Context based multimedia information retrieval

Mølgaard, Lasse Lohilahti

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Mølgaard, L. L. (2009). *Context based multimedia information retrieval*. Technical University of Denmark. IMM-PHD-2009-218

Context based multimedia information retrieval

Lasse Lohilahti Mølgaard

Kongens Lyngby 2009
IMM-PHD-2009-218

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Summary

The large amounts of digital media becoming available require that new approaches are developed for retrieving, navigating and recommending the data to users in a way that reflects how we semantically perceive the content. The thesis investigates ways to retrieve and present content for users with the help of contextual knowledge.

Our approach to model the context of multimedia is based on unsupervised methods to automatically extract meaning. We investigate two paths of context modelling. The first part extracts context from the primary media, in this case broadcast news speech, by extracting topics from a large collection of the transcribed speech to improve retrieval of spoken documents.

The context modelling is done using a variant of probabilistic latent semantic analysis (PLSA), to extract properties of the textual sources that reflect how humans perceive context. We perform PLSA through an approximation based on non-negative matrix factorisation NMF.

The second part of the work tries to infer the contextual meaning of music based on extra-musical knowledge, in our case gathered from Wikipedia. The semantic relations between artists are inferred using linking structure of Wikipedia , as well as text-based semantic similarity.

The final aspect investigated is how to include some of the structured data available in Wikipedia to include temporal information. We show that a multiway extension of PLSA makes it possible to extract temporally meaningful topics, better than using a stepwise PLSA approach to topic extraction.

Resumé

De store mængder af digitale medier, der er tilgængelige kræver, at der udvikles nye metoder til at hente, navigere og anbefale disse data til brugere på en måde, der reflekterer hvordan vi forstår indholdet af data. Denne rapport undersøger tilgange til at hente og præsentere indhold for brugere vha. baggrundsviden om data.

Vi baserer vores modeller af baggrundsviden i forbindelse med multimedie-data på ikke-superviserede metoder, som kan udtrække meningen af data automatisk. Vi undersøger to måder at modellere kontekst. Den første del af afhandlingen beskriver, hvorledes vi kan udtrække mening fra det primære medie, som i dette tilfælde er nyhedsudsendelser ved at finde emner fra en stor samling af transskriberet tekst. Vi viser, at denne metode forbedrer søgning i taleoptagelser.

Kontekst-modelleringen udføres vha. af en variant af probabilistisk latent semantisk analyse (PLSA), som udtrækker meningen fra tekst, der minder om den måde, som mennesker forstår tekst. Vi undersøger metoder til at implementere PLSA effektivt vha. en approksimation baseret på non-negativ matrix faktorisering (NMF).

Anden del af afhandlingen beskæftiger sig med at beskrive den kulturelle baggrund, der omgærder musik. Denne viden må udtrækkes fra data, der er udenfor musikken, i vores tilfælde fra Wikipedia. Vi forsøger at finde semantiske forbindelser mellem kunstnere baseret på link-strukturen i Wikipedia , og gennem semantisk modellering af teksten.

Til sidst undersøger vi hvorledes man kan inkludere noget af det strukturerede

data, som Wikipedia også indeholder. Dette giver mulighed for at finde emner, der indeholder en temporal dimension. Vi viser at en multivejs-variant af PLSA gør det muligt at finde temporalt meningsfulde emner, og at denne globale analyse af data giver bedre resultater end en trinvis modellering af emner.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfillment of the requirements for acquiring the Ph.D. degree.

The Ph.D.-project was financed by the Danish Technical Research Council through the 'Intelligent Sound' project.

The thesis consists of a summary report and two research papers written during the period 2006–2009, and elsewhere published.

Lyngby, 2009

Lasse Lohilahti Mølgaard

Papers included in the thesis

B Lasse Lohilahti Mølgaard, Kasper Winther Jørgensen, Lars Kai Hansen Castsearch - Context Based Spoken Document Retrieval *Int. Conf. Acoustics Speech and Signal Processing* Published in proceedings.

C Lasse Lohilahti Mølgaard, Jan Larsen, Cyril Goutte Temporal analysis of text data using latent variable models *Int. Workshop on Machine Learning for Signal Processing* Published in proceedings.

Acknowledgements

I want to thank my supervisors Jan Larsen and Lars Kai Hansen for our working relationship during the Ph.D. work. Special thanks goes to Ulla for managing all the practical matters during the course of the Ph.D.-study.

I am also very grateful to Cyril Goutte and the Interactive Language Technology group at the National Research Council Canada for hosting my external stay. I very much enjoyed the friendly atmosphere and the good discussions we had.

Abbreviations

AIC	Akaike's information criterion
ASR	Automatic speech recognition
GD-CLS	Gradient descent - constrained least squares algorithm
HITS	Kleinberg's Hypertext-induced topic search
KL	Kullback-Leibler (divergence)
LDA	Latent Dirichlet allocation
LSA	Latent semantic analysis
mAP	mean Average Precision
MFCC	Mel frequency cepstral coefficients
MIR	Music information retrieval
mwPLSA	Multi-way probabilistic latent semantic analysis
NLP	Natural language processing
NMF	Non-negative matrix factorisation
PARAFAC	Parallel factor analysis
PCA	Principal component analysis
PLSA	Probabilistic latent semantic analysis
POS	Part of speech

SVD	Singular value decomposition
TF-IDF	Term frequency - inverse document frequency
ICA	Independent Component Analysis
ML	Maximum likelihood
VQ	Vector Quantisation

Contents

Chapter 1

Introduction

The huge amounts of and the ever continuing growth of data available for users in the Internet era is breathtaking, and changes the way we perceive the world. The massive amounts of data has for instance changed the way natural language processing has been approached by researchers, as described by Halevy et al. [45]. Huge corpora of text earlier consisted of 100 million words, but the indexing efforts by Google has made collections of 10^{12} word corpora with frequency counts of all word sequences up to 5 words available. The data obtained from the Internet[1] is flawed at times, containing incomplete sentences spelling errors, grammatical errors and other errors, but the fact that it contains massive amounts of observations makes it possible to capture a range of aspects of human behaviour, and understanding. Other kinds of data in the Web can also be used describe human behaviour - including sound, videos, images, links, tables and user interactions, we just need to access them.

The availability of multimedia data, has also made audio and video a more integrated part of our everyday life. This is mainly due to consumer electronics such as memory and hard discs becoming cheaper, which has made our personal computers into dedicated media players. Now that everyone has a portable digital media player in the pocket, in the form of an Apple 'iPod' or any average cell phone, that puts several gigabytes of multimedia content in peoples pockets,

[1] We will use the designations Internet, World Wide Web, and Web interchangeably, for the numerous different sources of data present in the distributed database formed by the Internet.

has made sound and video an always available medium. The technological advances have also changed the way music is distributed as it has moved from the physical media over digital distribution of files, like Apple's itunes store[2], to on-demand music delivery through streaming services such as Pandora[3] and Spotify[4]. The delivery of other multimedia data such as speech and video have also become an on-demand service, for instance through streaming services, e.g., the ubiquitous presence of Youtube[5] on the Web, as well as podcasting and audio books.

The services such as Youtube and blogs rely on user generated content, and as it has become possible for regular users to produce material, at near-professional level, has meant that media consumption has moved from using traditional sources of media, such as record companies, tv stations and news papers towards user generated content. Making everything available on he Internet has also introduced new business models, described by the "the long tail" introduced by Anderson [9], which proposes that the future is in selling fewer copies of each item, but leveraging the sales in the niches of the market, in contrast to earlier, when the sales were driven by a few hits. The main problem with this business model is how to help consumers find the rare items that they are interested in, among the seemingly endless streams of data available.

This is still an open question for non-textual data such as audio and video that still lack systems that fully answer the need for organising and navigating these multimedia information. To date the principal approach for indexing and searching digital media has been via the metadata stored inside each media file.

Metadata has generally been manually created consisting of short text fields, which in the field of music contains information about the composer, performer, album, artist, title, and in some cases, more subjective aspects of music such as genre or mood. The creation of metadata, however, is labor-intensive and therefore not always available. Secondly, we often see a lack of consistency in metadata, which can render media files difficult or impossible to retrieve. Especially in the old regime of exact search, usually employed in music stores.

The need for metadata has lead to a large body of research in automatically generating these descriptions. The efforts within image retrieval for instance includes work ranging from performing shape recognition, to face recognition. Within the domain of sound we can identify a number of aspects to focus on depending on the kind of sounds contained in the recordings.

[2] http://www.apple.com/itunes/

[3] http://www.pandora.com

[4] http://www.spotify.com

[5] http://www.youtube.com

Given a sound recording, there seems to be a common agreement on classifying sound into three different classes, with their own distinct characteristics. The three classes are speech, music, and environmental sounds.

Speech needs to be described by finding the words and which language is spoken, as well as which person is speaking on the recording. The recording conditions may also be a useful feature. These efforts have until now been the most researched, producing very efficient speech recognisers and speaker recognition systems.

Music can be described in many different ways, including both objective and subjective descriptions. Typical objective measures are instrumentation, artist, whether the song has vocal or not, etc.

Environmental sounds are recordings without speech or music, such as jet engine, a bird, or street noise. These must be described with terms related to the kind of phenomenon producing the sounds, or other descriptive terms, such as the pleasantness or loudness of the sounds.

High level features such as the mood of a song, music genre and theme, have also been items of interest for the research community. The main problem of these kinds of data is the degree of subjectivity that is associated with the descriptions. The way a random person assigns music to genres, will be heavily based on the level of musical training the person has received, familiarity with the genre, and the cultural background. This aspect of genre classification was discussed by Aucouturier and Pachet [10], who shows that the automatic prediction of genre based on audio features hits a so-called 'glass ceiling', which means that the performance levels out at around 65-70% precision in their experiments. The genre is thus not a feature that can be predicted reliably, and likely other semantically complex features have the same inherent difficulties.

Another interesting aspect of the metadata-based retrieval of multimedia files is how to search for the media. The use of metadata for retrieval of for instance books has shown its use for librarians for the last thousand years, and the approach will work well for known item retrieval, in a setting, where the user knows the specific item he or she wants to be retrieved. However, in the age of Internet-size databases people do not necessarily know what they could find among all the data available. The ontology or directory based methods for search in the Internet have also shown to be insufficient as they do not capture all the diverse sorts of search people perform.

As a consequence music information retrieval research has in recent years moved from generating genre and other specific labels towards extracting more natural language-like descriptions of music, trying to define similarity, and the problem of creating playlists.

1.1 Context of multimedia data

The above observations on the limitations of metadata-based subjective descriptions of data leads us to discuss the use of context in the description and ultimately retrieval of multimedia data. The focus of the work described in this thesis is on audio, both speech and music.

The general goal of the work here has been to put *meaning* into the retrieval of sound. It is clear that our experience of music is influenced by a number filters that include cultural conventions, buzz and trends, and other marketing peculiarities. So every time music is received by a person it will be decoded individually based on the past history the person has with similar music, linking it with specific emotions and experiences. The song "You'll never walk alone", will for instance evoke strong connotations to Liverpool FC for any football fan, as it is the anthem of the club. The song has won so much notoriety as a football anthem that it is now adopted by numerous clubs around the world. These kinds of relations simply cannot be extracted solely from the audio signal.

Likewise speech analysis would benefit from this kind of contextual knowledge. Retrieval of speech recordings may also be aided by contextual knowledge, for instance if we are interested in reports from hurricane-catastrophes, it will be nice to have contextual knowledge to help navigation, such that the user can focus on recordings related to reports on the proportions on the catastrophe, emergency aids, or reconstruction.

What we have investigated is how to make models that understand these kinds of extra-signal information, so that the background knowledge humans use to understand sound signals can be incorporated in the retrieval and navigation of these signals. The work presented here does not directly address the relation between the audio signal and the contextual descriptions, but only considers the processing that can be done using the extra-signal sources of data.

The approach we have chosen is to use unsupervised methods to extract the relations between concepts, without resorting to genre classification or defining a closed set of descriptive terms. The methods described below rely on the unsupervised extraction of information based on the data at hand.

1.2 Contents of the report

The report contains the following chapters, describing the modelling of contextual knowledge. The report first includes 3 review chapters, setting the scene, followed by chapters describing the contributions of this project.

Chapter 2 Describes the field of multimedia retrieval setting the scene for the rest of the work presented in the thesis. It also gives a review of the field of Music Information Retrieval, describing the motivation for extracting extra-musical information from textual sources.

Chapter 3 Describes the basic techniques used for text mining, including the different pre-processing steps needed.

Chapter 4 Describes latent semantic modelling using factor models. It has been identified that human understanding of text can be modelled using latent semantics extracted from large corpora of texts. Latent semantic analysis is reviewed and further the extensions using probabilistic LSA and NMF are described. The chapter also reviews the estimation methods for these algorithms.

Chapter 5 Includes the work performed on using PLSA for retrieval of broadcast news, where the use of context helps in retrieving spoken documents despite challenges in retrieving text which has spelling errors and ambiguous document definitions. The work was also published in the paper in appendix B.

Chapter 6 Describes an analysis of the musical Wikipedia. The chapter describes the use of links and semantic relatedness to extract musical information from the Wikipedia data, and the use of these for playlist generation.

Chapter 7 Presents work on using tensor methods to integrate text data and structural knowledge such as time information for music descriptions. This work was also in part presented in the paper C.

Chapter 2

Context of multimedia data

This chapter will dwell on the concepts of assigning meaning to multimedia. Multimedia in the context of this thesis will primarily describe sound.

2.1 Meaning of sound

We will consider different types of meaning for a sound recording. The term 'meaning' is somewhat vague, but we will try to describe the different aspects of meaning that we try to model. Whitman [112] identifies three different kinds of musical meaning, that are also useful in the general case of sound; correspondence, relation, and reaction.

2.1.1 Correspondence

The first sort of meaning is the correspondence between what the original idea was by the people producing the sound recording, and what people receive from listening to the recording. In the context of music we can simply define correspondence as "what is the music about" or "how was the music made". This kind of information will often be encoded in the lyrics of a song, but the message may also be implied from the time the song was written in. For a speech signal it will simply be *what is the meaning of the spoken words.*

Another form of correspondence is the more explicit meaning of a signal, such as the structure of a composition, the instruments used, as well as timbre and notes used to play a piece of music. Speech can also be characterised in this way by identifying the speaker, recording conditions, and the words spoken.

2.1.2 Relation

Relational meaning is the connection between concepts. This sort of meaning is quite obvious in linguistics where word relationships are described through antonym, synonym, and hypernym relations. I.e. we can define a car through its synonym; automobile, hypernym; vehicle, and so forth. The relational knowledge is maybe most applicable for musical data, where musical similarity of artists and tracks has been a focus point in the Music Information Retrieval (MIR) community. Genres have been the premier way to organise music and describe relations between musical artists and tracks but the problems of using genre as the basis of musical organisation, include a number problems of subjectivity and marketing considerations. Furthermore some genres, such as *World Music* which is defined quite broadly as non-western music, or the even more inane definition ”non-anglophone” music, are quite indistinct. Finding more faceted connections between artists and tracks would provide a way to measure personal preferences does give us a way to understand the musical sphere. These kinds of relations are often seen in musical reviews where new artists are usually described as being similar to some other well-known artist.

Relational information in the context of speech amounts to linking together recordings that have the same semantic contents, i.e. describing the same events in the case of broadcast news. These kinds of linking are for instance useful for retrieval, where a user can be presented to all recordings on a given topic, described using the gist of the recordings.

2.1.3 Reaction

The last sort of meaning is reaction meaning. This form of meaning describes the reaction that the sound evokes. The reaction in conjunction with music is for instance what kind of emotions are evoked (joy, sadness) or how it reminds the listener of an earlier experience, e.g., a festival where they heard this band.

One of the more important reaction meanings that can be identified is maybe the "significance". Significance can be defined as what is meaningful to many listeners. The significance for instance depends on the way music is conveyed to a listener, given friends' opinions, popularity, buzz, and trends. It can be seen as a measure of what is well-known to the masses. The significance of some musical artist is thus completely outside the sound signal, and is quite a useful way to identify preference and taste and how these will be influence future music. The way to quantify significance of artists today is basically implemented using collaborative filtering recommendation systems, that measure usage patterns. These systems can be used to measure significance of artists as they give a global view of which artists are most played, but may not easily be useful for detecting new trends in music.

The reaction meaning of speech recordings is also meaningful, we may for instance want to measure the relevance of a news item based on the importance of the news on this particular day, i.e. if the topic covered in that recording is among the most featured on this day. Reaction in the form of emotions may also be useful to identify to find happy or sad news.

2.1.4 Features for meaning extraction

Generally music is difficult to describe as the perception differs based on their musical training, age, taste and familiarity [64]. Describing music therefore must consist of both objective features that can be extracted from the audio signal itself but also contextual information.

The signal processing and computer music communities have developed a wealth of techniques and technologies to describe audio and music content at the lowest level of representation. How these low-level descriptors and the concepts that music listeners use to relate with music collections needs to be bridged. There is still a kind of "semantic gap".

Celma and Cano [29] describes a multimedia information plane, illustrated in figure 2.1, splitting multimedia features into three levels of granularity or abstraction; low-level basic features, mid-level semantic features, and high-level human understanding. The first level includes physical features of the objects, such as the sampling rate of an audio file, as well as simple signal processing features. High-level features aim at describing concepts such as a guitar solo, or tonality information, such as the key and mode of a track. Finally, the highest level of abstraction should for instance be able to retrieve several audio files with "similar" guitar solos over the same key.

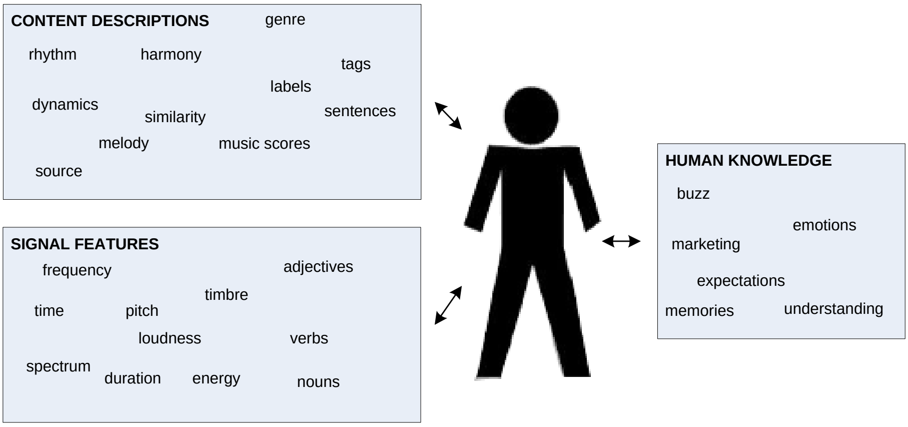


Figure 2.1: The musical information that influence the reception and understanding of a sound recording. The features may be low-level signal features, semantic features that are more useful to humans, and high-level features. The left side of features have until now been the main focus of sound retrieval. The right side on the other hand, encompasses the way a user interacts with the music content and the social network of users.

The meaning and description of speech recordings is often more intuitively produced than descriptions of music. Spoken document retrieval usually has more distinct and more easily described meaning which can be inferred using the low level features, i.e. we can form words from phonemes and transcribe the message through the spoken words. It is also possible to identify speakers and coherent segments so that it is easy to gather all utterances by a single speaker. These problems are not trivial but have been investigated thoroughly in the speech processing communities to produce very efficient features and algorithms. Spoken document retrieval will be described in chapter 5, so the rest of this chapter will mainly focus on features for music description and identify the needs for human-like understanding of music.

2.2 Acoustic music analysis

The extraction and analysis of low level acoustic measures has been extensively researched through the efforts in the Music Information Retrieval Evaluation eXchange (MIREX)[1] that has been run in conjunction with the annual Inter-

[1] http://www.music-ir.org/mirex/2009/index.php/Main_Page

national Conference on Music Information Retrieval (ISMIR).

Initial efforts concentrated on the prediction of musical genres based solely on features derived from the acoustic analysis of the music. The features derived from the acoustics include features quantifying:

Timbre This concept covers the differences in how instruments are played as well as the different methods inferred in the production process. The timbre is often described using the Mel-Frequency Cepstral Coefficients (MFCC) that were originally proposed to emulate the signal processing performed in the ear and providing a pitch independent representation of the sound.

Rhythm refers to all of the temporal aspects of a musical work, whether represented in a score, measured from a performance, or existing only in the perception of the listener [44]

Harmony The harmony of a piece of music can be defined by the combination of simultaneous notes, or chords; the arrangement of these chords along time, in progressions; and their distribution, which is closely related to the key or tonality of the piece. Chords, their progressions, and the key are relevant aspects of music perception that can be used to accurately describe and classify music content.

