quality provided in first round

	1	2	3	4	5	means	probability
classic	3	3	3	3	3	3.00	1.000
killling	3	3	3	3	3	3.00	1.000
life	3	1	1	1	3	1.80	0.064
living_and_dying	1	2	3	3	1	2.00	0.134
prison	3	3	3	2	3	2.80	0.852

Table B.105: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: life

Tags for significance level 30%: life living\_and\_dying

## Cluster #36

mind	change	lose	tags	quality
free	soul	crazy	feelings	3
body	peace	line	dreaming	2
blind	mean	losing	feelings	2
blow	dark	thinking		3
care	control	ease	literature	2
till	shine	times	mind	2
brain	drive	dreams		
days	relax	road		
changed	sweet	tried		

Table B.107: Tags and quality provided in first round

Table B.106: Most important terms in the cluster

	1	2	3	4	5	means	probability
dreaming	3	3	3	3	1	2.60	0.816
feelings	1	1	2	3	3	2.00	0.231
literature	3	2	3	3	3	2.80	0.921
mind	2	3	1	1	3	2.00	0.231

Table B.108: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: feelings mind

## Cluster #37

feel	makes	feeling	tags	quality
touch	sometimes	pain	feelings	1
alright	feels	alive	red-hot	1
body	words	anymore	feelings	1
feelin	control	bad	feelings	2
coming	fire	moment	skin	2
hide			emotion	1

Table B.109: Most important terms in the cluster

Table B.110: Tags and quality provided in first round

	1	2	3	4	5	means	probability
emotion	1	1	3	2	1	1.60	0.045
feelings	2	3	1	1	2	1.80	0.121
red-hot	3	3	3	3	3	3.00	1.000
skin	3	2	3	3	3	2.80	0.921

Table B.111: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: emotion

Tags for significance level 30%: emotion feelings

## Cluster #38

world	change	round	tags	quality
seen	dream	save	universal_peace	1
girls	found	cold	sex	2
living	wide	black		3
bring	looking	peace	future change world	2
comes	stand	sound	big thing	1
build	fly	children	world	2
top	stars	watching		
city	tomorrow	sometimes		
earth	rock	crazy		

Table B.113: Tags and quality provided in first round

Table B.112: Most important terms in the cluster

	1	2	3	4	5	means	probability
big	3	3	3	3	3	3.00	1.000
change	3	2	3	3	1	2.40	0.281
future	3	3	3	3	2	2.80	0.724
sex	3	3	3	3	3	3.00	1.000
thing	3	3	3	3	3	3.00	1.000
universal_peace	1	3	3	3	3	2.60	0.562
world	3	1	1	1	3	1.80	0.024

Table B.114: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: world

Tags for significance level 30%: world change

## Cluster #39

gotta	move	gon	tags	quality
body	bad	damn	hipHop	3
hit	aint	mean	dancing	3
bout	ladies	boy	party	3
lookin	comes	party	party sex dance	1
slow	trying	shake	hope	2
club	strong	help	party	2
money	pop	ride		
hands	yea	sometimes		
listen	goin	top		

Table B.116: Tags and quality provided in first round

Table B.115: Most important terms in the cluster

	1	2	3	4	5	means	probability
dance	1	1	1	3	3	1.80	0.038
dancing	3	3	3	1	3	2.60	0.631
hiphop	3	3	3	3	3	3.00	1.000
hope	3	3	3	3	3	3.00	1.000
party	3	3	2	2	1	2.20	0.214
sex	3	2	3	3	3	2.80	0.785

Table B.117: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: dance

Tags for significance level 30%: dance party

## Cluster #40

live	forever	living	tags	quality
free	fly	rock	life	2
control	dance	lord	freedom	2
money	fight	dead	dream	2
moment	learn	soul	relax freedom	1
hands	peace	roll	poetry	2
city	everyday	break	hope	1
rest	start	reason		
alive	dreams	dream		
pay	hope	wouldn		

Table B.119: Tags and quality provided in first round

Table B.118: Most important terms in the cluster

	1	2	3	4	5	means	probability
dream	1	2	3	3	1	2.00	0.084
freedom	3	3	1	3	3	2.60	0.631
hope	2	3	2	3	3	2.60	0.631
life	3	3	3	3	2	2.80	0.785
poetry	3	1	3	3	3	2.60	0.631
relax	3	3	3	3	3	3.00	1.000