Structure Music structure refers to the ways music materials are presented, repeated, varied or confronted along a piece of music. Strategies for doing that are artist, genre and style-specific (i.e. the intro-verse-chorus-verse-chorus-outro of ”pop music”). Detecting the different structural sections, the most repetitive segments, or even the least repeated segments, provide powerful ways of interacting with audio content based on summaries, fast-listening and musical gist-detecting devices, and on-the-fly identification of songs.

The generation of metadata through acoustic analysis initially concentrated on prediction of genre [5, 76, 87]. The classification of genres has been investigated using a broad range of classification algorithms, using different feature selection schemes and ways to aggregate features. The most efficient audio features and feature integration techniques found in the genre classification task have subsequently been utilized for song similarity applications. The state-of-the-art today thus seems to have settled on using the MFCC representation as basic features and thereby using the timbre of music as the basic similarity.

Recently the analysis of acoustic features has moved into investigating if it is possible to generate descriptive words or so-called tags that overlap with human descriptions of the music. Such efforts were for instance done in the work by Eck et al. [36], where tags generated in a social Web service are predicted based on MFCC's using the Adaboost classifier. The work by Torres et al. [105] also investigates the correlation between audio-based features and a predefined set of descriptive tags that were chosen to describe music. These efforts thus seem to go one step closer towards more human-like understanding of music.

Even though it seems that the research within acoustic analysis of music has moved tremendously, and recently has produced very good methods for extracting correspondence between the acoustic signals and representations (textual as well as graphical) that are meaningful in describing the musical signals and it's acoustic similarity, we still need ways of retrieving extra-signal information described earlier.

2.3 Cultural background of music

The collection of musical information based on acoustic analysis has recently been augmented by the analysis of the rich cultural background that our understanding of music is embedded in. The development of this aspect of MIR has been found useful in part because genre classification performance has reached a 'glass ceiling', as mentioned in section 6.1, as well as the considerations laid out in the previous section of this chapter. Another factor is that the availability of text describing music has grown explosively in the recent years with the advent of numerous Web-based music services and blogs.

The cultural background of music can be gathered and investigated using a broad range of data sources, which has primarily been based on using the Web as a platform. The data sources include:

Collaborative filtering Analysing how people use music, e.g. by analysing which tracks are played together (also called co-play). These data can be extracted using playlists compiled by users of a collaborative playlist service or gather information directly from user activity such as implemented by last.fm.

General Web mining The Web presents a broad range of Web pages related to music. These include band pages, music magazines, expert Web pages, blogs, and numerous others. Analysis of these sorts of data can be done either using Web content mining, link analysis, as well as usage analysis. Successful use of these data relies heavily on developing focused Web crawlers for music data.

Folksonomies The recent developments on the Web has given birth to a number of services that rely heavily on average users generating descriptions of data. The user-based organisation of data using short phrases, called tags, is one very popular way to incorporate users in music description. This unstructured organisation of data is also called folksonomies, and has shown to be very useful in conjunction with multimedia retrieval.

In the following we will present these approaches, with their pros and cons.

2.3.1 Collaborative filtering

Many Web-based music services and shops employ musical recommendations. The most common approach of these recommender systems is based on analysis of earlier acts by the user. Gathering these data and mining them for relationships between items is called collaborative filtering. The most well-known and successful collaborative filtering system is the *Amazon*[2] recommender, which has shown important for retaining customers and generating sales[3]. The Amazon recommender tracks all earlier purchases by its users, building a co-occurrence matrix of users and items. Tracking which items were sold together can thus identify items that are similar.

The collaborative filtering approach has received considerable academic interest because of the *Netflix* Prize dataset, which contains more than 100 million movie ratings. The collaborative filtering approach has been applied for music in commercial systems such as last.fm [4] and mystrands[5].

Collaborative filtering methods work by building a matrix of the user preferences, which as mentioned above can be user applied ratings, purchases, or counts of the number of plays of a music track, as it is employed by e.g. last.fm. The elements of the matrix $\mathbf{R}_{i,j}$ then represents the appreciation (measured either through an explicit scoring or implicitly through a playcount in the context of music) by user u_i for the item v_j.

[2] http://www.amazon.com

[3] http://glinden.blogspot.com/2007/05/google-news-personalization-paper.html

[4] http://www.last.fm

[5] http://www.mystrands.com

$$\mathbf{R} = \begin{pmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1m} \\ \vdots & \ddots & & & \\ r_{i1} & & r_{ij} & & r_{im} \\ \vdots & & & \ddots & \vdots \\ r_{n1} & \cdots & r_{nj} & \cdots & r_{nm} \end{pmatrix}$$

A user will only have applied a score to a small number items, forming a very sparse row of the matrix, so the challenge of collaborative filtering approaches is to predict how a user will rate an unseen track given this very limited number of observations. Two approaches have emerged as the principal methods to do recommendations using collaborative filtering. One is based on modelling the neighborhood of the user, while the other approach is based on modelling the neighborhood of items. The basic methods for predicting the rating for an active user for a new item contains two steps;

- find the users most similar to the active user. This is typically done using a measure of correlation between users (in the form of rows in the matrix **R**), such as Pearson's r correlation, or cosine similarity.
- use the mean of the similar user's ratings of the new item to predict the taste of the active user.

Similarly the item-based similarity is calculated by finding items that have been rated by the same users, to find items that are worth recommending, given the original item. This similarity may be measured using either cosine similarity or correlation as in the user-based similarity. Given an active user, the system then should select the items relevant among the similar ones. This is usually done by selecting the similar items the user has rated.

Limitations of collaborative filtering

The recommender systems have until now seemed to be the most successful in helping users find interesting music, and capture one of the basic meanings namely the relational knowledge, which based on large amounts of user data could be used to get a view of a general notion relation between musical artists or tracks, and even speech recordings. The collaborative filtering approach on the other hand does have some drawbacks.

Data sparsity The high number of users and items and the relatively few ratings by each user means that the resulting user by item matrix is very sparse, containing much less than 1% non-zeros. It may thus be hard to find reliable neighbors.
The problem of finding reliable neighbors is also a problem for users with atypical tastes.

Cold-start problem As Collaborative Filtering is based on users' ratings, new users who have only a few ratings are difficult to categorise. The same problem occurs with new items, because they do not have many ratings when they are added to the collection.

Popularity bias Because social recommenders rely on user input, popular items will often be similar to lots of items. These effects were for instance investigated in [29], where social network analysis of last.fm recommendations are performed. The study shows that collaborative filtering approaches will heavily favor the most popular artists, ignoring less popular artists.

Explainability (or transparency) of the recommendations is another important element. The recommendations from collaborative filtering are usually black box systems that do not explain why two items are similar. Studies have shown that explanations of the recommendations improve the user experience making the recommenders more acceptable. [47, 103]

The folksonomies and Web mining approaches to describing music can especially address the last part of the problem in collaborative filtering but also addresses the former two to some extent.

2.3.2 Web Mining

Recommender systems and acoustic feature analysis efforts have been good approaches for quantifying relational meaning for sound, but more direct modelling of cultural metadata in the form of text would enable users to query for music using arbitrary text queries, such as "moody trumpet jazz" or "psychedelic ambient dream pop".

The vast amounts of data describing music in the World Wide Web is an obvious high-volume source of textual data. Web mining has therefore been investigated for use in conjunction with MIR in a number of works. The emergence of Web

2.0[6] on the other hand has lead to an even more enhanced focus on using Web-resources for MIR .

The first example of Web-based MIR is found in the works, [111] and [110]. They gather artist related terms, by querying a general search engine with the name of the artist. To limit the size of the page results, they add some keywords to the query, such as ”music” and ”review”. From the retrieved pages, the authors extract n-grams, perform part-of-speech-tagging and extract noun phrases. The extracted terms are then used in a TF-IDF vector space model to calculate artist similarity. Similarly, [14] improved this approach by filtering irrelevant content of the Web pages (e.g. adverts, menus, etc.). The system described also uses bag-of-features approach using nouns, adjectives and simple phrases, as well as un-tagged unigrams and bigrams as features. Extracting artist-related information from Web resources was also investigated extensively by Knees et al. [55],[96].

In [40], the authors present another approach for describing artists using Web data, based on using a search engine as a semantic similarity engine, by using the number co-occurrences between an artist and a set of labels in Web pages. The labels can be genres, instruments, or moods. The counts of co-occurrence are given by querying the Google search engine and counting the number of hits. The ideas are closely related to the work on the Google Similarity Distance in [33]. Similar work based on term co-occurrences in Web pages is presented in [99] and [56]. They define artist similarity as the conditional probability of an artist that occurs on a Web page that was returned as response to querying another artist. In Knees et al. [56], the authors focus on artist genre classification, using three different genre taxonomies. An artist assignment to a genre is considered as a special form of co-occurrence analysis. Evaluation over a small dataset shows an accuracy of over 85%.

One of the main drawbacks of Web-MIR is the polysemy of some artists' names, such as *Kiss*, *Bush*, *Porn* [98]. This problem is partially solved by the same authors, in [97]. Based on TF-IDF , they penalise the terms with high DF, that is the terms that appear in lots of documents.

Another common drawback of all the previous approaches is the high dimensionality of the datasets and the highly noisy data retrieved from more or less random Web pages. To avoid this problem, Pohle et al. [91] use Non-negative Matrix Factorisation to reduce the dimensionality of the artist-term matrix. They also use a predefined vocabulary of music terms, and analyse the content of the top-100 Web pages related to each artist. To get the most relevant pages,

[6] *”Web 2.0 refers to what is perceived as a second generation of Web development and Web design. It is characterised as facilitating communication, information sharing, interoperability, and collaboration on the World Wide Web.”* [`http://en.wikipedia.org/wiki/Web_2.0`]

they use a similar approach as [111]. The original matrix contains all the terms applied to the artists, using TF-IDF weights. This matrix is decomposed using NMF to obtain 16 topics which is used to describe the artist. After that, a music browser application allows users to navigate the collection by adjusting the weights of the derived topics, and also can recommend similar artists using cosine distance over the artists' topic loadings.

Finally, [86] compute artist and song co-occurrences from radio sources, and also from a big database of CD compilations, extracted from CDDB, to obtain artist relatedness.

Recently social network analysis has also been investigated as a source for gathering artist information. [11] investigate how the network formed by tracks in playlists produced by users in a playlist Web service can be used to relate artists to different genres, as well as determining which artists are the most relevant for a genre. [30] analyse the artist similarity networks, obtained from collaborative filtering, audio content analysis, and an expert-created network. The analysis investigates the structure as well as the effect of popular artists in these networks.

Music lyrics are also a resource that is available in the Web from the many pages that gather these data. The lyrics of a song is an obvious way of finding some of the meaning of songs directly from the source. The work on using lyrics for MIR was first investigated by Logan et al. [71], using a database of 16,000 song lyrics that were used to train latent semantic models for estimating artist similarity. The performance was clearly poorer than audio-based similarity. Determining artist similarity from lyrics is probably also stretching the semantics that are possible to extract from the lyrics alone. The lyrics on the other hand is a valuable source for identifying the emotions [88], or theme [73] of a song.

2.3.3 Social tagging

Social tags (also known as Folksonomies, or Collaborative tags) aims at annotating multimedia content using short descriptive text. Tags are freely chosen keywords, not constrained to a predefined vocabulary, and are therefore a step beyond standard metadata. The tagging approach has been very successful in image and Web page organisation services, such as *flickr*[7] and delicious[8]. Tagging is also used for sound and video retrieval in Youtube as well as last.fm. The tags have especially shown to be useful for image and text retrieval, as they are

[7] http://flickr.com

[8] http://delicious.com

often descriptive of specific elements of the main data (for instance the tag 'dog' for an image containing a dog). Tags for personal organisation of music, on the other hand is usually ad hoc and highly informal, and may say as much about the tagger as about the item being tagged. Tags may range from the obvious tags such as "rock" and "female vocalist", to more individual descriptions, for instance: "sure go ahead and depress the hell outta me what do i care" [65]. This very free form of expressing personal comprehension of music has become an active field in MIR in the recent years.

Levy and Sandler [65] perform a comprehensive analysis of tags and build latent semantic models based on the tags. They show that the vocabulary used in music tagging is much richer than in the description of images and Web pages. Suggesting that music descriptions are much more subjective than the description of for instance blog entries.

Tags have also been combined with recommender systems, Tso-Sutter et al. [106] unfold the three-way array formed by the *user* × *item* × *tag* to use standard collaborative filtering and show that the tag data improves accuracy of ratings compared to classic collaborative filtering. The three way array has also been modelled using higher-order extensions of SVD and NMF . For instance Symeonidis et al. [104] apply a higher order SVD to a music dataset (*user* × *artist* × *tags*) taken from last.fm. Their results show significant improvements in terms of the effectiveness measured through precision and recall.

Limitations of Social Tagging

One of the main limitations of social tagging is the coverage. It is quite common that only the most popular items are described by several users, creating a compact description of the item. On the other hand, less known items usually do not have enough tags to characterise them. This makes the recommendation process very difficult, specially to promote these unknown items.

Another issue is that the unconstrained vocabulary used for tags present problems with, polysemy ('I love this song' versus 'this song is about love'), synonymy ('hip-hop', 'hiphop', and 'rap'), and the usefulness of tags used for personal music organisation (e.g. 'seen live', or 'to check') for similarity among users or items. Because the tags are very short it may also be difficult to apply natural language processing techniques on them to e.g. utilize part-of-speech-tags.

Treating the collection of tags as bag-of-words features for subsequent machine learning can be used to overcome some of the problems of sparsity, synonymy and polysemy, and seems to be an interesting direction of research.

2.3.4 Summary

The forms of meaning in sound go beyond simple similarity of songs. Therefore a system for sound retrieval will need to address both the relational meaning given through similarity measures, but also answer questions regarding the correspondence meaning, i.e. the intended meaning of the sound, as well as the reaction meaning, amounting to the significance of the sound clips. The cultural background shared by people in a group or in a part of the world will therefore be an important factor to include in sound retrieval and recommendation.

The use of cultural context for music information retrieval has earlier been investigated using both unstructured data such as general Web-mined text as well as the more focused efforts in social tagging. The use of unstructured data from the Web has the main drawback, that the data is very noisy as it is hard to determine whether a Web page containing the word *Queen* is related to the band or a royal person.

The more musically focused data obtained through folksonomies provides a very good that seems to be very useful for describing musical content. The main problem is that some tags are very subjective and sometimes spurious, so a more rich description of music could be needed.

2.4 Research questions

The above observations on the need for human-like understanding of multimedia for retrieval and presentation purposes is the basis of the work described in this thesis. The primary goal of this is to investigate the use of latent semantic analysis to extract knowledge bottom-up to describe multimedia data.

Above we identified different forms of meaning of sound. In the context of speech we will examine how to use latent contexts to infer a sort of correspondence meaning, i.e. identifying the actual meaning of spoken documents using the latent semantic analysis methodology. These efforts are described in chapter 5

The relational meaning, i.e. measuring how similar musical artists or tracks are, has been the most prominent research direction in MIR , and is also the focus of the work here, which extends similarity by creating more transparent similarity measures based on contextual knowledge.

In contrast to earlier approaches for mining community metadata, that use general Web data or tags, we will investigate Wikipedia that is an appealing compromise that provides a rich collaboratively created textual description of music in contrast to the short tags, while being a controlled medium which avoids much of the noise found in Web data.

We will investigate the semantic links between artists both through the textual descriptions of artists, but also through another useful feature of the Wikipedia namely the internal linking structure, which can also supply us with relational knowledge.

Chapter 3

Preprocessing for text mining

This chapter describes the basic preprocessing steps in text mining that are necessary before any statistical language models can be applied to the data. The preprocessing steps and choice of mathematical representation of the textual data are of paramount importance for the success of the results obtained through the following modelling.

3.1 Indexing of documents

The textual data obtained in the retrieval of the Web pages, Wikipedia articles or other text documents must be indexed in an efficient way to make the data easily accessible and compressed in some way to make the vast amounts of textual data manageable.

Although text is easily understandable to humans, it is still very difficult to have machines make sense of text sentences. One of the biggest problems is that text in its true nature is almost infinite dimensional for machines. The reason that humans can work with these high dimensional problems is our ability to grab the essential information from a text, and discard the rest of the information. This is because the essence of a sentence may be contained in a fraction of the words and few relations among these words.

The other problem for machine learning for text is the word order. The order in which the words appear carry much less information than the words themselves, and humans seem to understand something about a text by seeing which words are contained in the text, while none of the context can be recovered from knowing the order in which some unknown words appear. The order in which words appear can be very important though, when trying to understand a sentence. The two sentences ”are you old” and ”you are old” uses the same words, but the changes in word order, changes the meaning of the sentences from question to fact. If we look at the word order alone, we wouldn’t be able to make much sense out of it. The representation of the semantics for the two sentences could be ”1 2 3” and ”2 1 3” which wouldn’t make any sense at all. One reason that the word order carries little information about a text is that much word order information is contained in the grammatical rules of text, limiting the ways in which we can construct a sentence. Given the words ”are”, ”how” and ”you”, the grammatical rules dictate that the sentence must be ”how are you”.

Since the information about what words that occur in a text is much more valuable than the order in which they occur there is an easy way to represent text in a much more compact way that can be used for computers and machine learning algorithms to make sense of text. By considering only the counts of the words that appear in a text and not the order in which the words appear, text of any length is transformed from an infinite dimensional representation to a finite dimensional representation, allowing statistical machine learning algorithms to come into play. The transformation is not only simple, but also contains most of the information from the natural text.

In the vector space model, we represent documents and queries as vectors. Before we can construct the representation, we must compile a list of index terms by preprocessing the documents. We do this as follows:

1. Parse documents, removing all punctuation and numbers. Convert every letter to lower case.

2. Remove stop words. Stop words are common words that contain no semantic content such as ”of”, ”next”, and ”already.” These words are language and context dependent. Standard lists of stop words exist for the English language.

3. Stemming, i.e. reducing words to their semantic stem for instance transforming ’transformation’, ’transforming’ and ’transforms’ to a stem such as ’transform’.

4. Remove words that appear infrequently. Words that only occur few times in a corpus of text will typically be spelling errors. Words only occurring

in few documents do not contain enough discriminating power so these are also removed. This greatly reduces the number of terms we need to handle. The remaining words form the set of index terms (also called the vocabulary of the model).

The first step of indexing, i.e. the parsing step has been implemented in a number of systems, e.g. Rainbow[77], Lemur[2], and Lucene[3]. These systems build indexes including inverted files such that words can be mapped to documents easily. The words included in the index forms the vocabulary or dictionary used in the model.

Stop word removal is used to remove words that are deemed not to hold any semantic meaning. Numerous lists of stopwords have been compiled to include the most common and non-descriptive words. These lists are included in the standard indexing engines, but more specific stop word lists may also be useful for specific problems. In Web-based MIR [87], the emphasis of the indexing system is to be able to compare the musical terms related to musical artists or music pieces. In [87] a specialized music dictionary is assembled to only use musically descriptive. The dictionary has to be compiled manually based on some notion of which terms are important, which may be a hard task as the descriptions of music and the background knowledge used to describe music will be difficult to delimit.

The stemming is an attempt at addressing the problem of words reducing words that originate in root or base word to a common stem. This means that 'cats' are 'catlike' are reduced to the common stem 'cat'. Stemming for the English language is widely done using the Porter stemmer [92], which was further developed in the Snowball project [4]. These stemmers are based on simple lexical rules which may suffice for the English language. In other languages stemming is a non-trivial problem, which require more elaborate models of stemming. The simple stemming implemented in the Porter stemmers may sometimes perturb the meaning of some words, *blue* and *blues* will for instance both be reduced to the common stem *blue*, which in the context of music description is suboptimal. These problems must be addressed using more elaborate natural language processing that could identify the adjective 'blue' from the noun 'blues'.

3.1.1 Natural language processing techniques

Preprocessing of text can also employ more elaborate natural language processing (NLP) techniques, such as part-of-Speech tagging, noun phrase extraction or n-gram extraction.

A Part-Of-Speech (POS) tagger is an algorithm that reads text and for each token in the text returns the part-of-speech that the text-token belongs to, that is, whether it is a noun, verb or punctuation. The algorithms are usually based on statistical models that have reached very high precision today ($\sim$ 97% precision). The results from the POS tagger could be used in the vector space model directly for instance to disambiguate between the noun, adverb, adjective, and verb 'back' or as an extra bag-of-features in addition to the terms, as investigated in [72]. This study did conclude that the POS-tags provide improvements for small text corpora in a document classification setup.

The outputs from the POS tagger may also be used for extracting noun phrases. Noun phrases can be thought of as a noun extended with a maximal amount of descriptive text surrounding it. There is a defined grammar for noun phrase extraction, and once part-of-speech tagging has occurred, a simple rule-based noun phrase extractor can operate on any amount of text. Noun phrases suggest more than a simple bi- or trigram since their content is limited to one "idea". In the music domain, the sentence "Metallica employs screeching heavy metal guitars" leads to both "metal guitars" and "screeching heavy metal guitars" as noun phrases, but only the first is a possible bigram. The use of POS -tags noun phrases as well as n-grams was investigated in the context of MIR in the study [111]. The study shows that the NLP-techniques provides a way to attach more specific descriptions to artists, if one for instance wants to make very specific queries such as: "Something like Metallica, but quiet".

Natural language processing methods have also been developed to identify the language of a text. This was for instance employed by Mahedero et al. [73], for song lyrics processing. The knowledge of the language of documents will help in the following vector space representation, where different languages should be handled with different models for each language.

3.2 Vector space model

We will now formulate the vector space model [94], which is the basis of the following statistical modelling of text. After the preprocessing steps described in section 3.1, we have n documents and m index terms, hereafter referred to as just terms. We represent the documents as an $n \times m$ matrix, $\mathbf{X} = [x_{ij}]$, where x_{ij} represents the count of term i in document j. The jth column of $\mathbf{X}$

represents document j:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \dots & x_{1,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \dots & x_{n,m} \end{bmatrix} \tag{3.1}$$

This term × document matrix or term-doc matrix will be very sparse since only a few of the possible terms in the dictionary will appear in each document.

3.2.1 Queries in the vector space model

Given two documents in the vector space model i.e. two columns $\mathbf{x}_{.i}, \mathbf{x}_{.j}$ in the termdoc matrix, the vector space model measures the similarity of two documents using the cosine similarity:

$$sim(\mathbf{x}_{.i}, \mathbf{x}_{.j}) = \frac{\mathbf{x}_{.i}\mathbf{x}_{.j}}{\|\mathbf{x}_{.i}\|_2 \|\mathbf{x}_{.j}\|_2} \tag{3.2}$$

3.3 Term weighting schemes

The main factors determining the success or failure of vector space models are the parsing steps described above, and the term weighting described here. A review of the literature on term weighting presented in [117], identified the following principles of the term weighting approaches:

- Less weight is given to terms that appear in many documents
- More weight is given to terms that appear many times in a document
- Less weight is given to documents that contain many terms.

The observations are based on the idea that the terms that distinguish a document from the rest of collection are the most important, and therefore a weighting function should assign a higher weight for words that occur in some documents but not in all and assign lower weights to high-frequency terms that occur in many documents. The weighting function can have severe effects on the performance of the machine learning algorithms applied subsequently, so this is why we address the subject here at some length.

Method	Calculation
Binary	$\chi(f_{ij})$
Term frequency	f_ij
Augmented Normalized Term Frequency	$\frac{1}{2}\chi(f_{ij}) + \frac{1}{2}(\frac{f_{ij}}{\max_k f_{kj}})$
Log	$log(f_{ij} + 1)$
Alternate Log	$\chi(f_{ij})(log(f_{ij} + 1))$

Table 3.1: Local term weighting schemes

Method	Calculation
IDF	$g_t = \frac{1}{\sum_d \chi(f_{td})}$
GF-IDF	$g_t = \frac{\sum_d f_{td}}{\sum_d \chi(f_t d)}$
Probabilistic inverse	$g_t = \log\left(\frac{n - \sum_d \chi(f_{td})}{\sum_d \chi(f_{td})}\right)$
Entropy	$g_t = 1 + \sum_d \frac{h_{ij} \log h_{ij}}{\log n}$, with $h_{ij} = \frac{f_{td}}{\sum_{d'} f_{td'}}$

Table 3.2: Global term weighting schemes

The weighting of each element x_{ij} in the termdoc matrix is composed of three factors,

$$x_{ij} = g_i t_{ij} d_j. \tag{3.3}$$

The three factors are: g_i which is the global weight of the ith term, t_{ij} is the local weight of the ith term in the jth document, and d_j is the normalization factor for the jth document.

Some of the typical weighting schemes, as described in [17] are listed in tables 3.1, 3.2, and 3.3. f_{ij} designates the frequency of term i in document j in the original termdoc matrix, all logs are assumed to be in base two, and χ denotes the signum function,

$$\chi(t) = \begin{cases} 1, t > 0, \\ 0, t = 0. \end{cases} \tag{3.4}$$

The local term weight is the weight of a term t in each document d, so the weight only depends on the count of words within each single document. The local weight can be the term count f_{ij}, or a binary (0/1) indicator of the presence of the word. A major drawback of the binary formula is that it gives every word

Method	Calculation
No change	1
Normalised	$\sqrt{\sum_{k=1}^{m}(g_k t_{kj})^2}$

Table 3.3: Document normalisation schemes

that appears in a document equal relevance. This might, however, be useful when the number of times a word appears is not considered important. The pure frequency formula gives more credit to words that appear more frequently, but this is often too much credit. For instance, a word that appears ten times in a document is not usually ten times more important than a word that only appears once. So what one would like to do is to give credit to any word that appears and then give some additional credit to words that appear frequently. Therefore log weighting is very useful. Two different logarithm term frequency components are popular in the literature; here we call them the log and alternate log.

The global weighting schemes include the classical inverse document frequency (IDF), and others such as global frequency-IDF, probabilistic inverse, and entropy weighting. The calculation of the global are shown in table 3.2. All these global weighting schemes generally assign low weight to terms occurring in many documents. The use of global weighting can, in theory, eliminate the need for stop word removal since stop words should have very small global weights. In practice it is easier to remove the stop words in the preprocessing phase so that there are fewer terms to handle.

The document normalisation is crucial for document collections that contain a mixture of documents of varying length. If no normalisation is applied, long documents will dominate the results returned for a query. In addition to no normalisation or the Frobenius norm one could also use some other scaling, such that long documents still have a bigger weight than short documents.

3.4 Discussion

The preprocessing steps for text mining are as important as proper feature extraction in signal processing. The success of the following steps is primarily determined by these steps, and we have here sketched the main approaches for this preprocessing.

The vector space model using TF-IDF-weighting introduced here has shown to be very efficient in information retrieval and is often hard to beat in a known-item retrieval setup. But the method does for instance have problems with synonyms, so that a query for car will not return documents containing the word automobile.

Chapter 4

Latent semantic models

Learning the meaning of natural language (as well as pictures, audio and video) is an important task for accessing and retrieving the vast amounts of textual data available for instance on the Web. The approaches to automatic language understanding have traditionally been addressed either from a top-down perspective or a bottom up approach.

The top-down approach is the traditional linguistics school, which assumes that linguistic theory and logic can instruct computers to "learn" a language. This can for instance be accomplished by applying ontologies such as WordNet[38], which provides semantic relations between words based on linguistic relations such as antonyms, or hypernym (denoting a generic word that stands for a class or group of equally-ranked items, such as "car" which is a hypernym for the hyponyms "hatchback" and "sedan"). Ontologies have also been a standard way to incorporate topical knowledge in information retrieval, e.g. using the Open directory project[1] ontology which organises Web-pages hierarchically in groups. A review of using ontologies for information retrieval is found in [19]. The ontologies are usually employed in specialised applications where the database relates to a clearly delimited topical domain or where users have knowledge of the domain that can aid the search.

The initial TF-IDF measures have shown very useful and are a standard method

[1] http://www.dmoz.org/

of retrieving documents to a given query. In general search may have a tendency to get biased towards word meanings that are predominant in the corpus used in training the models. Web critics have for instance reported that Google, for instance, suffers perceived bias in some searches because of the overriding statistics of word usage in its corpus ("the Web") in contrast with their dictionary word senses: *on the internet an "apple" is a computer, not something you eat, "Madonna" is an often-times risque pop icon, not a religious icon, and moreover "latex" is not a typesetting system, but apparently something the certain people wear in certain situations.*[2] It may therefore be very useful to identify whether the topic of a search for Madonna should be biased towards the pop icon or Christianity. These kinds of problems can be addressed using the algorithms for extraction of latent semantics described in this chapter.

The bottom-up approaches described in this chapter therefore employ statistical methods to learn the semantics of text, based on (large) document collections and text corpora. The main challenge of the statistical methods is to bridge the gap between the lexical level of "what actually has been said or written" and the semantical level of "what was intended" or "what was referred to" in a text or an utterance. The main issues that are addressed in the following models are:

1. polysems, i.e., word that have multiple senses and multiple types of usage in different context

2. synonyms and semantically related words, i.e., words may have a similar meaning, or at least describe the same concept

The use of ontologies for information retrieval also seems to be a problem if the ontology is unknown to the users, so that one does not know how to navigate in the structure. Navigation will also be quite unwieldy if the ontology is very large, maybe consisting of 200 topics [102]. Inferring the possible topics automatically from a query was therefore investigated by Buntine et al. [26]. This paper investigates the use of a topic model with TF-IDF scoring of documents to retrieve documents from Wikipedia. They find that the use of "topic-guided" Pagerank and topic-based TF-IDF ranking of search results provides more meaningful results than purely using TF-IDF.

This section investigates different methods to extract these latent topics to aid retrieval of documents.

[2] http://slate.msn.com/id/2085668/index.html

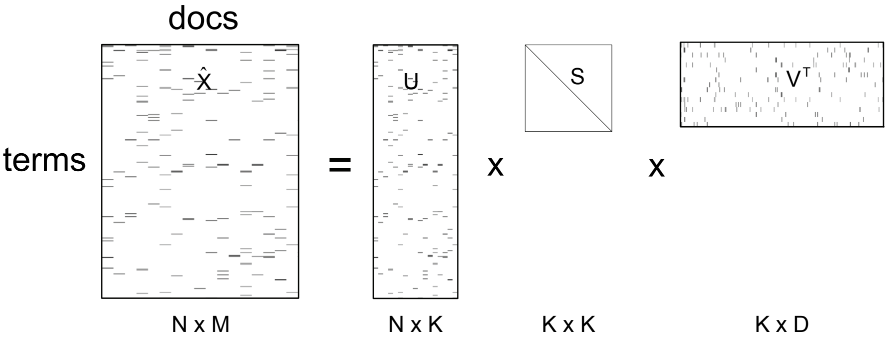


Figure 4.1: LSA model of term × document matrix. U contains the k term profiles, S is a diagonal matrix containing the k singular values, and $V^\top$ contains the document weightings.

4.1 Latent semantic analysis

The use of latent semantic models for information retrieval was introduced by Deerwester et al. [35]. The key idea in LSA is to map the high-dimensional count vectors in the vector space model, to a lower dimensional representation in a so-called latent semantic space. In doing so, LSA aims at finding a data mapping which provides information beyond the lexical level of word occurrences. The ultimate goal is to represent semantic relations between words and/or documents in terms of their proximity in the semantic space.

In practice the implementation of LSA model is based on a truncated Singular Value Decomposition (SVD) of the term-document matrix $\mathbf{X}$ which we defined in section 3.2;

$$\mathbf{X}_{(N\times M)} = \mathbf{U}_{(N\times M)}\mathbf{S}_{(M\times M)}\mathbf{V}^\top_{(M\times M)} \approx \hat{\mathbf{X}}_{(N\times M)} \approx \mathbf{U}_{(N\times K)}\mathbf{S}_{(K\times K)}\mathbf{V}^\top_{(K\times M)}, \tag{4.1}$$

graphically depicted in figure 4.1.

The truncated SVD can be calculated very efficiently because of the high sparsity of the term-doc matrix using sparse SVD-methods. For small problems one can solve the full SVD using dense solvers, but for a large $\mathbf{X}$ matrix this is not computationally possible. For sparse matrices it is possible to extract the k largest singular values and corresponding left and right singular vectors using Arnoldi-based methods [63, 16]. The development of these efficient numerical methods for SVD approximation makes the method useful even for millions of documents.

The space created using the truncated SVD has been shown to catch an underlying semantic space, where similar words are "near" each other in the K-dimensional space even though they don't occur in the same documents.[59] Because the components used to describe the documents are composed of the old terms, it is possible that a query using the word "buy" matches documents that does not contain this term, but contains other related terms such as "purchase". That is the new components represent an artificial "topic" described by values of the old terms, and it deals with synonymy of words, which are believed to have high values in the same component. In this way LSA can be argued to capture the latent semantics of the query. The two main problems of TF-IDFnamely handling synonymy and polysemy has been shown to be handled by LSA , for instance through tests using questions from the Test Of English as a Foreign Language (TOEFL)[59]. The test is based on choosing one of 4 words that is closest in meaning to a stem word. The performance was shown to be comparable to students who had English as their second language.

Polysemy is only partially solved by LSA. Since every term is represented by just one point in space it is not possible to represent multiple meanings of a word. Instead LSA provides a kind of averaging over the different meanings of polysemous words [35].

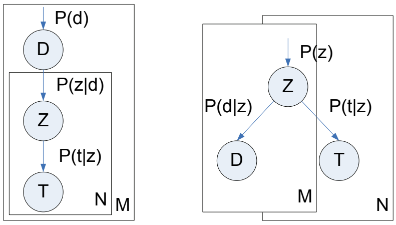
Queries in the LSA model are represented as a t-dimensional term vector $\mathbf{q}$, also called a "pseudo-document" [35], containing counts for the terms included in the query. The query vector should be weighted according to the weighting scheme used in the preprocessing of the term-doc matrix, i.e. using the same sort of term-weighting and the idf weights obtained from the corpus.

The query vector is then projected onto the K dimensional SVD-space,

$$\text{query} = \mathbf{q}^\top \mathbf{U}\Sigma^{-1} \tag{4.2}$$

The similarity between a query and a document is then evaluated by calculating the normalised cosine-distance, just as in the basic vector space model. As these calculations only employ simple dot products of vectors, queries can be calculated very efficiently.

The LSA model has shown to be useful in many applications of information retrieval, e.g. cross-language information retrieval [70], but also for other non-textual applications such as recommender systems [95]. The model has also been suggested to be a model how humans learn language, suggesting that it may be a good model of human cognition [59]. A basic problem of the LSA model is the underlying assumption that the terms can be represented in a Euclidean space. The spatial representation makes it difficult to interpret the latent dimensions of the LSA model, as the LSA components contain both positive and negative values for the loadings.



(a) asymmetric PLSA

(b) symmetric PLSA

Figure 4.2: Graphical model representation of PLSA models

4.2 Probabilistic Latent Semantic Analysis

A better approach to latent semantic analysis is Probabilistic Latent Semantic Analysis (PLSA), proposed by Hofmann [48], which like LSA, assumes that the high-dimensional term-document co-occurrence matrix can be represented in a lower dimensional latent semantic space, but formulates the latent semantics in terms of probabilistic topics. The PLSA model, or aspect model as it is also called, assumes that the data is generated from a latent variable model that associates an unobserved class variable, $z \in \{z_1, ..., z_k\}$, to each observation of an occurrence of a term, t in a document d.

The model is described using the following probabilities; $p(d)$ denotes the probability of a word occurring in a particular document d, $p(t|z)$ is the class-conditional probability of a specific word conditioned on the unobserved class variable z, $p(z|d)$ is a probability distribution for each document over the latent topics. Given these definitions the generative model of word, document co-occurrences is as follows:

1. select a document d with probability $p(d)$
2. pick a latent class z with probability $P(z_k|d)$,
3. generate a term t with probability $P(t|z)$.

PLSA can be expressed as a graphical model as depicted in figure 4.2, and can be written as a joint probability model as follows:

$$p(d,t) = p(d)p(t|d), p(t|d) = \sum_{k=1}^{K} p(t|z_k)p(z_k|d) \tag{4.3}$$

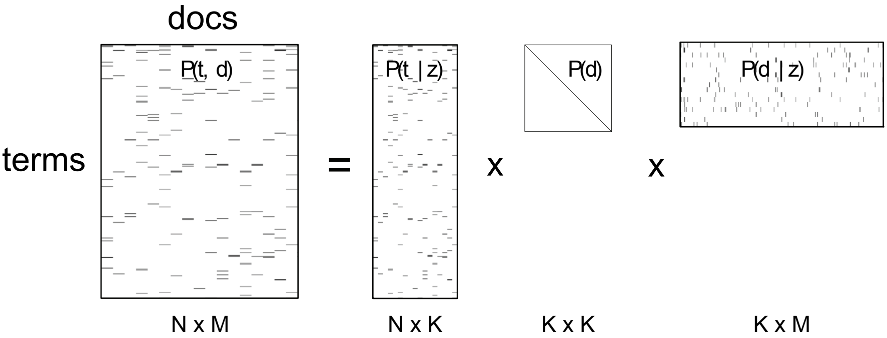


Figure 4.3: PLSA model of term × document matrix. $p(t|z)$ contains the K term profiles, the diagonal matrix contains the K topic probabilities $p(z)$, and $p(d|z)$ contains the conditional document probabilities

The model can also be represented by reversing the arrow between D and Z in the model figure 4.2(a), to form a model that is symmetric in words and documents. The parametrization of the model using this representation is:

$$p(d,t) = \sum_{k=1}^{K} p(z_k)p(t|z_k)p(d|z_k) \tag{4.4}$$

The model structure is thus quite similar to the original LSA formulation, as seen by comparing figure 4.1 and 4.3

4.2.1 Estimation of model parameters

Essentially, to obtain 4.4 one has to sum over the possible choices of z by which an observation could have been generated. The PLSA model introduces a conditional independence assumption, like other latent variable models, namely that the documents d and words t are independent conditioned on the state of the associated latent variable.

A very intuitive interpretation for the aspect model can be obtained by a closer examination of the conditional distributions $P(t|d)$ which are seen to be convex combinations of the K class-conditionals or aspects $P(t|z)$. Loosely speaking, the modeling goal is to identify conditional probability mass functions $P(t|z)$ such that the document-specific word distributions are as faithfully as possible approximated by convex combinations of these aspects.

The model fitting is done using maximum likelihood, i.e., we maximize,

$$\begin{aligned} p(X|model) &= p(t_1,d_1)^{x(t_1,d_1)}p(t_2,d_1)^{x(t_2,d_1)}\ldots p(t_n,d_m)^{x(t_N,d_M)} && (4.5)\\ &= \prod_d \prod_t p(t,d)^{x(t,d)} && (4.6) \end{aligned}$$

where the data observations, $x(w,d)$, is simply how many times a word occurred in a document, or the TF-IDF weight, i.e. the entries in the $\mathbf{X}$ matrix. The log likelihood of the data becomes

$$\begin{aligned} \log p(X|model) &= \sum_d \sum_t x(t,d) \log p(t,d) && (4.7)\\ &= \sum_d \sum_t x(t,d) \log \sum_z p(z)p(t|z)p(d|z) && (4.8)\\ & && (4.9) \end{aligned}$$

This function is difficult to optimize, because of the logarithm to a sum. We will make use of Jensen's inequality [20],

$$\log(\sum_j \lambda_j x_j) \geq \sum_j \lambda_j \log(x_j) \qquad (4.10)$$

where the λ_j's have to sum to one. The rule therefore is quite useful as we use the rule for the posterior probabilities of the latent variables, $p(z|t,d)$. This gives:

$$\begin{aligned} \log p(X|model) =& \sum_d \sum_w x(t,d) \log \sum_z p(z|t,d) \frac{p(z)p(t|z)p(d|z)}{p(z|t,d)} && (4.11)\\ \log p(X|model) =& \sum_d \sum_t x(t,d) \sum_z p(z|t,d) \log \frac{p(z)p(t|z)p(d|z)}{p(z|t,d)} && (4.12)\\ \equiv& F(\Theta, p(z|t,d)) && (4.13) \end{aligned}$$

The function, F, introduced represents a lower bound on the likelihood of the data, because of the inequality. If we want to maximize the likelihood of data then we also have to maximize L. The function depends partly on the parameters of the model, $\Theta = p(z), p(t|z), p(d|z)$, and partly on the mixing proportions, $p(z|t,d)$. By optimizing F with respect to the parameters and the mixing proportions respectively we will arrive at the two iterative updating steps for the EM algorithm.