Table B.120: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: dream

Tags for significance level 30%: dream

## Cluster #41

run	boy	gun	tags	quality
hide	sun	hit	hipHop gangster	1
bad	fun	running	killling	2
town	free	fast	running_away	2
ride	ground	block	gang crime street	1
shot	cover	girls	underdog	1
watch	friend	land	run	2
soul	comin	fire		
scared	red	ran		
runnin	ready	break		

Table B.122: Tags and quality provided in first round

Table B.121: Most important terms in the cluster

	1	2	3	4	5	means	probability
crime	3	3	1	3	1	2.20	0.102
gang	3	3	3	3	2	2.80	0.622
gangster	3	3	3	3	3	3.00	1.000
hiphop	3	3	3	3	3	3.00	1.000
killling	3	3	3	3	3	3.00	1.000
run	3	1	3	1	3	2.20	0.102
running_away	1	3	3	3	3	2.60	0.459
street	3	2	3	2	3	2.60	0.459
underdog	3	3	3	3	3	3.00	1.000

Table B.123: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: crime run

## Cluster #42

won	forget	stand	tags	quality
change	hope	anymore	god	3
till	free	hurt	challenge	2
tomorrow	understand	help		3
wait	pain	til		3
waiting	lord	shine	fear	1
lose	door		future hope	1

Table B.124: Most important terms in the cluster

Table B.125: Tags and quality provided in first round

	1	2	3	4	5	means	probability
challenge	1	1	3	3	3	2.20	0.293
fear	3	3	3	3	3	3.00	1.000
future	3	2	2	1	1	1.80	0.064
god	3	3	3	3	3	3.00	1.000
hope	3	3	1	2	2	2.20	0.293

Table B.126: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: future

Tags for significance level 30%: future challenge hope

## Cluster #43

gonna	alright	break	tags	quality
change	rock	friends	rock	3
till	wait	fun	heartbreaking	2
lose	music	door		3
town	chance	days		3
someday	woman	told	happy	2
aint	shine	happen	future	1
help	shit	party		
comin	floor			

Table B.128: Tags and quality provided in first round

Table B.127: Most important terms in the cluster

	1	2	3	4	5	means	probability
future	3	3	2	1	1	2.00	0.231
happy	1	1	3	3	3	2.20	0.423
heartbreaking	3	3	3	2	3	2.80	0.921
rock	3	3	1	3	3	2.60	0.816

Table B.129: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: *(None)*

Tags for significance level 30%: future

## Cluster #44

tonight	alright	body	tags	quality
party	tight	lights	party	3
ready	dance	fly	disco	1
ride	sleep	feeling	party dance clubbing	1
girls	air	hot	dance party sex	1
club	floor	bright	feel_good	1
rock	close	tomorrow	party dance	1
wake	town	lookin		
arms	feelin	waiting		
sight	alive	sexy		

Table B.131: Tags and quality provided in first round

Table B.130: Most important terms in the cluster

	1	2	3	4	5	means	probability
clubbing	3	1	3	2	1	2.00	0.084
dance	3	3	2	3	2	2.60	0.631
disco	3	3	3	3	3	3.00	1.000
feel_good	3	3	3	3	3	3.00	1.000
party	1	3	1	3	3	2.20	0.214
sex	3	2	3	1	3	2.40	0.354

Table B.132: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: clubbing

Tags for significance level 30%: clubbing party

## Cluster #45

left	floor	forget	tags	quality
told	hit	feeling	love	2
goodbye	didn	cold	depression	1
hands	broken	empty	past_memories	2
tears	hand	lose	alone loneliness	1
tired	loved	care	legend	1
looking	dance	died	loss	1
eye	hurt	memories		
hell	goin	memory		
damn	wasn	air		

Table B.134: Tags and quality provided in first round

Table B.133: Most important terms in the cluster

	1	2	3	4	5	means	probability
alone	3	3	3	3	3	3.00	1.000
depression	1	3	1	2	3	2.00	0.056
legend	3	3	3	3	3	3.00	1.000
loneliness	3	1	3	3	1	2.20	0.162
loss	3	2	3	3	3	2.80	0.724
love	3	3	2	3	3	2.80	0.724
past_memories	2	3	3	1	2	2.20	0.162