The optimization of F has to constrain the densities to sum to one. Therefore we define a Lagrangian function, $L(\Lambda, \Theta, p(z|t,d))$, where we have added to F, the constraints multiplied by Lagrange multipliers, Λ. Now the gradient, ∇L, has to be zero in order to satisfy the constraints and in order to maximize F.

$$\begin{aligned}
\log p(X|model) =& \sum_d \sum_t x(t,d) \sum_z p(z|t,d) \log \frac{p(z)p(t|z)p(d|z)}{p(z|t,d)} && (4.14)\\
&+ \sum_z \lambda_z (\sum_t p(t|z) - 1) + \sum_z \mu_z (\sum_t p(d|z) - 1) && (4.15)\\
&+ \gamma(\sum_z p(z) - 1) + \sum_{t,d} \rho_{t,d} (\sum_t p(z|t,d) - 1) && (4.16)
\end{aligned}$$

We first find the updating rules for the parameters by differentiating L with respect to the parameters.

$$\begin{aligned}
0 =& \frac{\partial L}{\partial p(z)} = \sum_{d,t} p(z|t,d) \frac{p(z|t,d)}{p(z)p(t|z)p(d|z)} \frac{p(t|z)p(d|z)}{p(z|t,d)} + \gamma && (4.17)\\
p(z) =& -\frac{1}{\gamma} \sum_{d,t} x(t,d) \sum_z p(z|t,d) && (4.18)
\end{aligned}$$

The constant, $-\frac{1}{\gamma}$, can be found by summing over z and using the constraint that p(z) has to sum to one. That is

$$\begin{aligned}
\sum_z p(z) =& -\frac{1}{\gamma} \sum_{d,t} x(t,d) \sum_z p(z|t,d) = 1 && (4.19)\\
-\frac{1}{\gamma} =& \frac{1}{\sum_{d,t} x(t,d)} && (4.20)
\end{aligned}$$

And so we obtain the updating rule for p(z) to be

$$p(z) = \frac{\sum_{d,t} x(t,d)p(z|t,d)}{\sum_{d,t} x(t,d)} \qquad (4.21)$$

Derivation of the update rules for the other parameters, $p(t|z)$ and $p(d|z)$ is analogous, giving the following:

$$\begin{aligned}
p(t|z) =& \frac{\sum_d x(t,d)p(z|t,d)}{\sum_{d,t} x(t,d)p(z|t,d)} && (4.22)\\
p(d|z) =& \frac{\sum_w x(t,d)p(z|t,d)}{\sum_{d,t} x(t,d)p(z|t,d)} && (4.23)
\end{aligned}$$

Updating the parameters, $p(z)$, $p(t|z)$ and $p(d|z)$ composes the M-step of the EM algorithm, because the parameters are maximized given the current value of the latent variable, z.

The approach to find the mixing proportions, $p(z|t,d)$, is similar.

$$\begin{aligned} 0 = \frac{\partial L}{\partial p(z|t,d)} =& x(t,d)(\log \frac{p(z)p(t|z)p(d|z)}{p(z|t,d)} && (4.24) \\ & + p(z|t,d)\frac{p(z|t,d)}{p(z)p(t|z)p(d|z)}\left(-\frac{p(z)p(t|z)p(d|z)}{p(z|t,d)^2}\right)) + \rho_{t,d} && (4.25) \\ \Rightarrow p(z|t,d) =& e^{-\frac{\rho_{t,d}+1}{x(t,d)}} p(z)p(t|z)p(d|z) && (4.26) \end{aligned}$$

where the constant in front of the parameters can be found by summing over z and utilizing the constraint on $p(z|t,d)$, so that

$$\begin{aligned} \sum_z p(z|t,d) =& e^{-\frac{\rho_{t,d}+1}{x(t,d)}} \sum_{d,t} p(z)p(t|z)p(d|z) = 1 && (4.27) \\ e^{-\frac{\rho_{t,d}+1}{x(t,d)}} =& \frac{1}{\sum_{d,t} p(z)p(t|z)p(d|z)} && (4.28) \end{aligned}$$

And the updating rule for $p(z|t,d)$ becomes

$$p(z|t,d) = \frac{p(z)p(t|z)p(d|z)}{\sum_z p(z)p(t|z)p(d|z)} \quad (4.29)$$

This step then constitutes the Expectation step of the EM-algorithm.

4.2.2 PLSA properties

A comparison in terms of computational complexity might suggest some advantages for LSA : ignoring potential problems of numerical stability, the SVD can be computed exactly, while the EM algorithm is an iterative method which is only guaranteed to find a local maximum of the likelihood function.

The PLSA model has two issues,

1. How much does the model accuracy suffer from the fact that EM is only able to find a local maximum?
2. The cost of computing a solution, i.e., how does the EM algorithms scale with the size of the data set?

The issue of not finding the global maximum can be addressed by using regularization techniques like early stopping or tempered EM [49]. Another approach is to use a number of random initialisations of algorithm and pick the best solution.

The number of arithmetic operations used to train the model of course depends on the number of EM iterations that have to be performed. Each iteration requires $\mathcal{O}(R \times K)$ operations, where R is the number of distinct observation pairs (d, t) i.e., the dimensionality of the termdoc matrix, $N \times M$, multiplied by the degree of sparseness of the termdoc matrix. This can easily be verified as there are $R \times K$ posterior probabilities have to be computed in the E-step 4.29 each of which contributes to exactly one re-estimation of the posterior probabilities in equation 4.22.

4.3 Nonnegative Matrix Factorisation

As was also noted by Hofmann [49], PLSA is related to Nonnegative Matrix Factorisation (NMF), we will return to the relation later. Non-negative matrix factorization (NMF) is a method for finding parts-based representations as described by [61]. The method was also investigated in an earlier work [84], as positive matrix factorisation. The basic approach is to approximate a matrix, $\mathbf{X}$, as the product of two matrices, $\mathbf{W}$ and $\mathbf{H}$, under the constraint that all elements in the factorising matrices are non-negative,

$$\mathbf{X} \approx \mathbf{WH}, \tag{4.30}$$

where $\mathbf{X} \in \mathbb{R}^{N \times M}_{\mathbb{R} \setminus \mathbb{R}_-}$, $\mathbf{W} \in \mathbb{R}^{N \times K}_{\mathbb{R} \setminus \mathbb{R}_-}$, $\mathbf{H} \in \mathbb{R}^{K \times M}_{\mathbb{R} \setminus \mathbb{R}_-}$, so that all elements in $\mathbf{X}$, $\mathbf{W}$, and $\mathbf{H}$ are non-negative. This definition fits well with the count matrices we operate with in the vector space model. NMF is related to many other dimensionality reduction techniques used in text mining, such as vector quantization (VQ), SVD used in classic LSA, and independent component analysis (ICA) [58], that can all be written as matrix factorizations on the form $\mathbf{X} \approx \mathbf{WH}$. The differences between these methods and NMF can be seen in the constraints used to obtain the factorizing matrices, $\mathbf{W}$ and $\mathbf{H}$: in VQ the columns of $\mathbf{H}$ are constrained to be unary vectors (all zero except one element equal to unity); in SVD columns of $\mathbf{H}$ and rows of $\mathbf{W}$ are constrained to be orthogonal; in ICA rows of $\mathbf{H}$ are maximally statistically independent; and in NMF all elements of $\mathbf{W}$ and $\mathbf{H}$ are non-negative. Several hybrid methods that combine these constraints have also been proposed, such as non-negative PCA [90] and non-negative ICA [89].

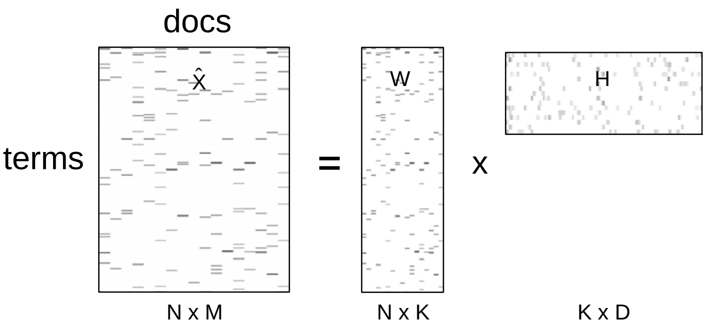


Figure 4.4: NMF model of term × document matrix. $\mathbf{W}$ contains the K term profiles, and $\mathbf{H}$ contains the document loadings.

4.4 Computing the NMF

NMF can be computed as a constrained optimization problem

$$\{\mathbf{W}, \mathbf{H}\} = \arg \min_{\mathbf{W},\mathbf{H} \geq 0} \mathcal{D}(\mathbf{X}; \mathbf{W}, \mathbf{H}), \tag{4.31}$$

based on the cost function $\mathcal{D}$. In principle, any constrained optimization algorithm can be used to compute $\mathbf{W}$ and $\mathbf{H}$. In the literature, many different algorithms have been proposed for NMF, many of which take advantage of the special structure of the NMF problem as well as properties of specific cost functions. Albright et al. [7], Berry et al. [18], and Sra and Dhillon [101] review a broad range of different algorithms. The most popular cost functions used in the algorithms are usually either least squares or Kullback-Leibler divergence.

4.4.1 Optimization strategies

Several optimization algorithms have been proposed in the literature, that can be divided into three categories: direct optimization methods, alternating optimization methods, and alternating descent methods.

Direct optimization methods solve the NMF problem,

$$\mathbf{W}, \mathbf{H} \leftarrow \arg \min_{\mathbf{W},\mathbf{H} \geq 0} \mathcal{D}(\mathbf{X}; \mathbf{W}, \mathbf{H}), \tag{4.32}$$

directly using some (general-purpose) bound constrained optimization algorithm. In general, this is a non-negativity constrained non-linear optimization problem, for which many efficient algorithms exist. Since the number of parameters, $(N + M)K$, in the full standard NMF problem is usually very large (especially in text mining problems), it will clearly be infeasible to use optimization methods that require the explicit computation of a Hessian matrix.

Alternating optimization methods partition the NMF problems into two subproblems for the matrices **W** and **H**, that are solved in alternating turns until convergence, In each iteration, the NMF problem is solved for **W** while **H** is kept fixed and vice versa, this is then repeated until **W** and **H** converge to a solution of the full NMF problem.

Algorithm 1 Alternating optimisation NMF method

```
repeat
    $\mathbf{W} \leftarrow \arg\min_{\mathbf{W}\geq 0} \mathcal{D}(\mathbf{X}; \mathbf{W}, \mathbf{H})$
    $\mathbf{H} \leftarrow \arg\min_{\mathbf{H}\geq 0} \mathcal{D}(\mathbf{X}; \mathbf{W}, \mathbf{H})$
until convergence criterion is met
```

In general, alternating optimization may have several advantages over direct optimization. While the full NMF problem is not jointly convex in **W** and **H**, some cost functions have the desirable property that the subproblems are convex in the parameters, which allows the computation of the globally optimal solution of each subproblem in each step. Also, for some cost functions, the rows of **W** are decoupled when **H** is fixed which means that each subproblem consists of N independent problems, and similarly of M independent problems for fixed **W**. The division of the problem is for instance useful for the least squares cost function, where the subproblems are sets of non-negatively constrained least squares problems that can be solved efficiently [23].

Alternating descent methods relax the previously described approach by not computing an optimal solution for each subproblem in each step. Instead, an approximate solution is computed at each update of **W** and **H**, so that the cost function is reduced, but not necessarily minimized.

Algorithm 2 Alternating descent NMF

```
repeat
    $\mathbf{W} \leftarrow \mathbf{W}^*$ where $\mathcal{D}(\mathbf{X}; \mathbf{W}*, \mathbf{H}) < \mathcal{D}(\mathbf{X}; \mathbf{W}, \mathbf{H})$
    $\mathbf{H} \leftarrow \mathbf{H}^*$ where $\mathcal{D}(\mathbf{X}; \mathbf{W}, \mathbf{H}*) < \mathcal{D}(\mathbf{X}; \mathbf{W}, \mathbf{H})$
until convergence criterion is met
```

This approach can be advantageous when an optimal solution of each subproblem can be computed by an iterative procedure where each iteration is fast and guaranteed to reduce the cost function. In this case, the method proceeds in turns by computing a single iteration on each subproblem. Although algorithms of this type reduce the cost function in each iteration, there is not in general any guarantee that the algorithm will converge to a local minimum of the NMF cost function. The multiplicative update algorithms proposed by Lee and Seung [61], which is described below, are examples of alternating descent NMF methods.

4.5 NMF algorithms

This section presents a number of NMF algorithms that were investigated in the context of topic modelling. We will use a simplified vector notation,

$$\min_{\mathbf{x}\geq 0} f(\mathbf{x}), \tag{4.33}$$

to describe either the full NMF problem of eq. 4.32 or a sub-problem in an alternating optimization strategy as in algorithm 2.

4.5.1 Multiplicative updates

The interest in Non-negative Matrix Factorisation was probably kick-started by the Nature paper [61]. They present an iterative NMFalgorithm with multiplicative updates, that can be seen as a gradient descent algorithm with a specific choice of step size. When the gradient can be expressed as the subtraction of two non-negative terms, $\nabla_f(\mathbf{x}) = \nabla_f(\mathbf{x})^+ - \nabla_f(\mathbf{x})^-$, a step size can be chosen individually for each element of $\mathbf{x}$ as $\alpha_i = x_i/\nabla_f(\mathbf{x})_i^+$, which leads to a multiplicative gradient descent update

$$x_i \leftarrow x_i \frac{\nabla_f(\mathbf{x})_i^-}{\nabla_f(\mathbf{x})_i^+}. \tag{4.34}$$

One of the main advantages of this algorithm is that the updates in this algorithm are formulated as a multiplication by a non-negative quantity, it is ensured that x remains non-negative, if it is initialized with positive elements. This approach is used in [61] to derive a least squares multiplicative update algorithm which is given as algorithm 3, and the Kullback-Leibler divergence based method that is given as algorithm 4. The initial value of $\mathbf{x}$ must be strictly positive, since any elements that are zero will remain zero in the following iterations. [61] prove for the least squares and Kullback-Leibler divergences that the multiplicative updates are guaranteed to reduce the cost function in each step, and that the update rules are unity only at stationary points of the cost function. However, the proofs of convergence do not guarantee that the algorithm will converge to a stationary point within any reasonable number of iterations, as discussed in [43] and [69]. Recently [81] also showed empirically that the convergence of the multiplicative updates is much slower than other optimizations methods, and may not even converge to the global minimum.

Algorithm 3 least-squares multiplicative updates

```
Input: initial $\mathbf{W} \in \mathbb{R}^{N \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times M}$
  repeat
    $\mathbf{W}_{n,k} \leftarrow \mathbf{W}_{n,k} \frac{(\mathbf{XH})_{n,k}}{(\mathbf{WHH}^\top)}$
    $\mathbf{H}_{k,m} \leftarrow \mathbf{H}_{k,m} \frac{(\mathbf{W}^\top \mathbf{X})_{k,m}}{(\mathbf{W}^\top \mathbf{WH})_{k,m}}$
  until convergence criterion is met
```

Algorithm 4 Kullback-Leibler multiplicative updates

```
Input: initial $\mathbf{W} \in \mathbb{R}^{N \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times M}$
  repeat
    $\mathbf{W}_{n,k} \leftarrow \mathbf{W}_{n,k} \frac{\sum_m \mathbf{H}_{k,m} \frac{X_{n,m}}{(\mathbf{WH})_{n,m}}}{\sum_{m'} (\mathbf{H}_{k,m'})}$
    $\mathbf{H}_{k,m} \leftarrow \mathbf{H}_{k,m} \frac{\sum_n \mathbf{W}_{n,k} \frac{X_{n,m}}{(\mathbf{WH})_{n,m}}}{\sum_{n'} (\mathbf{H}_{n',k})}$
  until convergence criterion is met
```

4.5.2 Projected gradient descent

Lin [67] proposes the use of projected gradient descent methods for NMF, used for either direct or alternating optimization. The approach is simple in that it searches for a local minimum of the cost function by iteratively just taking steps in the direction of the negative gradient,

$$\mathbf{x} \leftarrow \mathbf{x} - \alpha \nabla_f f(\mathbf{x}), \tag{4.35}$$

where α is a step size parameter and $\nabla_f(\mathbf{x})$ is the gradient. The only problem is that the gradient may produce negative values for $\mathbf{x}$. The projected gradient methods therefore extend the basic gradient descent by taking steps that are projected onto the feasible region (the non-negative orthant),

$$\mathbf{x} \leftarrow \max\left[\mathbf{x} - \alpha \nabla_f(\mathbf{x}), 0\right], \tag{4.36}$$

In contrast to the multiplicative update algorithm described in section 4.5.1, which uses a fixed step size, α must be chosen as a constant, by an adaptive procedure, or by line search, as used in [68].

A simple projected gradient is given as algorithm 5, using the function R;

$$R(x) = \begin{cases} x, x > 0, \\ 0, otherwise \end{cases} \tag{4.37}$$

that performs the projection into the non-negative orthant.

Algorithm 5 Alternating least-squares projected gradient descent

```
Input: Step-size $\alpha$, initial $\mathbf{W} \in \mathbb{R}^{I \times N}$ and $\mathbf{H} \in \mathbb{R}^{N \times J}$
  repeat
    $\mathbf{W} \leftarrow R\left(\mathbf{W} - \alpha(\mathbf{W}\mathbf{H}\mathbf{H}^\top - \mathbf{X}\mathbf{H}^\top)\right)$
    $\mathbf{W} \leftarrow R\left(\mathbf{H} - \alpha(\mathbf{W}^\top\mathbf{W}\mathbf{H} - \mathbf{W}^\top\mathbf{X})\right)$
  until convergence criterion is met
```

4.5.3 Hybrid methods and sparsity

The NMF does not always produce a parts-based representation as described by [61], which led to the suggestion by Hoyer [50], of using a sparseness constraints on the factors to enforce the parts-based representation. The method proposed in [50] has an important feature that enforces a statistical sparsity of the **H** matrix. As the sparsity of **H** increases, the basis vectors become more localised, i.e., the parts-based representation of the data in **W** becomes more and more distinct.

The use of sparse representations for topic detection was for instance proposed by [100], using a hybrid method that includes the sparsity constraint in the document loadings **H**, called GD-CLS (Gradient Descent with Constrained Least Squares). In this approach, the multiplicative method from section 4.5.1, is used at each iterative step to approximate the basis vector matrix **W**, forming the gradient descent part of the algorithm. The second step is then to calculate **H** using a constrained least squares (CLS) as the metric. The CLS serves to penalise the non-smoothness and non-sparsity of **H**. As a result of this penalisation, the basis vectors or topics in **W** should become more localised, thereby reducing the number of active parameters needed to represent each document. We investigated the use of the GD-CLS algorithm for our semantic modelling as it was shown to work well in a text mining setup for topic detection, e.g., in [100, 16].

The algorithm for GD-CLS is shown as algorithm 6. The subscript i for the **H** matrix denotes the ith column, for $i = 1, ..., m$. Any negative values in H_i are

Algorithm 6 Gradient descent with constrained least squares

```
Input: Step-size $\alpha$, initial $\mathbf{W} \in \mathbb{R}^{N \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times M}$
  repeat
    $\mathbf{W}_{n,k} \leftarrow \mathbf{W}_{n,k} \frac{(\mathbf{XH})_{n,k}}{(\mathbf{WHH}^\top)}$
    Rescale the columns of W to unit norm
    Solve the constrained least squares problem:
        $$\min_{\mathbf{H}_m} \left\{ \|\mathbf{V} - \mathbf{WH}\|_2^2 + \lambda \|H_m\|_2^2 \right\} \qquad (4.38)$$
  until convergence criterion is met
```

set to zero. The parameter λ is a regularisation value that is used to balance the reduction of the metric $\|\mathbf{V} - \mathbf{WH}\|_2^2$ with the enforcement of smoothness and sparsity in $\mathbf{H}$.

4.6 Relation between PLSA and NMF

As noted earlier the similarity of PLSA and NMF has been noted. The similarity of the NMF model and PLSA has been investigated in some studies. Buntine [25] derives a so-called multinomial PCA (LDA), and also shows that NMF and PLSA can be seen as instances of the multinomial LSA depending on the normalisation and prior distributions used. The paper by Gaussier and Goutte [39], more formally shows that the PLSA algorithm solves the NMF problem with a Kullback-Leibler divergence, given a special normalisation step in the NMF estimation.

The main advantage of the PLSA as stated in [39] is the probabilistic setting, so it is possible to benefit from the considerable body of work dedicated to address the various shortcomings of EM, as mentioned in section 4.2. The probabilistic approach also provides the possibility to use techniques for model selection. Additionally the PLSA model has hierarchical extensions that may be useful for clustering and text categorisation, where it is common to have tree structured organisations of classes. The probabilistic setup also gives principled ways to include additional features in the model, for instance including link information[34].

The NMF method on the other hand provides very efficient update formulae, especially using the least squares objective function. The most costly step of the PLSA algorithm and Kullback Leibler divergence (KL)-based NMF algorithm 4 is the reconstruction of the data matrix. In our work [79], we therefore suggest modification of the PLSA framework the model is based on a term-doc matrix that is normalised such that it can be interpreted as frequency estimates of model probabilities $\hat{\mathbf{X}}_{td} = \mathbf{X}_{td} / \sum_{td} \mathbf{X}_{td} \approx p(t, d)$. We approximate the optimal non-negative factorization $\mathbf{X} = \mathbf{WH}$, using instead a Euclidean loss which essentially amounts to a 'large count' limit in which the count distributions can be approximated by Gaussians. Thus $p(t|k)$ and $p(d|k)$ of the PLSA-model are identified as $\mathbf{W}$ and $\mathbf{H}$ of the NMF respectively, if columns of $\mathbf{W}$ and rows of $\mathbf{H}$ are normalized as probability distributions.

$$p(t, d) = \sum_{k=1}^{K} \mathbf{W}_{t,k} \mathbf{H}_{k,d} = \sum_{k=1}^{K} \frac{\mathbf{W}_{t,k}}{\alpha_k} \frac{\mathbf{H}_{k,d}}{\beta_k} \alpha_k \beta_k, \tag{4.39}$$

where $\alpha_k = \sum_t \mathbf{W}_{t,k}$, $\beta_k = \sum_d \mathbf{H}_{k,d}$, and $p(k) = \alpha_k \beta_k$. Thus, the normalized $\mathbf{W}$ is the probability of term t given a context k, while the normalized $\mathbf{H}$ is the probability of document d in a context k. $p(k)$ can be interpreted as the prior probability of context k.