Table B.135: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: depression

Tags for significance level 30%: depression loneliness past\_memories

## Cluster #46

lost	found	soul	tags	quality
hope	control	pain	loss	2
dead	cold	words	killing	1
feels	miss	ground		3
broken	heaven	lonely		3
save	running	lose	past	2
cost	memory	change	loss	1
sad	free	feeling		
loved	dreams	truth		
arms	trying	caught		

Table B.137: Tags and quality provided in first round

Table B.136: Most important terms in the cluster

	1	2	3	4	5	means	probability
killing	3	3	3	3	3	3.00	1.000
loss	1	3	1	1	1	1.40	0.052
past	3	1	2	3	2	2.20	0.640

Table B.138: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: loss

Tags for significance level 30%: loss

## Cluster #47

girl	girls	boy	tags	quality
body	damn	club	hipHop woman party	2
ass	shake	hair	hip-hop	1
guy	wit	bad	boys_and_girls	1
hook	pretty	lookin	party relations dance	1
dance	bout	hit	love	2
floor	ladies	lovin	party dance sex	1
repeat	yea	freak		
sexy	lady	gon		
shorty	party	kinda		

Table B.140: Tags and quality provided in first round

Table B.139: Most important terms in the cluster

	1	2	3	4	5	means	probability
boys_and_girls	1	3	3	3	1	2.20	0.102
dance	3	1	1	3	2	2.00	0.028
hip-hop	3	3	3	3	3	3.00	1.000
hiphop	3	3	3	3	3	3.00	1.000
love	3	3	3	3	3	3.00	1.000
party	3	2	2	1	3	2.20	0.102
relations	3	3	3	2	3	2.80	0.622
sex	3	3	3	3	3	3.00	1.000
woman	3	3	3	3	3	3.00	1.000

Table B.141: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: dance

Tags for significance level 30%: dance boys\_and\_girls party

## Cluster #48

real	money	rap	tags	quality
deal	word	niggas	HipHop	1
game	top	hood	street life	1
check	dreams	mean	dreaming_gangster_rapper	1
hope	rock	girls	hiphop	3
told	straight	slow		2
dog	talk	friends	gangsta money	1
ride	fake	lot		
aint	feels	king		
gon	bridge	lady		

Table B.143: Tags and quality provided in first round

Table B.142: Most important terms in the cluster

	1	2	3	4	5	means	probability
dreaming_gangster_rapper	3	3	3	3	1	2.60	0.631
gangsta	1	1	1	3	3	1.80	0.038
hiphop	3	2	3	3	2	2.60	0.631
life	3	3	3	1	3	2.60	0.631
money	3	3	2	3	3	2.80	0.785
street	3	3	3	2	3	2.80	0.785

Table B.144: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: gangsta

Tags for significance level 30%: gangsta

## Cluster #49

love	sweet	forever
touch	repeat	kiss
found	lover	woman
else	darling	loving
words	loves	

Table B.145: Most important terms in the cluster

tags	quality
love	1
love	1
love	1
love forever	1
slime	2
love	1

Table B.146: Tags and quality provided in first round

	1	2	3	4	5	means	probability
forever	3	3	2	2	3	2.60	0.925
love	1	1	1	1	1	1.00	0.003
slime	3	3	3	3	3	3.00	1.000

Table B.147: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: love

Tags for significance level 30%: love

## Cluster #50

people	rock	black	tags	quality
music	care	round	hipHop	3
money	start	party	clubbing	1
dance	shit	lot	party dance	1
town	words	brother		3
beat	hand	house	on_stage	1
listen	matter	sound	party music	2
bring	bout	word		
brothers	hands	land		
bad	hip	hop		

Table B.149: Tags and quality provided in first round

Table B.148: Most important terms in the cluster

	1	2	3	4	5	means	probability
clubbing	3	3	3	3	1	2.60	0.631
dance	1	1	3	3	2	2.00	0.084
hiphop	3	3	3	2	3	2.80	0.785
music	3	3	2	1	3	2.40	0.354
on_stage	3	3	3	3	3	3.00	1.000
party	2	2	1	3	3	2.20	0.214