4.7 Other topic models

The assumptions of the PLSA-model, i.e. that the topic distribution is given by a single multinomial distribution has been questioned, as it is not a generative model for new documents. Another shortcoming that has been emphasized is that the number of parameters grows with the number of documents in the training data. Based on these observations [21] propose a more elaborate model called Latent Dirichlet Association (LDA).

The formulation of the LDA model describes the document collection of the n terms t and m documents d, using the following generative procedure for each document:

1. Choose the number of terms, $n \sim Poisson(x)$.
2. Choose $\theta \sim Dirichlet(\alpha)$.
3. For each of the n terms t_i:
   (a) Choose a topic $z_i \sim Multinomial(\theta)$.
   (b) Choose a term t_i from $p(t_i|z_i, \beta)$, which is a multinomial probability conditioned on the topic z_i.

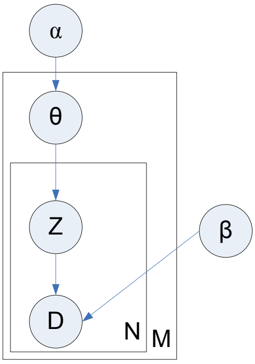


Figure 4.5: Graphical model depicting the LDA model. The PLSA model is extended with hyperparameters to make the model a true generative model of words and documents.

The LDA model thus expands the PLSA model by adding hidden parameters to control the topic mixture weights, as seen in graphical representation in figure 4.5. This addition means that the number of model parameters do not grow with the number of training documents, countering overfitting problems of PLSA. The LDA model can be trained using a variational procedure, but requires some dexterity to train, and has therefore not been considered in this work.

4.8 Discussion

The latent topic modelling methods presented above are a useful statistical tool to infer contextual knowledge from text. The probabilistic models and NMF should provide interpretable topics, which makes them useful for describing the relation meaning between documents, as we will show in the following chapters.

The sorts of correlations that are captured by the latent semantic models may sometimes be somewhat spurious, so methodologies to steer the relations inferred from such models must be used, e.g. as in the synonymy tests described.

One issue of these models is how to choose the number of latent factors. The standard LSA approach does not provide a principled way to choose the number of topics, while the probabilistic view gives a number of ways to perform model selection. We will return to these issues in the following chapters.

Chapter 5

Context based spoken document retrieval

This chapter describes work on using context for retrieval of spoken documents, i.e. audio recordings containing speech. Broadcast news and other podcasts often include multiple speakers in widely different environments which makes indexing hard, combining challenges in both audio signal analysis and text segmentation. This chapter describes our work on the development of a system for retrieval of relevant stories from broadcast news. The system utilizes a combination of audio processing and text mining. The work was published in the paper in appendix B. This chapter mainly focuses on the use of latent semantics for spoken document retrieval.

5.1 Spoken document retrieval

The efforts in spoken document retrieval were initially spurred by the U.S. National Institute of Standards and Technology (NIST) in the evaluations at the Text REtrieval Conferences (TREC) [28], where the speech recognition and information retrieval communities joined forces to develop systems especially for broadcast news retrieval. These efforts resulted in many successful systems, for instance in the SpeechBot search system developed at HP Labs. The retrieval of further information on spoken documents was for instance in the 'Speechfind' project described in [46]. Other notable examples of combining multimedia analysis for better retrieval performance have for instance been investigated in projects or systems such as the Informedia System at Carnegie Mellon University [1], the Rough'n'Ready System at BBN Technologies [74], and the Speech Content-based Audio Navigator (SCAN) System at AT&T Labs-Research [113], the SpeechBot Audio/Video Search System at Hewlett-Packard (HP) Labs [107]. These successful projects have shown the importance and potential of improved technologies for efficient retrieval/browsing for the massive quantities of spoken documents and multimedia content. In general we see that there is a need for a combination of methods for signal processing, and post-processing methods to organise and present the multimedia data to users.

Broadcast news and other spoken documents are often very long recordings containing many topically different segments. Access to these recordings can thus be aided by dividing the recordings into topically or at least auditorily coherent parts. This is for instance an important part of the Speechfind search engine. The segmentation of broadcast news to find topic boundaries can be approached at two different levels, starting from analysis of the audio, or using text segmentation. We can use audio analysis to locate parts that contain one speaker in the same environment, such a speaker turn can be used to find the structure in the recording, and be used to indicate when one story ends, and the next one starts. This segmentation can also be used to improve automatic speech recognition performance. We investigated this approach in earlier work [53].

Performing speech recognition on the speech segments enables a top-down segmentation based on the semantic content of the news stories. The area of topic detection in text has been widely researched for the last decade, see e.g., [8] for a presentation. Automatically transcribed text poses additional difficulties than human-made transcripts, due to the imperfect transcriptions produced by the automatic speech recognition. The use of topic extraction for spoken document organisation was also presented in [62, 31], the papers present data-driven topic detection through PLSA which is used to extract summaries of the spoken documents and for visualisation of retrieved documents.

The use of automatic clustering or topic detection to present document collections to users has been utilised in different systems, for instance in an academic search engine for Wikipedia [26], and in commercial systems such as the Clusty search engine[1], where users can narrow in their search results using clusters.

The main contributions of the work presented in this chapter is in the use of

[1] Available at `http://clusty.com`

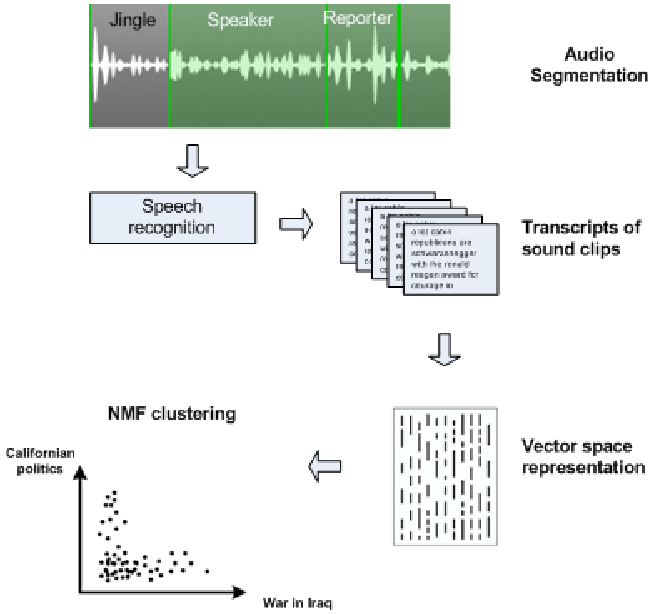


Figure 5.1: The system setup. The audio stream is first processed using audio segmentation. Segments are then using an automatic speech recognition (ASR) system to produce text segments. The text is then processed using a vector representation of text and non-negative matrix factorization (NMF) to find a topic space.

topic extraction techniques for query expansion and explanation of search results in the context of spoken document retrieval. The context knowledge obtained through the topic modelling is shown to be a good postprocessing step to cope with the more or less erroneous output from the automatic audio analysis and speech recognition.

5.2 System overview

The system presented here operates in two domains combining audio based processing and text based document retrieval. The overall system pipeline is shown in figure 5.1. The audio segmentation part as well as the speech recognition part was developed and described in [52] and [53] and will only be briefly outlined here. The main focus is on the possibilities and work done in the text processing step.

5.2.1 Audio segmentation

The basic sound representation is a feature set consisting of 12-dimensional Mel Frequency Cepstral Coefficients (MFCC's) as well as features included in the set proposed by the Moving Picture Experts Group (MPEG) such as the zero-crossing rate, short-time energy, and spectral flux. The first step is to separate speech parts, excluding 'jingles'. The step is performed in a supervised classification step using a trained linear classifier. The classifier operates on 1 sec windows and detects the two classes speech and music. The audio classification step was shown to have a correct classification rate of 97.8%, see [52] for additional details and feature set evaluation.

To aid topic spotting and locate story boundaries in the news stream we use additional audio cues to segment the audio into sub-segments containing only one speaker in a given environment. Broadcast news shows typically have unknown number and identities of speakers, including news anchors and guests appearing both in the radio studio and in the field. Therefore we invoke an unsupervised speaker change detection algorithm, described in detail in [53]. The algorithm is based on a 12-dimensional MFCC feature set, that is statistically summarized by vector quantization in sliding windows on both sides of a hypothesized change-point. The similarity of these windows is then measured using the vector quantization distortion (VQD) and possible speaker changes are detected by simple thresholding. A second step is invoked for removal of false alarms inferred by the change detection algorithm. This step uses larger windows around a hypothesized change-point to yield more precise statistically models. An overall F-measure of 0.85 is found using this algorithm [52].

5.2.2 Automatic speech recognition

The sound clips produced by the segmentation step is transcribed using the Sphinx4 [109] speech recognition system. Sphinx4 is a large vocabulary speaker independent open source speech recognition system from Carnegie Mellon University. The system was set up using the pretrained acoustic and language models available on the Sphinx4 Web-page [2]. Using these models Sphinx4 gives a word accuracy of $50-80\%$ depending on the speaking style of the speaker and the background noise level. This accuracy could be improved by training models on the material used in our experiments but this is not the focus of this work, and the performance of the model will therefore suffice.

[2] http://cmusphinx.sourceforge.net

5.3 Topic detection

The result of the audio segmentation and speech recognition performed on these clips are considered as a collection of documents. The database of documents could be approached using standard indexing methods. To further increase the user friendliness we propose to use text modeling tools to spot relevant contexts of the documents.

5.3.1 Document representation

Given the speaker documents $D = \{d_1, d_2, ..., d_m\}$ and the vocabulary set $T = \{t_1, t_2, ..., t_n\}$, the $n \times m$ term-document matrix $\mathbf{X}$ is created, where $\mathbf{X}_{i,j}$ is the count of term t_i in the speaker document d_j.

The columns of $\mathbf{W}$ forms a r-dimensional semantic space, where each column can be interpreted as a context vocabulary of the given document corpus. Each document, columns of $\mathbf{H}$, is hence formed as a linear combination of the contexts.

5.3.2 NMF for document retrieval

Let N be the sum of all elements in $\mathbf{X}$. Then $\widetilde{\mathbf{X}} = \frac{\mathbf{X}}{N}$ form a frequency table approximating the joint probability of terms t and documents d

$$\widetilde{\mathbf{X}} \equiv \frac{\mathbf{X}}{N} \approx p(t, d). \tag{5.1}$$

Expanding on a complete set of disjoint contexts k we can write $p(t, d)$ as the mixture

$$p(t, d) = \sum_{k=1}^{K} p(t|k)p(d|k)p(k), \tag{5.2}$$

where $p(t|k)$ and $p(d|k)$ are identified as $\mathbf{W}$ and $\mathbf{H}$ of the NMF respectively, if columns of $\mathbf{W}$ and rows of $\mathbf{H}$ are normalized as probability distributions

$$\begin{aligned} p(t, d) &= \sum_{k=1}^{K} \mathbf{W}_{t,k}\mathbf{H}_{k,d} && (5.3) \\ &= \sum_{k=1}^{K} \frac{\mathbf{W}_{t,k}}{\alpha_k} \frac{\mathbf{H}_{k,d}}{\beta_k} \alpha_k \beta_k, && (5.4) \end{aligned}$$

where $\alpha_k = \sum_t \mathbf{W}_{t,k}$, $\beta_k = \sum_d \mathbf{H}_{k,d}$, and $p(k) = \alpha_k \beta_k$. Thus, the normalized $\mathbf{W}$ is the probability of term t given a context k, while the normalized $\mathbf{H}$ is the probability of document d in a context k. $p(k)$ can be interpreted as the prior probability of context k.

The relevance (probability) of context k given a query string d^* is estimated as

$$p(k|d^*) = \sum_t p(k|t)p(t|d^*), \tag{5.5}$$

where $p(t|d^*)$ is the normalized histogram of (known) terms in the query string d^*, while $p(k|t)$ is found using Bayes theorem using the quantities estimated by the NMF step

$$\begin{aligned} p(k|t) &= \frac{p(t|k)p(k)}{\sum_{k'} p(t|k')p(k')} && (5.6) \\ &= \frac{\mathbf{W}_{t,k}p(k)}{\sum_{k'} \mathbf{W}_{t,k'}p(k')}. && (5.7) \end{aligned}$$

The relevance (probability) of document d given a query d^* is then

$$\begin{aligned} p(d|d^*) &= \sum_{k=1}^{K} p(d|k)p(k|d^*) && (5.8) \\ &= \sum_{k=1}^{K} \mathbf{H}_{k,d}p(k|d^*). && (5.9) \end{aligned}$$

The relevance is used for ranking in retrieval. Importantly we note that high relevance documents need not contain any of the search terms present in the query string. If the query string invokes a given subset of contexts the most central documents for these context are retrieved. Thus, the NMF based retrieval mechanism acts as a kind of association engine: 'These are documents that are highly relevant for your query'.

5.4 Broadcast news retrieval evaluation

In the following section we evaluate the use of NMF for topic detection and document retrieval.

To form a document database 2099 CNN-News podcasts were automatically transcribed and segmented into 30977 speaker documents, yielding a vocabulary

Label	No. segments
Crisis in Lebanon	8
War in Iraq	7
Heatwave	7
Crime	5
Wildfires	1
Hurricane season	2
Other	30
Total	60

Table 5.1: The specific contexts used for evaluation by manual topic delineation.

of 37791 words after stop-word removal. The news show database was acquired during the period 2006-04-04 to 2006-08-09.

Based on the database a term-document matrix was created and subjected to NMF decomposition using $K = 70$ contexts producing matrices $\mathbf{W}$ and $\mathbf{H}$. The implementation of the NMF-algorithm was done using the approach described in [67]. For each context the ten most probable terms in the corresponding column of $\mathbf{W}$ were extracted as keywords. The choice of 70 contexts for the model was done by qualitatively evaluating a range of values for K by inspection of the keywords extracted for the contexts. Fewer components generally seemed to cluster topics that were not semantically similar, while a larger number of contexts produced very small topics that effectively only contains single documents. Based on the keyword list each context was manually labeled with a short descriptive text string (one-two words).

For evaluation of the topic detector eight CNN-News shows were manually segmented and labeled in a subset of six contexts out of the $K = 70$ contexts identified by NMF. Segments that were not found to fall into any of six topics were labeled as 'other'. The six labels and the number of segments for each label can be seen in table 5.1.

5.4.1 NMF for query expansion

As described above the probabilistic interpretation of the NMF factorization allows query expansion by 'association'.

To illustrate the usefulness of this system let us consider a specific example. We query the system with the term 'schwarzenegger', the governor of the state of California. The query expansion first uses equation (5.5) to evaluate prob-

> ... california governor arnold's *fortson agar* inspected the california mexico border by helicopter wednesday to see ...
>
> ... president bush asking california's governor for fifteen hundred more national guard troops to help patrol the mexican border but governor orville *schwartz wicker* denying the request saying...

Figure 5.2: Two examples of the retrieved text for a query on 'schwarzenegger'.

abilities of the contexts given the query string. The result of the 'schwarzenegger' query produces the following three most probable contexts that were hand-labeled from the automatically generated keyword list:

- 'California Politics' $p(k|d^*) = 0.38$
- 'Mexico border' $p(k|d^*) = 0.32$
- 'Politics' $p(k|d^*) = 0.17$

Illustrating that the system indeed is able to find associate relevant topics from broadcasts in the database, consisting of data from the summer of 2006.

Traditional text indexing would return documents containing the exact term 'schwarzenegger'. This can be sufficient but using imperfect transcriptions might mean that relevant sound clips are ignored. The method presented here overcomes this problem by expanding the query onto the 'topic space', given by equation (5.8). That is, documents with the highest relevance conditioned on the k topics are returned. This is useful when errors occur in automatic transcription. This is indeed the case for the 'schwarzenegger' query, where two relevant documents include wrongly transcribed versions of the word, as can be seen in figure 5.2. So the method compensates for transcription errors when the objective is to retrieve relevant spoken documents. These documents would have been missed by a conventional search.

5.4.2 Text-based document segmentation

To perform a more quantitative evaluation eq. (5.5) is used to calculate the posterior probabilities for each context given short query strings d^*. This can be used to segment a news cast into homogenous topic segments. In particular we treat a short sequence of ten words as a query string, and associate topics to these queries based on the probability $p(k|d^*)$. 'Sliding' d^* along the news cast, topic changes can be found when $\arg\max_k p(k|d^*)$ changes.

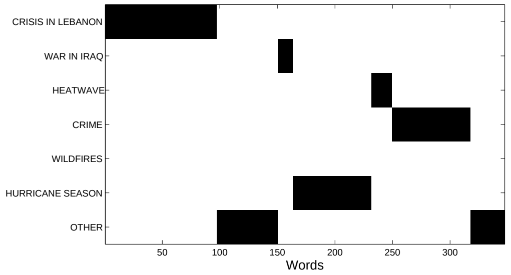
For evaluating the segmentation task we use the recall (RCL) and precision (PRC) measures defined as:

$$\begin{aligned} \mathrm{RCL} &= \frac{\textit{no. of correctly found change-points}}{\textit{no. of manually found change-points}} && (5.10) \\ \mathrm{PRC} &= \frac{\textit{no. of correctly found change-points}}{\textit{no. of estimated change-points}}, && (5.11) \end{aligned}$$

where an estimated change-point is defined as correct if a manually found change-point is located within ±5 words.

In the test we concatenated the eight manually segmented news shows and removed stop-words. Running the topic detection algorithm resulted in a recall of 0.88 with a precision of 0.44. It shows that almost every manually found change-point is found by our algorithm. On the other hand the method produces a number of false alarms. This is mostly because some of the manually found segments are quite long and include subsegments. For instance some of the segments about 'crisis in Lebanon' contain segments where the speaker mentions the US Secretary of State Condoleezza Rice's relationship to the crisis. These subsegments have a larger probability with other 'US politic' contexts, so the system will infer a change-point, i.e., infer an off-topic association induced by a single prominent name. If such events are unwanted, we probably need to go beyond mere probabilistic arguments, hence, invoke a 'loss' matrix penalizing certain association types. Figure 5.3 shows an example of the segmentation process for one of the news shows. Figure 5.3(a) shows the manual segmentation, while figure 5.3(b) shows the $p(k|d^*)$ distribution forming the basis of figure 5.3(c). The NMF-segmentation is consistent with the manual segmentation, with exceptions, such as a segment which is manually segmented as 'crime' but missed by the NMF-segmentation.

The topic-based segmentation could be used as a second pass way to perform segmentation of the text streams produced by the ASR system. Fusing the audio-based segmentation with the topic-based segmentation as a false-alarm detector, could produce a very efficient segmentation.



(a) Manual segmentation.

(b) $p(k|d^*)$ for each context. Black means high probability.

(c) The segmentation based on $p(k|d^*)$.

Figure 5.3: Figure 5.3(a) shows the manual segmentation of the news show into 7 classes. Figure 5.3(b) shows the distribution $p(k|d^*)$ used to do the actual segmentation shown in figure 5.3(c). The NMF-segmentation is in general consistent with the manual segmentation. The segment manually segmented as 'crime' is erroneously labeled 'other' by the NMF-segmentation

	c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**
c1	**370**	32	10	0	8	0	380
c2	0	**131**	1	2	0	8	52
c3	0	0	**105**	7	0	8	88
c4	0	16	2	**9**	0	0	112
c5	0	0	0	0	**13**	0	0
c6	0	0	29	0	0	**88**	0
c7	3	29	8	0	6	0	**759**

Table 5.2: Classification confusion matrix, where rows are manual labels and columns are estimated labels. The used classes are: (c1) crisis in Lebanon, (c2) war in Iraq, (c3) heatwave, (c4) crime, (c5) wildfires, (c6) hurricane season, and (c7) other.

5.4.3 Topic classification

The segmentation procedure described above provides labels for all instances of d^*. As in [115, 100] we use the accuracy (AC), defined as

$$\text{AC} = \frac{1}{n}\sum_{i=1}^{n}\delta\big(c_m(i), c_s(i)\big) \tag{5.12}$$

to quantitatively evaluate the classification performance. $\delta(x, y)$ is 1 if $x = y$, 0 otherwise. c_m and c_s denotes the manually and system labels respectively and n is the number of test strings. Using the same data as in the segmentation task we achieve an overall AC of 0.65.

The confusion matrix for the experiment is shown in table 5.2. The table shows that most of the errors occur when the system is classifying c1, c2, c3, and c4 as c7 ('other'). The system outputs the class 'other' when none of the six selected topic classes has the highest relevance $p(k|d^*)$. In the above classification the query string d^* is based on a sequence of ten words. If instead we use the 60 manually found segments as query string we are able to detect 53 correctly, which gives an accuracy of AC = 0.88.

5.4.4 Known-item retrieval evaluation

Anecdotal evaluation of the topic-based query expansion shown in section 5.4.1, suggests that the retrieval of erroneously transcribed speech is aided by the semantic topic model. This section gives an evaluation of the method in a more controlled experiment that mimics some of the circumstances of erroneously transcribed speech.

The experimental setup in this case uses a corpus of text retrieved from Wikipedia, containing articles describing musical artists. The data retrieval is described in further detail in section 7.2, but the data could be any kind of corpus, so it was just chosen for the easy availability of the data. There are two main differences between the Wikipedia dataset and the CNN news broadcast, namely that the CNN news has a large redundancy of content in the documents, every news event is mentioned in many different documents. The other is the length of documents, as the CNN corpus contains quite a few very short documents. The redundancy in the CNN data might be one of the reasons that our context based method works. A news story is usually repeated in numerous newscasts over successive hours. Despite these differences the experiments give an indication on the reason of our assumption, that the context detection makes retrieval more robust to the noise inferred by transcription errors.

The basic idea of these tests is that the speech documents contain some true words that are pronounced by the speakers. A correct result to a query should thus return the document which contained the words uttered in the recording, i.e. the correspondence meaning of the speech, ref. 2.1. The transcription errors can therefore be seen as a kind of noise which should be removed by the retrieval method.

Error model

The process that generates transcription errors is hard to model, in a simple way. The functionality of the speech recogniser both depends on acoustic and language modelling, which means that strange pronunciations as well as out of vocabulary words can infer errors. We have therefore settled on the following simple model to simulate the effects of erroneous transcription. Given the $n \times m$ termdoc matrix $\mathbf{X}_{td}$, we infer errors doing the following:

- Choose a word perturbation rate e
- For each document $X_{\cdot d}$ choose $e \cdot m$ terms to be perturbed
- Exchange each of the $e \cdot m$ terms with a another word not present in the document

Using this procedure we generated a number of different termdoc matrices, using different settings of the perturbation rate.

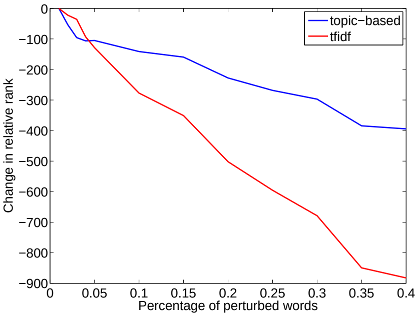


Figure 5.4: Target document retrieval test for TF-IDF and topic-based document retrieval. The performance is given as mean difference between the query rank of a target document using clean data and using the perturbed data

Evaluation

The experiment we perform here simulates a user performing a known-item search, also referred to as the target testing paradigm of information retrieval. The paradigm assumes that one specific document is the correct result for a query. It may be a somewhat simple approach to evaluate retrieval results, but does give an idea of how well user tests will fare [108].

The advantage of using target testing is that we can evaluate performance without users, as we know which document that should be returned to a given query. Target testing does mean that we may generate queries that do not necessarily simulate "real" user queries, but it will suffice for this simple setup, where the error model is also quite simple.

The queries for our tests are generated by randomly selecting target documents from the corpus. The user query is then formed as a chosen number of terms from the target document. This can be performed in a variety of ways (cf. [108]). In this setup we will choose queries that simulate the 'Schwarzenegger' example mentioned above. The query will be chosen to contain at least one word that has been erroneously transcribed plus a number of additional random words contained in the target document.

We evaluated the performance of 500 queries containing 2 words using both cosine similarity with TF-IDF weighted terms, and topic-based queries using a model with 100 latent topics. The focus of these experiments is to see how the models behave when the number of transcription errors grows. Therefore we picked out the queries that returned the target documents at a rank better than 20 using the clean text data, i.e. if the documents were returned among the 20 results closest to the query. We find that 20 documents is a reasonable number of results to show to the user at one time in a single screen, and this therefore resembles what could be realised in a real retrieval system. The queries were then repeated for the perturbed datasets registering the rank of the target documents. Figure 5.4, shows the mean difference between the query rank of a target document using clean data and using the perturbed data. As expected the performance deteriorates rapidly for both methods when more transcription errors are inferred, but the topic based method is clearly more resistant to the errors. At 20% perturbed words, which is a transcription performance only reached for clearly pronounced speech in good recording conditions, we see that the topic based model clearly outperforms the basic tf-idf method. The topic-based retrieval should therefore help retrieval of this type of data.

5.5 Web-based demo

The topic based retrieval of spoken documents was implemented as a Web-based search engine available at `http://castsearch.imm.dtu.dk`.

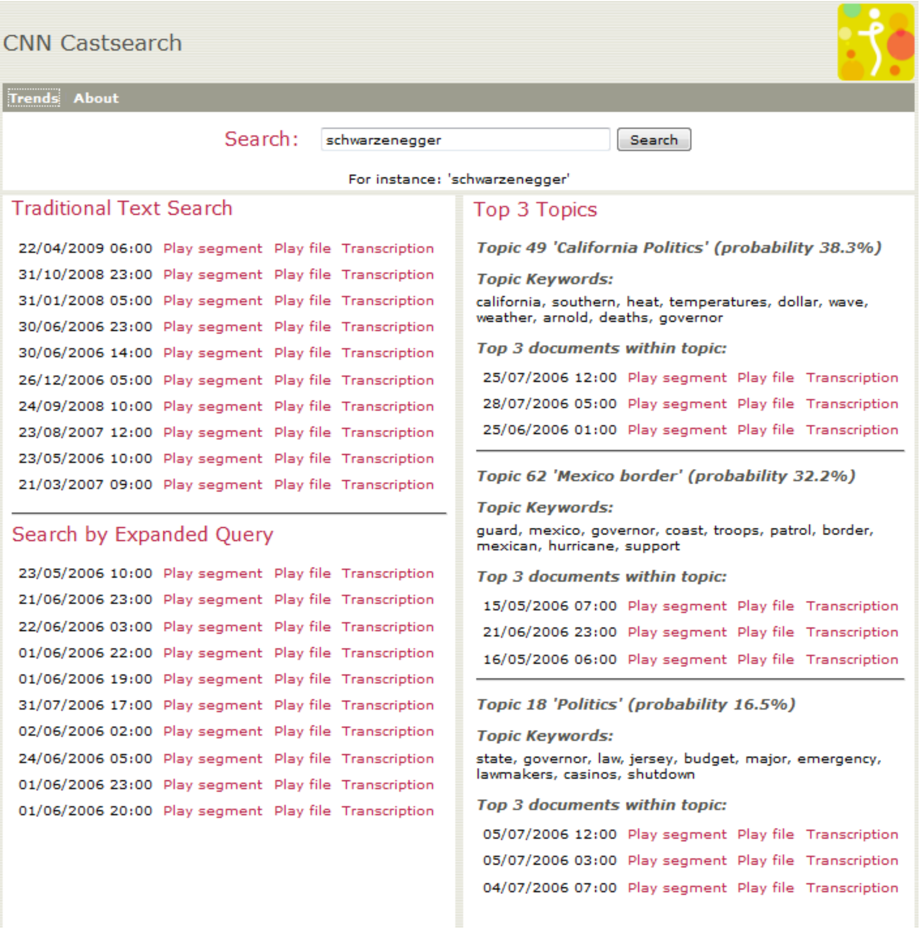
The Castsearch engine was built to index the hourly 'CNN News Update'-podcasts[3]. The mp3 podcasts have been downloaded continuously since the 10th of May 2006 with some shorter interruptions. The collection of mp3's now comprises 5̃5,900 files[4], which corresponds to 3340 hours of audio as each podcast is about 4 minutes long. The mp3's have been preprocessed using the pipeline described in section 5.2. The pipeline has produced a collection of documents that now contains:

- 510,000 documents
- 16,400,000 word tokens
- 57,379 distinct words, i.e. the dictionary.

The dictionary of the extracted documents is limited by the speech recogniser, which has a vocabulary of 64,000 words, which does not seem to be fully utilised yet.

[3] The podcasts are retrieved from this link: `http://rss.cnn.com/services/podcasting/newscast/rss.xml`

[4] The statistics reflect the state at the end of June 2009



Search: schwarzenegger

For instance: 'schwarzenegger'

Traditional Text Search

22/04/2009 06:00	Play segment	Play file	Transcription
31/10/2008 23:00	Play segment	Play file	Transcription
31/01/2008 05:00	Play segment	Play file	Transcription
30/06/2006 23:00	Play segment	Play file	Transcription
30/06/2006 14:00	Play segment	Play file	Transcription
26/12/2006 05:00	Play segment	Play file	Transcription
24/09/2008 10:00	Play segment	Play file	Transcription
23/08/2007 12:00	Play segment	Play file	Transcription
23/05/2006 10:00	Play segment	Play file	Transcription
21/03/2007 09:00	Play segment	Play file	Transcription

Search by Expanded Query

23/05/2006 10:00	Play segment	Play file	Transcription
21/06/2006 23:00	Play segment	Play file	Transcription
22/06/2006 03:00	Play segment	Play file	Transcription
01/06/2006 22:00	Play segment	Play file	Transcription
01/06/2006 19:00	Play segment	Play file	Transcription
31/07/2006 17:00	Play segment	Play file	Transcription
02/06/2006 02:00	Play segment	Play file	Transcription
24/06/2006 05:00	Play segment	Play file	Transcription
01/06/2006 23:00	Play segment	Play file	Transcription
01/06/2006 20:00	Play segment	Play file	Transcription

Top 3 Topics

Topic 49 'California Politics' (probability 38.3%)

Topic Keywords:

california, southern, heat, temperatures, dollar, wave, weather, arnold, deaths, governor

Top 3 documents within topic:

25/07/2006 12:00	Play segment	Play file	Transcription
28/07/2006 05:00	Play segment	Play file	Transcription
25/06/2006 01:00	Play segment	Play file	Transcription

Topic 62 'Mexico border' (probability 32.2%)

Topic Keywords:

guard, mexico, governor, coast, troops, patrol, border, mexican, hurricane, support

Top 3 documents within topic:

15/05/2006 07:00	Play segment	Play file	Transcription
21/06/2006 23:00	Play segment	Play file	Transcription
16/05/2006 06:00	Play segment	Play file	Transcription

Topic 18 'Politics' (probability 16.5%)

Topic Keywords:

state, governor, law, jersey, budget, major, emergency, lawmakers, casinos, shutdown

Top 3 documents within topic:

05/07/2006 12:00	Play segment	Play file	Transcription
05/07/2006 03:00	Play segment	Play file	Transcription
04/07/2006 07:00	Play segment	Play file	Transcription

Figure 5.5: Screenshot of the Castsearch demo.

The documents were then indexed using the vector space representation, using the following settings.

- Remove stopwords using a 648 word stop list, containing standard stopwords plus some words found to be non-informative in the news domain.
- Remove documents with 2 words or less. These short documents can occur because of the automatic sound segmentation
- No stemming

The resulting index of the documents is used as the basis of two different search methods. The index is stored in a MySQL-database, which we use to perform traditional search which is based on retrieval of documents that have all occurrences of the query terms. These results are presented in the upper left corner of the results window.

The topic-based search is presented in the right column and the lower left box. The topics are based on a PLSA model of 136,144 documents from April 4th 2006 until April 3rd 2007. Investigating the keywords extracted from a number of PLSA models with different numbers of components we settled on 70 components for this implementation of the demonstration. The number of topics could be chosen higher based on the large number of documents, but we lacked a more quantitative measure of performance to qualify the chosen number of components and thus settled on the 70 topics.

The topics are presented in the right column, using the keywords that the NMF model extracts. As the demo only uses this limited number of topics, we inspected the keywords and assigned a ahort title to each topic, for ease of interpretation. The final retrieval results using the NMF query expansion are presented in the lower part of the left column. The demo thus quite intuitively shows what kind of context is used to retrieve documents. Figure 5.5 shows the result of a query for 'schwarzenegger', producing the topics 'California Politics', 'Mexico Border', and 'Politics', describing the documents relevant for Governor Arnold Schwarzenegger.

5.6 Discussion

The chapter presented a system capable of retrieving relevant segments of audio broadcast news. The system uses audio processing to segment the audio stream into speech segments. A speech-to-text system is utilized to generate transcriptions of the speech. Furthermore a strategy for application of non-negative matrix factorization of joint probability tables allows us to retrieve relevant spoken documents in a broadcast news database. We have demonstrated that the system retrieves relevant spoken documents even though the query terms have been transcribed wrongly, hence showing that global topic models can assist the interpretation and thereby finding the correspondence meaning of speech-to-text data.

The latent topics will of course rely on the data that is used to train the model. The news data is clearly quite non-stationary, such that the topics extracted back in 2007 may not be very useful in the context of the presidential election in 2008. 'Barack' 'Obama' are thus quite unknown words in the topics of our demo. This of course means that topics should be extracted again at different intervals, but also that topic modelling could benefit from temporal information.

Chapter 6

Wikipedia music information retrieval

Wikipedia has significant potential as a resource for music information retrieval. It contains a very large body of music related articles and with its rich interlinked structure it is an obvious candidate as a musical 'common sense' database. This chapter investigates both the textual data as well as the linking structure of Wikipedia for MIR.

6.1 Introduction

Wikipedia's popularity has steadily grown and is an important source for knowledge for large groups of users. Wikipedia is collaboratively written by a large open community of users, currently contains almost 3,000,000 articles in the English version, and has been one of the 10 most visited Web sites in recent years. Given the popularity and wide use of Wikipedia many researchers have discussed the quality and trust of Wikipedia articles [114], and [41]. Wikipedia articles are iteratively improved through collaborative content editing, however, editors may have many different objectives, different frames of reference and different writing styles. These problems are handled by ongoing debates in the discussion pages, and further by the formulation of style manuals that provide general guidelines for the users.

Wikipedia resembles the concept of folksonomies in the way that there is more control on the context of the data compared to Web data. As Wikipedia has templates for identifying musical artists it is possible to control the data that is retrieved. In contrast to general Web mining, Wikipedia also supplies a rich hypertext representation of the musical descriptions. The concept of hypertext was defined by Ted Nelson in 1965[1] and has been viewed as a step towards realising Vannevar Bush' 1945 vision of the Memex[2]. Memex is short for 'memory extender' and the link structure in hypertext is the equivalent of extended associative memory. For quantitative analysis of human associative memory see e.g. [75] Just like personal associative memory has its own obscure dynamics, what is linked to in hypertexts can often be surprising or at times even unwanted. The use of Wikipedia with its rich hypertext structure may thus resemble musical understanding by humans. The associations built through the link structure was for instance shown to be useful for establishing semantic relationships between categories [32].

While the problems of distributed authorship mentioned above leads to somewhat heterogeneous content the textual parts of Wikipedia, the link structure may be even more problematic. Links in Wikipedia are different from links in the Web at large. Wikipedia articles often and intendedly present 'encyclopedic' links, i.e., links to other Wikipedia articles that describe a given concept. Also, links in Wikipedia can reflect non-semantic associations, say, in a biography where we find a link to another article which contains a list of un-related people who lived in a given city. This liberal use of links has produced a Wikipedia which some find to be overlinked. This self-referential example from Wikipedia's guidelines shows the problem: "...An article may be overlinked if any of the following is true:...", in this sentence the word overlinked has a link to the statement: "Overlinking in a Web page or another hyperlinked text is the characteristic of having too many hyperlinks". This link (which may be a joke?) hardly improves readability. So we conclude that even core editors do not quite live up to the general recommendations in the Wikipedia style guide: *Make links only where they are relevant to the context: It is not useful and can be very distracting to mark all possible words as hyperlinks. Links should add to the user's experience; they should not detract from it by making the article harder to read. A high density of links can draw attention away from the high-value links that you would like your readers to follow up. Redundant links clutter the page and make future maintenance harder. A link is the equivalent of a footnote in a print medium. Imagine if every second word in an encyclopedia article were followed by "(see: ...)". Hence, links should not be so numerous as to make the*

[1] http://hyperland.com/mlawLeast.html

[2] http://www.theatlantic.com/doc/194507/bush

article harder to read.[3]

The fact that Wikipedia links are different from links on the Web means that link analysis schemes that have been useful for the Web such as PageRank [22] and Kleinberg's HITS [54] may not be equally useful for Wikipedia, see e.g., [15], for an analysis. PageRank and HITS are both based on the assumption that a link represents an 'endorsement' hence, a high number of in-links is a sign of quality. Clearly, neither encyclopedic links nor links to lists are endorsements. On the other hand, the notion of endorsement is implicitly present in Wikipedia as well, as articles with few links from other articles are marked as 'orphans'. Finally, we note that links in Wikipedia can also reflect malicious 'product placement' behavior. An editor may want to enhance the visibility and credibility of a given article by making links to or from other more authoritative articles. The strong focus on internal links in Wikipedia also means that computing PageRank for a set of documents containing both Wikipedia and external pages, the Wikipedia articles will automatically obtain high scores because of the overlinking.

The work presented in this chapter is an attempt at understanding the properties of links in Wikipedia, observed through the limited set of articles describing musical artists. The aim is partly descriptive, partly constructive. We first describe the Wikipedia data, that we use as the basis of our analysis. We describe some of the basic properties of the Wikipedia articles, and go on to describe some measures from network analysis which we will use to investigate the structure of the links in Wikipedia. This is the basis of understanding how meaningful the semantics of the links are.

The following part investigates the semantic relationships between articles based on a PLSA model, to quantify the semantic relations. The PLSA method is first evaluated quantitatively on an artist similarity dataset. Secondly we evaluate the linking structure using the semantic similarity. The main idea is that links should be more likely between articles with similar content. Other works [34, 82] investigate hypertexts by modelling the text in combination with the linking structure but we do not explicitly incorporate links in the model on account of the many different facets of links in Wikipedia.

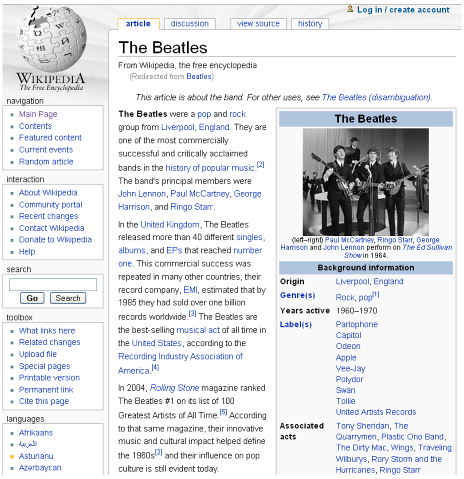
Furthermore we investigate how a directed stream of consciousness, i.e., random walk in the link structures may be used for generating playlists. The use of the semantic similarity can be used to guide the playlisting.

[3] `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style`

6.2 Wikipedia Data Set

One of the compelling reasons for using Wikipedia data is that the whole encyclopedia including articles, edit history, and discussion pages is freely available. The article data from Wikipedia is available as an XML-based corpus downloadable at 'download.wikimedia.org'. The XML-file used in this work contains a dump of the database containing all the user-readable articles of Wikipedia as of March 2nd 2008. These include ordinary articles on topics, but the corpus also includes auxiliary pages such as the list-like pages that have links to articles related to a date, country or other general topics.

The main advantage of using Wikipedia instead of Web-mined text is that the articles in the XML-file only contain the actual content describing the artists, without the menus, commercials and other irrelevant text, that inevitably would be present if we were to crawl the general Web for data. One aspect of Web mined data, that is not mimicked by Wikipedia are the many subjective views on music, for instance found in blogs, as well as descriptions of all the bands existing. In the following section we will therefore do a brief investigation on the coverage of the artists described in Wikipedia.



The Beatles

From Wikipedia, the free encyclopedia
(Redirected from Beatles)

This article is about the band. For other uses, see The Beatles (disambiguation).

The Beatles were a pop and rock group from Liverpool, England. They are one of the most commercially successful and critically acclaimed bands in the history of popular music.[2] The band's principal members were John Lennon, Paul McCartney, George Harrison, and Ringo Starr.

In the United Kingdom, The Beatles released more than 40 different singles, albums, and EPs that reached number one. This commercial success was repeated in many other countries; their record company, EMI, estimated that by 1985 they had sold over one billion records worldwide.[3] The Beatles are the best-selling musical act of all time in the United States, according to the Recording Industry Association of America.[4]

In 2004, *Rolling Stone* magazine ranked The Beatles #1 on its list of 100 Greatest Artists of All Time.[5] According to that same magazine, their innovative music and cultural impact helped define the 1960s[2] and their influence on pop culture is still evident today.