Table B.150: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: dance

Tags for significance level 30%: dance party

## Cluster #51

call	phone	boy	tags	quality
name	house	wait	hipHop woman	2
roll	town	bitch	night_with_hoes	1
bout	talk	lil	boys_and_girls	1
hook	money	miss	relations talk	2
boys	six	soul	communication	2
five	crazy	repeat		3
feelin	tha	morning		
damn	hell	ladies		
ass	hands	friends		

Table B.152: Tags and quality provided in first round

Table B.151: Most important terms in the cluster

	1	2	3	4	5	means	probability
boys_and_girls	3	1	3	2	1	2.00	0.056
communication	3	3	3	3	3	3.00	1.000
hiphop	3	3	3	3	2	2.80	0.724
night_with_hoes	1	3	3	3	3	2.60	0.562
relations	3	3	2	1	3	2.40	0.281
talk	3	2	1	3	3	2.40	0.281
woman	3	3	3	3	3	3.00	1.000

Table B.153: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: boys\_and\_girls

Tags for significance level 30%: boys\_and\_girls relations talk

## Cluster #52

re	coming	feeling	tags	quality
looking	else			3
			love	3
				3
				3
			senses	3
				3

Table B.154: Most important terms in the cluster

Table B.155: Tags and quality provided in first round

	1	2	3	4	5	means	probability
love	1	3	3	3	3	2.60	0.997
senses	3	1	3	3	3	2.60	0.997

Table B.156: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: (*None*)

## Cluster #53

eyes	close	blue	tags	quality
sky	tears	smile		3
arms	lies	dreams	killing	1
lips	looked	seen	sky melancholy	1
skies	moment	touch	love deep sad	2
blood	hair	stars	impressions	1
surprise	kiss	dry	emotion	2
sleep	watch	hand		
deep	looking	soul		
dream	alive	realize		

Table B.158: Tags and quality provided in first round

Table B.157: Most important terms in the cluster

	1	2	3	4	5	means	probability
deep	3	3	3	3	3	3.00	1.000
emotion	1	3	1	1	1	1.40	0.001
impressions	3	1	3	3	3	2.60	0.506
killing	3	3	3	3	3	3.00	1.000
love	3	3	2	3	3	2.80	0.670
melancholy	3	2	3	3	3	2.80	0.670
sad	3	3	3	2	3	2.80	0.670
sky	3	3	3	3	3	3.00	1.000

Table B.159: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: emotion

Tags for significance level 30%: emotion

## Cluster #54

head	dead	bed	tags	quality
lay	hands	hit	dance	2
feet	dance	remember	killling	2
cut	body	move		3
floor	shot	hand	crime dead street gang	1
words	ground	shake	dance	2
door	blow	mean	crime	1
red	fuckin	hole		
feeling	looking	dat		
bad	bout	heard		

Table B.161: Tags and quality provided in first round

Table B.160: Most important terms in the cluster

	1	2	3	4	5	means	probability
crime	3	3	1	1	1	1.80	0.038
dance	3	3	3	3	3	3.00	1.000
dead	3	2	3	3	3	2.80	0.785
gang	2	3	3	3	3	2.80	0.785
killling	1	3	2	3	3	2.40	0.354
street	3	1	3	2	3	2.40	0.354

Table B.162: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: crime

Tags for significance level 30%: crime

## Cluster #55

tell	friends	told	tags	quality
truth	lie	dont	love	3
friend	hell	bout	friendship	1
lies	trying	kiss	talking	1
talking	reason	hope	friends talk cheat	2
talk	check	aint	friendship	1
drop	thinking		trust	1

Table B.163: Most important terms in the cluster

Table B.164: Tags and quality provided in first round

	1	2	3	4	5	means	probability
cheat	3	1	3	3	3	2.60	0.562
friends	1	3	3	3	2	2.40	0.281
friendship	3	3	2	3	3	2.80	0.724
love	3	3	3	3	3	3.00	1.000
talk	3	2	3	1	1	2.00	0.056
talking	3	3	3	3	3	3.00	1.000
trust	3	3	1	2	3	2.40	0.281