The Beatles

(left–right) Paul McCartney, Ringo Starr, George Harrison and John Lennon perform on *The Ed Sullivan Show* in 1964.

Background information

Origin	Liverpool, England
Genre(s)	Rock, pop[1]
Years active	1960–1970
Label(s)	Parlophone Capitol Odeon Apple Vee-Jay Polydor Swan Tollie United Artists Records
Associated acts	Tony Sheridan, The Quarrymen, Plastic Ono Band, The Dirty Mac, Wings, Traveling Wilburys, Rory Storm and the Hurricanes, Ringo Starr

Figure 6.1: Typical Wikipedia article, with an infobox containing summarized data about the band.

Retrieving all pages that are reachable starting from the *Music*-category produce a massive 150,000 pages, which includes all kinds of related pages describing music museums, musical instruments, and composers. Pruning this collection to only contain musical artists is thus a problem. Luckily, the Wikipedia-entries are written in a simple (and easily parsed) markup-language that defines simple templates for headers, itemised lists, and images. Specific topics often have special templates that are used to provide info-boxes to present structured data, as can be seen on the right in figure 6.1. We wrote scripts to match the templates for info-boxes for musical artists, producing an initial collection of ∼ 35,200 articles.

The initial set of articles contained a number of very short documents. Such very short and improvable articles are typically tagged as 'stubs' by the Wikipedia editors, to signal that the pages still need substantial improvement. Articles tagged as stubs were therefore removed from our data set. The data was further pruned by removing the terms that occurred in two documents or less, and finally documents with fewer than 30 words were removed.

Artist similarity experiments using Web-based data [56] have shown that these methods rely very heavily on the occurrence of artist and album names in the Web pages. Our efforts here are to try to retrieve the semantics of music understanding, which we believe is not very well-reflected if we just measure artist co-occurrence. We therefore remove all words that are used in titles of articles, e.g. all occurrences of the terms 'The' 'Rolling' 'Stones' are removed. One twist to this approach is that there are band names, such as 'Rock Hard Power Spray'[4], that contain many of the musically meaningful words. Therefore we employ a threshold so that words that are common in the database are retained.

The preprocessing steps resulted in a set of 30,172 articles with 34,548 unique words, containing 4,893,015 words in total.

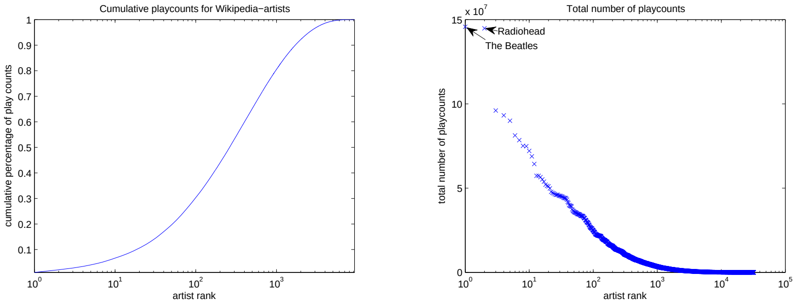
The Wikipedia articles form a very densely interlinked structure, which we want to investigate. Retrieving all the outlinks from the 30,172 articles produced 175,141 unidirectional links.

6.2.1 Wikipedia data coverage

Generally social media sources may lack coverage of less popular items, Wikipedia editors may not be interested in describing more obscure artists, so that only the biggest and most well-known artists are in the data set. To quantify what kind of artists are described in Wikipedia, we compare the extracted artists with the last.fm service described earlier. We found 35,597 artists from our set in the last.fm database. The last.fm system measures how often the different artists are played by the users, and using the API[5] they provide we can quan-

[4] A Danish hard rock band

[5] `http://www.last.fm/api`



(a) (b)

Figure 6.2: Log-linear plot showing the artist rank in terms of total playcounts

tify how popular the artists in our dataset are. We extracted the playcounts for the artists, as shown in figure 6.2(b). We can see that the Wikipedia articles span the most popular artists, such as The Beatles and Radiohead that have had tracks played $\sim$ 150 million times, to the least popular ones with only single plays. Celma and Cano [30] performed an extensive analysis of the last.fm recommendation networks, showing that the artist popularity forms a 'long tail' distribution. Figure 6.2(a) shows the cumulative playcounts, which resembles the results in [30], so our use of Wikipedia as a musical common-sense database seems to be viable, and semantics extracted from this collection should reflect the general cultural background of modern music.

6.3 Link analysis

As mentioned in section 6.1 links in Wikipedia have been investigated to provide a semantic relatedness between concepts. As we see Wikipedia, it is a growing repository of distributed and interconnected knowledge. Its hyperlinked structure and the ongoing, incremental editing process behind it make it very peculiar domain to study the impact of (social) network analysis techniques to a restricted, controlled corpus where resources are not generic Web pages, but content relevant entries edited and constantly revised by a community of committed users. We therefore apply some quantitative tools from network analysis to get an idea of how Wikipedia understands music. The following defines the basic notions.

The network of Wikipedia articles can be seen as a graph where articles form the set of vertices and links are the edges. Given the graph $G(V, E)$ we can evaluate a number of standard descriptive measures from network analysis. The links can either be seen as undirected or as directed edges, where a directed link makes it possible to put more meaning into the semantic relation of documents. A link from an unknown artist to a well-established artist may thus be less interesting than the converse.

In the following we present measures using the graph interpretation. In the following the degree of a vertex is the number of edges adjacent to the vertex. In a directed graph we can define the out-degree as the number of edges pointing from the vertex to others, and the in-degree being the number of edges pointing at the vertex.

6.3.1 Average shortest path

The average shortest path (or mean geodesic length) measures the distance between two vertices v_i and v_j. They are connected if one can go from v_i to v_j following the edges in the graph. The shortest path distance (or geodesic path) is the shortest path from v_i to v_j. Given shortest path distance between two vertices, d_{ij}, we can calculate the average shortest path of the full network:

$$\langle d \rangle = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i,j \in V, i \neq j} d_{ij} \tag{6.1}$$

In a random graph, the average path approximates to:

$$\langle d \rangle \approx \frac{\log N}{\log \langle k \rangle} \tag{6.2}$$

where N is the number of vertices and $\langle k \rangle$ denotes the mean degree of all the nodes.

The longest path in the network is called its diameter (D). In a graph we can see how easy it is to navigate from one artist to another, showing how easily a playlist can reach all artists.

6.3.2 Giant component

The strong giant component, SGC, of a network is the set of vertices that are connected via one or more paths, and are disconnected from all other vertices. Typically, networks have one large component that contains most of the vertices. If we are to use the link structure of Wikipedia for playlists it is important that all documents are reachable.

6.3.3 Connectivity

Degree distribution

The degree distribution, p_k, is the number of vertices with degree k:

$$p_k = \sum_{v \in V, deg(v)=k} 1 \tag{6.3}$$

where $deg(v)$ is the degree of the vertex v. More frequently, the cumulative degree distribution (the fraction of vertices having degree k or larger), is plotted:

$$P_c(k) = \sum_{k'=k}^{\infty} p(k') \tag{6.4}$$

A cumulative plot avoids fluctuations at the tail of the distribution and facilitates the computation of the power coefficient γ, if the network follows a power law. $P_c(k)$ is, usually plotted as the complementary cumulative distribution function (ccdf). The complementary cumulative distribution function, $F_c(x)$, is defined as:

$$F_c(x) = P(X > x) = 1 - F(x) \tag{6.5}$$

where $F(x)$ is the cumulative distribution function (cdf):

$$F(x) = P(X < x) \tag{6.6}$$

$F(x)$ can be regarded as the proportion of the population whose value is less than x. Thus, $P_c(k)$, derived from $F_c(x)$, denotes the fraction of nodes with a degree greater than or equal to k.

In a directed graph, $P(k_{in})$ and $(P(k_{out}))$, the cumulative incoming and outcoming degree distribution, are more informative. Complementary cumulative indegree distribution, $Pc(k_{in})$, detects whether a recommendation network has some nodes that act as hubs. That is, vertices that have a large amount of attached links. This clearly affects the recommendations and navigability of the network.

Also, the shape of the curve helps us to identify the topology of the network. Regular networks have a constant distribution, ”random networks” have a Poisson degree distribution [37] meaning that there are no hubs, while ”scale-free networks” follow a power-law distribution in the cumulative degree distribution [13], so there are a few hubs that control the network. Many real world networks, including the world wide Web linking structure, are known to show a right-skewed distribution forming a power law distribution

$$P(k) \sim k^{-\lambda}, \tag{6.7}$$

where $\lambda > 1$ and typically with $2 < \lambda < 3$.

6.3.4 Document authority

Web search engines such as Google rely heavily on methods from network analysis to find the pages that can be seen as the "go to" places on the Web for reliable information. The way to find the most reliable pages are found by measuring how important the pages are seen by other pages by linking to them. The pages that are deemed most important are called *authorities* and those pages that link to many authorities are called *hubs*. This setting means that the scoring of pages is reinforcing, where good hubs link to good authorities and good authorities link to good hubs.

Two basic methods are popular for network analysis, the HITS- algorithm [54] and PageRank [22], which is a part of Google's search engine.

6.3.5 HITS

The Hyperlinked Induced Topic Search (HITS) algorithm was developed to extract authorities and hubs for a set of topically related Web pages, which are for instance retrieved as the result to a regular term-based query. The algorithm tries to quantify the notion that good authorities are pointed to by good hubs and good hubs point to good authorities. The algorithm thus defines two scores for a Web page P_i, the authority score a_i and a hub score h_i. The network formed by the Web pages can be described as a graph with the set Web pages V and the set of directed links between the Web pages E. The graph is described using an adjacency matrix $\mathbf{L}$, such that L_{ij} is 1 if there is a link from page i to page j. The HITS method then calculates the hub and authority scores through the following updates:

$$a_i^{(k)} = \sum_j h_j^{(k-1)}, \text{ where } \mathbf{L}_{ji} = 1, \tag{6.8}$$

$$h_i^{(k)} = \sum_j a_j^{(k-1)}, \text{ where } \mathbf{L}_{ij} = 1, \tag{6.9}$$

The initial scores $a_i^{(0)}$ and $h_i^{(0)}$ are typically initialised to 1's. These iterations can quite easily be rewritten in matrix form, using the adjacency matrix defined above, to obtain:

$$\mathbf{a}^{(k)} = \mathbf{L}^{\top}\mathbf{h}^{(k-1)} \text{ and } \mathbf{h}^{(k)} = \mathbf{L}\mathbf{a}^{(k-1)} \tag{6.10}$$

where the $n \times 1$-dimensional vectors $\mathbf{a}^{(k)}$ and $\mathbf{h}^{(k)}$ the hub and authority scores for the n pages in the graph. The iterations were shown to converge to the authority and hub scores for some k. A few simple substitutions show that the iterations can be written as

$$\mathbf{a}^{(k)} = \mathbf{L}^{\top}\mathbf{L}\mathbf{a}^{(k-1)} \text{ and } \mathbf{h}^{(k)} = \mathbf{L}\mathbf{L}^{\top}\mathbf{h}^{(k-1)}, \tag{6.11}$$

which is essentially the power method of calculating the dominant eigenvectors of the symmetric matrices $\mathbf{L}\mathbf{L}^{\top}$ and $\mathbf{L}^{\top}\mathbf{L}$ [42]. The power method should be normalised at each iteration to guarantee convergence, so $\mathbf{a}^{(k)}$ and $\mathbf{h}^{(k)}$, could be normalised as:

$$\mathbf{a}^{(k)} = \frac{\mathbf{a}^{(k)}}{\|\mathbf{a}^{(k)}\|_1} \text{ and } \mathbf{h}^{(k)} = \frac{\mathbf{h}^{(k)}}{\|\mathbf{h}^{(k)}\|_1}. \tag{6.12}$$

6.3.6 Pagerank

The PageRank algorithm relies on the vast link structure of the Web as an indicator of an individual page's relevance metric. In the words of its authors, Pagerank interprets a link from page A to page B as a "vote" by page A for page B. However, analogously to the HITS algorithm, votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important."

The basic definition of Pagerank assigns a rank r to page P_i, using the following recursive formula:

$$r^j(P_i) = \sum_{Q \in \mathcal{T}_{P_i}} \frac{r^{j-1}(Q)}{|Q|} \text{ for } j = 1, 2, 3, \ldots \tag{6.13}$$

$\mathcal{T}_{P_i}$ is the set of pages that point to P_i, and $|Q|$ is the number of outlinks from the Web page Q. If we form the vector $\mathbf{x}_j^{\top} = [r_j(P_1), r_j(P_2), \ldots, r_j(P_n)]$, the update can be written as:

$$\mathbf{x}_j^{\top} = x_{j-1}^{\top}\mathbf{A}. \tag{6.14}$$

The matrix $\mathbf{A}$ is the $n \times n$ transition matrix of a Markov chain, given by

$$a_{ij} = \begin{cases} \frac{1}{|P_i|} & \text{if } P_i \text{ links to } P_j \\ 0 & \text{otherwise,} \end{cases} \tag{6.15}$$

where $|P_i|$ is the number of outlinks from the page P_i.

The resulting left-most eigenvector of the power iteration in 6.14 defines the Pagerank for each page in the graph. The calculation of the eigenvector is not guaranteed to converge to the stationary distribution vector for the Markov chain, so a number of adjustments to $\mathbf{A}$ have been proposed.

The Pagerank algorithm can be seen as a surfer traversing the graph randomly starting at a page and then taking any of the outlinks from that page. The Pagerank is then the number of hits a page will get from this surfer. Brin and Page [22] suggest adding a random component so that the surfer at some point ”jumps” to a random page, so that pages without inlinks can be reached. The adjustment is given as

$$p\mathbf{A} + (1-p)\mathbf{E}, \text{ where } \mathbf{E} = \frac{\mathbf{e}\mathbf{e}^\top}{n}, \tag{6.16}$$

where $(1-p)$ is the probability that a random page is chosen instead of an outlink in the random walk. p was set to 0.15 in [22] based on observations of average surfer’s usage patterns. $\mathbf{e}$ is an n-vector of 1’s.

6.4 Network analysis of musical Wikipedia

Wikipedia links exhibit a somewhat different function than hyperlinks in the Web. The links thus represent more cognitive relations between articles than normal Web pages. The relations can be described as belonging to one of the following classes:

Describing links These are links that are used to explain a word. These could typically be the links in a dictionary, where links point to another article that defines the word.

Lexical links/Semantic links If two articles contain the same information or describe related concepts, they have a semantic link. This kind of link could be a direct link, but even more prominently Wikipedia uses categories to make these associations.

The semantic links are the most useful in the context of music information retrieval, as we want to be able to infer similarities between artists. First we perform a high-level general description of the network structure, before we inspect the semantics of the links.

	Wikipedia	Last.fm	AMG
N	30,172	122,801	74,494
Avg. shortest path	4.8	5.64	5.92
Diameter	18	10	9
SGC	63%	99.53%	95.80%
λ	2.7	2.3	exp.

Table 6.1: Network properties for the Wikipedia link structure, and the collaborative filtering recommendation network of last.fm and recommendations in the expert-based All Music Guide presented in [29]. N is the number of nodes in the network and the other, and lambda is the power-law exponent of the cumulative indegree distribution.

The links we analyse in this section are the direct links between artists, described in section 6.2. As described above natural networks typically form a structure where nodes that have many links are more likely to get new ones. Fitting a power-law distribution to the count of inlinks of the artist network gives a fit with a coefficient of 2.7, which means that we have a network with hubs, which means that the network will be useful for navigating artists.

To further characterise the network, we calculated some of the standard measures from social network theory. First we find the largest strongly connected component, i.e., the component that contains nodes that are all reachable starting from any node. This connected component contains 18980 nodes of the 30172 nodes in the network (63%). So if we were to use the direct links as the only way to navigate the artists, typically only 63% of the artists would be reachable.

Given the large connected component we can measure the *diameter*, which is the longest shortest path between two nodes in the network. The two nodes that are furthest apart in our network is the Australian black metal band *The Furor* and the Philippine pop artist *Ariel Rivera*, having a shortest path of 18 hops. If we instead look at what is the typical distance between two artists, i.e. measure the average shortest path distance between two nodes, we get 4.8 hops, which indicates that we have a highly interlinked hypertext, which means that we can get from one artist to the other in relatively few hops. Summarising these Performing the same analysis for the network derived from recommendations in last.fm, produces a network with an average shortest path distance of 4.8, so the two networks seem to resemble each other on an overall scale.

The analysis by Celma and Cano [29] of recommendation networks based on collaborative filtering (last.fm) and expert based recommendations (All Music

Guide[6]), provides the properties given in table 6.1. The networks in the analysis are considerably larger than our Wikipediadata, but does give an indicator of the relations between the sorts of links that are in the three different networks. The Wikipedia network resembles the last.fm -based network most, which makes sense as they are both produced in a collaborative fashion. The largest connected component in the Wikipedia network on the other hand constitutes a much smaller percentage of the nodes compared to the other two networks. The direct links included in this analysis are thus not sufficient if we for instance want to produce playlists, as all artists are not reachable. One solution could be to extend the direct links with the links that go through other documents, such that artists that have links to rock music for have a link. This procedure would certainly make the largest connected component span more nodes, but would also produce more noisy links.

6.4.1 Authoritative artists

Even though we argue that the Wikipedia structures are essentially different from endorsement-like links in the Web at large, we apply the Pagerank and HITS algorithms to the Wikipedia-graph. We calculated the hub and authority scores using HITS as well as the Pagerank score using the full link matrix described above. The Pagerank algorithm was used with random surfer parameter $p = 0.15$.

The results of the Pagerank algorithm is shown in table 6.2 and HITS authorities and hubs in 6.3 and 6.4. The artist names are given by sorting the HITS and Pagerank scores, showing the pages with the largest scores. The top ranked pages in Pagerank and the most authoritative pages in HITS correspond to some of the most well-known artists, primarily those having a long history. The method used to extract the artist articles does mean that the data only contains modern (meant as artists from the last century), so the influence of classical rock artists, such as The Beatles, Bob Dylan, and Elvis Presley, should emerge if the links are to represent human-like association between artists. The results of calculating the hub scores should produce pages that point to good authorities, which includes some of the same artists as the most authoritative, but also artists such as *Slash* who is a guitarist who has performed together with numerous other artists. Another more spurious example is Paulinho Da Costa who is a percussionist, credited as one of the artists that has had most collaborations with other artists. The article consists of a very long alphabetical list of artists who he has been collaborating with. This sort of article is clearly not very informative, and may be an attempt at skewing the results of algorithms

[6] http://allmusic.com

1. The Beatles
2. Bob Dylan
3. Elvis Presley
4. The Rolling Stones
5. David Bowie
6. Jimi Hendrix
7. John Lennon
8. Led Zeppelin
9. U2
10. Michael Jackson
11. Paul McCartney
12. Frank Sinatra
13. Pink Floyd
14. Metallica
15. Madonna
16. Stevie Wonder
17. Duke Ellington
18. Nirvana
19. Johnny Cash
20. George Harrison
21. Black Sabbath
22. The Clash
23. Queen
24. Louis Armstrong
25. Neil Young
26. Bruce Springsteen
27. Quincy Jones
28. Frank Zappa
29. Jay-Z
30. Kiss

Table 6.2: Top articles found using Pagerank algorithm on Wikipedia links.

such HITS and Pagerank.

The Wikipedia infoboxes and rich linking culture also efficiently associates articles to other concepts, which forms semantic links from artists to these concepts. One example is that we can find articles from the Wikipedia corpus, that describe musical genres, and include outlinks from artists to these genres as well as the outlinks from the genres in the link matrix. Extracting all musical genre or style related articles from Wikipedia produced 1200 articles that were included into the link matrix. The genres span from very generic ones like *rock music* to obscure styles such as *Shibuya-kei*[7].

We perform the Pagerank algorithm on the link matrix, shown in table 6.5, which gives an impression of which musical genres are most important. It is no surprise that Rock and Pop music come out as important articles. More interestingly we can see that Jazz, Country and Blues genres also are important genres, which is maybe not evident from the most popular artists, in for instance last.fm . The Pagerank algorithm thus seems to catch the importance of these genres as stylistic origins of most contemporary music.