Table B.165: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: talk

Tags for significance level 30%: talk friends trust

## Cluster #56

home	coming	phone	tags	quality
road	house	sweet		3
days	boy	city	city_life	1
goin	daddy	cold	family home	1
door	soon	mama	family home home-	1
land	sleep	hell	coming return	
money	streets	bed	allday	2
wish	ride	told	home	1
wonder	aint	family		
street	arms	comin		

Table B.167: Tags and quality provided in first round

Table B.166: Most important terms in the cluster

	1	2	3	4	5	means	probability
allday	3	3	3	3	3	3.00	1.000
city_life	3	3	3	3	3	3.00	1.000
family	3	3	2	1	3	2.40	0.354
home	3	1	1	3	1	1.80	0.038
homecoming	1	3	3	2	3	2.40	0.354
return	3	2	3	3	3	2.80	0.785

Table B.168: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: home

Tags for significance level 30%: home

## Cluster #57

day	sun	rain	tags	quality
wait	days	til	weather	2
morning	tomorrow	change	seasons	2
comes	pay	wish	weather	1
getting	remember	sunshine	weather future change	2
town	blue	minute	nature	2
wake	till	cold	hope	2
words				

Table B.169: Most important terms in the cluster

Table B.170: Tags and quality provided in first round

	1	2	3	4	5	means	probability
change	3	1	3	3	3	2.60	0.631
future	3	3	2	2	3	2.60	0.631
hope	3	3	1	3	1	2.20	0.214
nature	3	3	3	3	2	2.80	0.785
seasons	3	2	3	3	3	2.80	0.785
weather	1	3	3	1	3	2.20	0.214

Table B.171: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: (*None*)

Tags for significance level 30%: hope weather

## Cluster #58

little	bit	sweet	tags	quality
daddy	boy	goes	woman	2
child	blue	mama	love	2
lot	house	town	family love	1
pretty	white	start	relations romance	2
woman	mouth	closer		2
girls	drive	crazy	childhood love	1
kiss	piece	hand		
makes	middle	black		
soul	honey	told		

Table B.173: Tags and quality provided in first round

Table B.172: Most important terms in the cluster

	1	2	3	4	5	means	probability
childhood	2	3	1	1	3	2.00	0.084
family	1	1	2	3	1	1.60	0.010
love	3	2	3	3	3	2.80	0.785
relations	3	3	3	3	2	2.80	0.785
romance	3	3	3	2	3	2.80	0.785
woman	3	3	3	3	3	3.00	1.000

Table B.174: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: family childhood

Tags for significance level 30%: family childhood

## Cluster #59

true	dream	dreams	tags	quality
forever	remember	hand	love	3
wish	blue	kiss	rainy_winter_night	3
promise	sky	lie	dreams	2
moment	lonely	days	love dream	2
dreaming	sleep	wait	trash	2
loving	heaven	rain	love future	1
wake	stars	couldn		
hit	smile	miss		
waiting	star	deep		

Table B.176: Tags and quality provided in first round

Table B.175: Most important terms in the cluster

	1	2	3	4	5	means	probability
dream	1	3	1	3	1	1.80	0.038
dreams	3	3	3	1	3	2.60	0.631
future	3	1	2	3	3	2.40	0.354
love	3	3	3	2	3	2.80	0.785
rainy_winter_night	3	3	3	3	3	3.00	1.000
trash	3	3	3	3	3	3.00	1.000

Table B.177: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: dream

Tags for significance level 30%: dream

## Cluster #60

try	change	lie	tags	quality
understand	trying	fight	problem	3
matter	tried	start	brawl	2
help	tryin	else		3
forget	hide	told	maybe fear	2
hurt	comes	mean	feelings	1
lies	step	wake		3
buck	easy	watch		
afraid	chance	living		
buy	smile	listen		

Table B.179: Tags and quality provided in first round

Table B.178: Most important terms in the cluster

	1	2	3	4	5	means	probability
brawl	3	3	3	3	3	3.00	1.000
fear	3	1	3	3	3	2.60	0.715
feelings	3	2	3	3	3	2.80	0.852
maybe	3	3	1	2	3	2.40	0.457
problem	1	3	3	1	3	2.20	0.293

Table B.180: Tag scores assigned in the second round, average score and prior probability of attaining the average score.

Tags for significance level 10%: *(None)*

Tags for significance level 30%: problem