6.4.2 Discussion

These initial investigations of the link structure shows that the important articles in the network correspond to influential artists, and also identifies the basic musical genres as the most important. So if we were to use the linking structure to make a playlist based on a random walk in the graph, we would typically visit the well-known artists such as The Beatles and Rolling Stones

[7] Shibuya-kei is a special variant of Japanese pop

1. The Beatles
2. Bob Dylan
3. Led Zeppelin
4. The Rolling Stones
5. David Bowie
6. Jimi Hendrix
7. Elvis Presley
8. John Lennon
9. U2
10. Pink Floyd
11. Paul McCartney
12. Queen
13. Stevie Wonder
14. Michael Jackson
15. Metallica
16. Black Sabbath
17. Kiss
18. Nirvana
19. Neil Young
20. Bruce Springsteen
21. Madonna
22. George Harrison
23. Guns N' Roses
24. Aretha Franklin
25. Johnny Cash
26. The Clash
27. AC/DC
28. Ozzy Osbourne
29. Elvis Costello
30. Aerosmith

Table 6.3: Top authorities found using the HITS algorithm on Wikipedia links.

1. Bob Dylan
2. Slash
3. Donovan
4. Paulinho Da Costa
5. Queen
6. Chuck Berry
7. Black Sabbath
8. Jimi Hendrix
9. Emmylou Harris
10. Def Leppard
11. Linda Ronstadt
12. Paul McCartney
13. Billy Joel
14. Bo Diddley
15. Alice Cooper
16. Jeff Beck
17. Ramones
18. Ozzy Osbourne
19. Dave Grohl
20. U2
21. Devo
22. Pure Rubbish
23. Metallica
24. The Rolling Stones
25. Elvis Costello
26. Genesis
27. Aretha Franklin
28. Stevie Wonder
29. The Archers
30. Travis

Table 6.4: Top hubs found using the HITS algorithm on Wikipedia links.

1. Jazz
2. Rock music
3. Pop music
4. Punk rock
5. Alternative rock
6. Singer-songwriter
7. Hip hop music
8. Country music
9. Blues
10. Indie rock
11. Rock and roll
12. Hard rock
13. Heavy metal music
14. Electronic music
15. Rhythm and blues
16. The Beatles
17. Hardcore punk
18. Soul music
19. Progressive rock
20. Funk

Table 6.5: Pagerank for the extended link matrix containing both artists and genres.

most often. This sort of random playlist generation may be a little too relaxed to be useful for a user, so to make a useful playlist we need to be able to steer the recommendations.

Even though the overall linking structure seems useful for MIR, it is also clear that Wikipedia editors may insert irrelevant links or even malicious links for instance to enhance the visibility of unknown artists. One example of semantically useless links is the huge collection of links in the article of Paulinho Da Costa, which just lists every artist he has worked with. These links are clearly not all equally relevant for a user interested hip hop music, as Da Costa has worked with artists spanning almost all possible modern music genres.

We therefore go on to analyse the semantic relatedness of articles using PLSA to evaluate the sanity of the links.

6.5 Latent semantic modelling

We formed the basic word-space model counting the words in each article forming a term-document matrix $\mathbf{X}$, from the data described in section 6.2, where the element $\mathbf{X}_{ij}$ of the matrix is how many times the term t_i occurs in the document d_j.

The latent semantics were captured using the PLSA mixture model

$$p(t,d) \quad = \quad \sum_{k=1}^{K} p(t|k)p(d|k)p(k). \qquad (6.17)$$

Giving us the model parameters $p(t|k), p(d|k)$, and $p(k)$, that can be used to describe the latent topics of the data. The estimation of the parameters was performed using the same NMF based method described in section 5.3.

For this work we both investigated the use of the projected gradient descent method by Lin [68], as well as the sparsity enforcing method by Shahnaz et al. [100]. We apply the sparse method to investigate if it produces more easily interpretable topics.

6.5.1 Relevance of links

The similarity of two documents d and d' of our corpus is determined using the simple projection of the documents onto the K latent topics.

$$p(d|d') = \sum_{k=1}^{K} p(d|k)p(k|d') \tag{6.18}$$

This formula is very useful for regular retrieval, but may have a bias towards "large" documents that dominate the topics.

Given the probabilistic interpretation of similarity we can evaluate which documents are interesting using the concept of model based "surprise", introduced by Itti and Baldi [51]. They measure the relevance of an observation to maximize the difference between the prior and posterior distributions of an observation. The "surprise" is measured as the Kullback-Leibler divergence between the prior and posterior distributions. This measure was shown to be a good model human attention in a television setup.

Inspired by this notion of relevance we apply a normalized measure of relevance, which should remove the bias on large documents

$$\log \frac{p(d,d')}{p(d)p(d')} = \log \frac{p(d|d')}{p(d)}, \tag{6.19}$$

where $p(d)$ is the prior probability of the document d estimated by the model, i.e.

$$p(d) = \sum_{t} \hat{\mathbf{X}}_{td}^{NMF}, \tag{6.20}$$

where $\hat{\mathbf{X}}_{td}^{NMF} = \mathbf{WH}$ is the count table estimated by the NMF procedure.

6.6 PLSA model evaluation

This section investigates the PLSA models trained on the Wikipedia corpus. The general question is what kind of information is modelled, i.e. which kinds of topics are extracted, and how does the reduced representation perform for artist similarity calculations.

The evaluation of PLSA models is often based on datasets with a very clear cluster structure such as the *20Newsgroups*[8], *WebKB*[9], or the *Reuters*[66] cor-

[8] http://www.cs.cmu.edu/ TextLearning/datasets.html

[9] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html

pus. These collections have clear assignment of documents to categories, which makes systematic evaluation straightforward. Within the music domain there is no ground truth which can be agreed upon. As mentioned earlier genre labels have been argued to be difficult to use as a basis, because genre assignments may be skewed by commercial considerations, e.g. assigning the genre rock to an artist to make it visible for consumers[85].

Despite the general question of the usefulness of genre, we will evaluate our models on a standard dataset for artist similarity, and investigate the latent semantics extracted by the model.

6.6.1 Artists similarity evaluation data

To allow comparison with previous work, we test the PLSA-model similarity measure to replicate the experimental set-up used in the series of influential papers following [55], in which artist-artist similarities were calculated for a set of 224 artists split equally over 14 mainstream genres: country, folk, jazz, blues, RnB/soul, heavy metal, alternative/indie, punk, rap/hiphop, electro, reggae, classical, rock n roll and pop. Unfortunately our subset of Wikipedia articles did not include all of the artists in this test set. The number of artists in each genre is shown in table 6.6, and the actual artists are listed in appendix A. The construction of the text corpus for our PLSA model was based on modern artists, so the classical composers included in the original ground truth dataset, are excluded from the following evaluations.

We use the artist similarity test to compare parameter settings for the PLSA models. Training of the PLSA model was performed by omitting the artists from the test set, to see if we can find latent semantics that are generally useful as basic descriptors for musical knowledge. We trained models using the NMF GD-CLS method, varying the number of latent dimensions and level of sparsity used in the model. The performance is measured as the precision at rank 2, analogously with the method employed in [55].

6.6.2 Performance measures

Knees et al. [55] measure the performance of the genre-based artist similarity as a Leave One Out 1-nearest neighbour classification rate (LOO-1), i.e. measuring the classification performance of an artist query using the rest of the data as training set. The LOO-1 classifications can also be used to construct a *confusion matrix*, which compactly shows the structure of the classification

Genre	No. of artists
Country	16
Folk	14
Jazz	12
Blues	14
RnB/Soul	16
Heavy	16
Alternative/Indie	15
Punk	15
Rap/Hip hop	15
Electro	13
Reggae	12
Classical	1
Rock ’n Roll	13
Pop	11

Table 6.6: The 14 mainstream genres that act as ground truth data for judging artist similarity. The second column shows the number of the 16 possible artists that are present in the Wikipedia data.

errors. Confusion matrices are usually normalised, such that all numbers in one row sum to 1. Each column thereby contains the probability of estimating a given class for all the true classes.

Another more standard IR performance metric is Average Precision, which is calculated as

$$AP = \frac{\sum_{r=1}^{N} Pr(r)rel(r)}{R}, \tag{6.21}$$

where $Pr(r)$ is the precision at rank r, i.e. the number correctly retrieved artists among the first r entries. $rel(r)$ is 1 if the document at rank r is relevant (i.e. is labelled with the same genre as the query on our experiments) and 0 otherwise. R is the total number of relevant documents, i.e. the number of documents in the genre from which the query artist is from, and N is the total number of documents in the collection. AP therefore measures the average precision over the ranks at which each relevant artist is retrieved. Besides being a standard IR performance metric (which has become consensual in literature in the field of image retrieval), AP rewards the retrieval of relevant artists ahead of irrelevant ones, and is consequently a good indicator of how the semantic space is organised by each model. Calculating the AP for each artist query, gives us the mean AP (mAP).

6.6.3 PLSA -based artist similarity genre-classification performance

We set up a number experiments to test the influence of parameter settings for the latent topic models. We trained NMF models using the projected gradient approach, as well as the GD-CLS model which imposes sparsity on the found topics.

The tests were performed by splitting the data set into a test set consisting of the 186 artists from the 13 genres in table 6.6, using the remaining 31,266 artist documents to train the NMF models.

As a baseline we tested the performance of a basic vector space model using TF-IDF weighting, evaluating the similarity of artists using normalised cosine similarity. Training the model consists basically consists of calculating the idf-weights, to be used for the query documents. The performance measured using the LOO in terms of precision was 66.5%, and the mAP score was 0.441. The corresponding confusion matrix is shown in figure 6.3. It is evident that artists in some genres are more closely related within the genres than others. Jazz and Heavy Metal seem very closely related, while the alternative/indie genre seems somewhat more difficult to distinguish from Heavy Metal and Folk music. There also seems to be a tendency that all genres have some artists that are deemed similar to the Folk genre. The performance of our method is somewhat lower than the performance reported in [55], where TF-IDF features extracted from Web-mined text gave a precision of 87% for the LOO setup using the same set of test artists.

We trained a number of NMF models using the projected gradient algorithm be Lin[67]. We also investigate the use of sparsity using the algorithm by Shahnaz et al. [100], described in section 4.5.3. The LSA literature suggest the use of 100-600 dimensions to get reasonable semantic topics. This corpus is somewhat limited so we investigate models for using a range of latent components K from 8 to 300. The training of the models seemed to converge when the updates did not change the objective more than 10^{-9}. We did not employ any methods like "early stopping"[49], even though it may help to avoid overfitting.

The results of using the projected gradient method is shown in figure6.4. The reduced representation performs somewhat worse compared to the basic TF-IDF approach in this task, but does come quite close with 0.040 in the mAP-measure. The performance for different numbers of components are shown in figure 6.4, which shows that models with less than 90 topics seem to perform significantly worse than the higher dimensional models. The performance after 100 topics seems to level out, suggesting that the hundred topics may be appropriate for

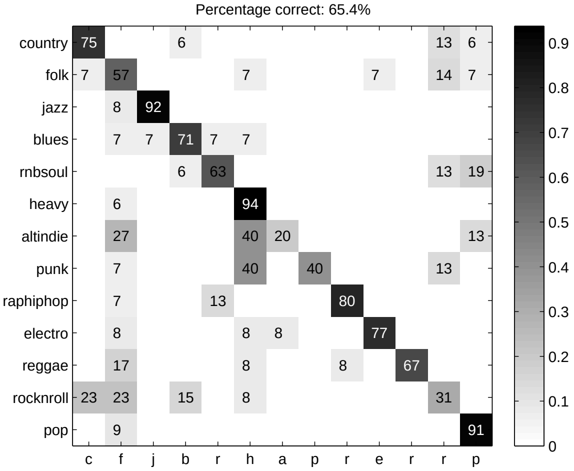


Figure 6.3: Confusion matrix for vector space model using TF-IDF weighting. Each cell shows the precision in predicting an artist in the same genre as the query. The is row normalised

artist similarity. Quite interestingly the 100 topics corresponds quite well with the observations in [65], where a latent semantic representation of musical tags suggests around 90 latent topics to suffice.

Building a confusion matrix, shown in figure 6.5 of the LOO-1 nearest neighbor setup, shows an interesting property of the latent semantics based method, as the errors seem to be more easily explainable. The main confusion is between the genres is between jazz, blues, and R'n'B/soul as well as a clear confusion between the genres heavy metal, punk, and alternative/indie.

The influence of sparsity

We also investigate the influence of using regularisation to impose sparseness on the document loadings. This may produce more clear assignment of documents to topics, and may thus be more easily interpretable. We therefore performed a small investigation of the impact of sparsity on the performance of the latent topic modelling. The training using the sparsity parameter unfortunately makes the estimation of parameters less robust, as the objective gets more local minima, that the optimisation may get stuck in. We did test the GD-CLS model using different settings of the regularisation parameter λ. The figure 6.6 shows

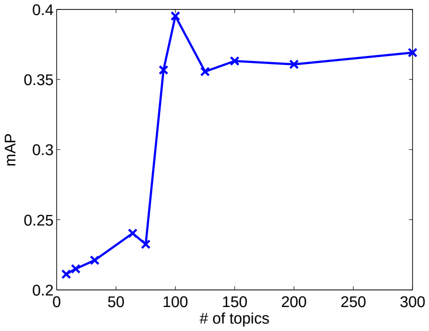


Figure 6.4: Genre based artist similarity performance for varying numbers of topics. Performance is measured using the mean Average Precision described in section 6.6.2

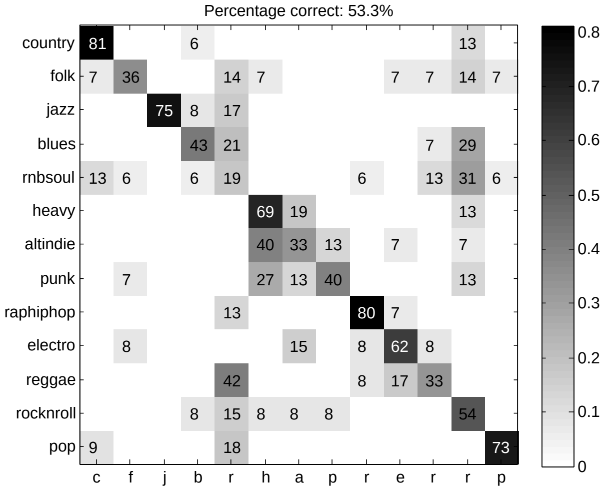


Figure 6.5: Confusion matrix for vector space model using the projected gradient based latent semantics. Each cell shows the precision in predicting an artist in the same genre as the query. The is row normalised

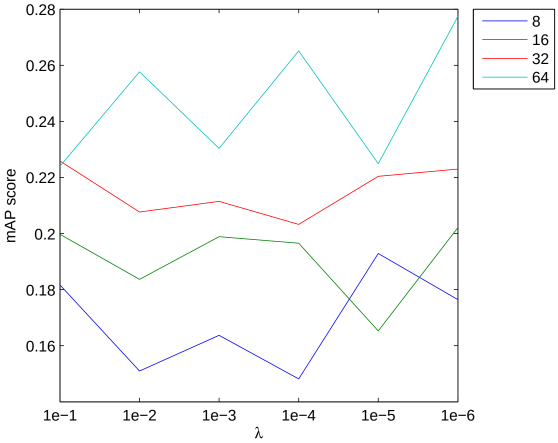


Figure 6.6: mAP performance using various settings for the sparsity parameter λ in GD-CLS. The performance for different numbers of latent components are shown.

the mAP-score for experiments with the GD-CLS algorithm using 8, 16, 32, and 64 components. The figure also shows the mAP performance for the projected gradient method. It is clear that the higher levels of sparsity do not improve performance, as a very low $\lambda = 10^{-6}$ sparsity parameter performs best. On the other hand the sparse methods do perform better than the projected gradient method.

The main problem of the GD-CLS algorithm, as was also observed in [100], is that the performance using a higher number of components is quite poor. We observed that the mAP score dropped to 0.12, using 100 contexts, for all tested settings of the sparsity parameter.

6.6.4 Topic interpretation

The main advantage of the latent semantic modelling over plain TF-IDF is that we can inspect the keywords of the topics the model has extracted. The contexts are characterised using the term profiles for each topic, $p(t|k)$. The keywords can be found by looking at the top-ranking words for each topic. Table 6.7 gives the keywords for four of the topics produced by the PLSA-model. The first three topics describe quite general topics, the first concerning general rock artists, the other describes general pop terms, and the third describes a combination of jazz

Topic				
#1	#2	#3	#4	#5
members	top	jazz	meda	neurologic
tour	awards	pianist	quilmes	psilocybin
rock	award	composer	espacio	psychedelics
drummer	million	piano	paran	kesey
bands	television	orchestra	argentinean	flashbacks
guitarist	hot	compositions	programa	cryonics
formed	hits	musicians	brazil	interpersonal
guitar	career	saxophonist	revista	millbrook
bass	singer	classical	hecho	mushrooms
bassist	year	composition	libro	sante

Table 6.7: Four topics from a 100 topic model. The table shows the ten most relevant keywords based on $p(t|k)$. The first three components return quite musically relevant semantics, while the fourth reveals a topic related to South American artists

and classical music. The fourth topic on the other hand captures some South American terms, which is also evident for a number of other topics that are related to geography, while the fifth topic is related to drugs. The latent topics thus seem to span quite diverse aspects of the cultural background of music.

6.7 Comparison of Wikipedia and PLSA links

As discussed in section 6.4 we would like to use the semantically relevant links in Wikipedia for artist recommendations and playlist creation. This kind of links would typically be the ones that link related artists, and would also be the ones that reflect the kinds of relations that can be made using the PLSA model. This section compares the Wikipedia links with the PLSA based similarity.

The topics that are extracted using the PLSA model thus resembles the categories used in Wikipedia, and the direct links between articles can be inferred from the relevance measure, $p(d|d')$.

As an initial indicator we investigated how well the observed Wikipedia links measure against the relevance measure. First, we compare the PLSA relevance of the observed links versus the global distribution of relevance of all articles in figure 6.7. We are comforted by the finding that the actual pages linked in Wikipedia are typically closer in semantics than unlinked documents.

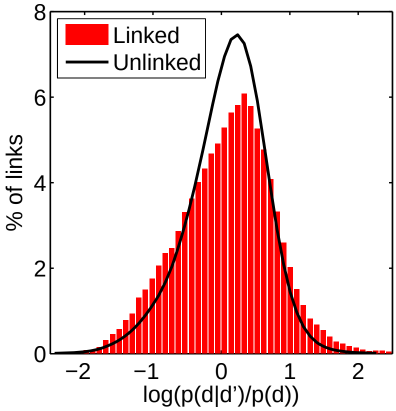


Figure 6.7: Histogram of similarity of linked and non-linked music articles.

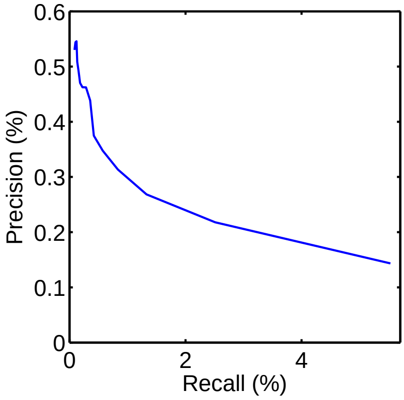


Figure 6.8: Precision and recall measures for predicting links

We next use the PLSA relevance to impose an alternative link structure. A simple strategy for generating links is to use a threshold on the artist similarities $p(d|d')/p(d)$ and choose links for those artists that are closest in the metric. In this way we can generate the same number of links as in the real link structure.

This predicted link structure can be evaluated against the real link structure using the standard measures of precision and recall;

$$\text{precision} = \frac{|\text{predicted links}| \bigcap |\text{real links}|}{|\text{predicted links}|} \tag{6.22}$$

and

$$\text{recall} = \frac{|\text{predicted links}| \bigcap |\text{real links}|}{|\text{real links}|} \tag{6.23}$$

The results of initial setup is shown in figure 6.7. The overall correspondence between the two networks is poor, which can be explained by two different aspects. The first is that the links in Wikipedia contain diverse links that are not described through the semantics that can be extracted from the text. Atypical collaborations, for instance between the German rock band 'The Scorpions' and the 'San Francisco Symphony', will not be captured by the semantic similarity, but actually is a relevant link in Wikipedia. The second sort of difference between the networks comes from what I consider to be an actual problem of the PLSA-based similarity, namely that the model finds even quite subtle relations between documents even though they are not musically relevant. These subtle relations are usually not a problem in the LSA methodology, because the paradigms used to test the models infer a top-down limitation of the possible

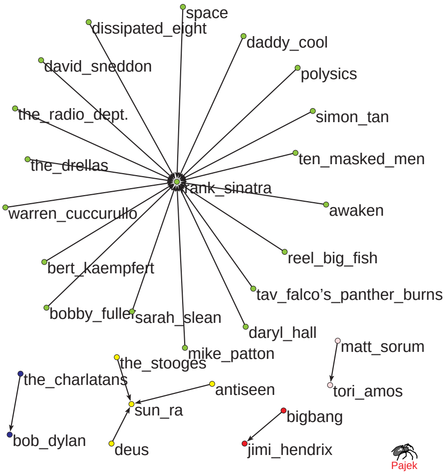


Figure 6.9: Most unlikely links based on the PLSA-model

answers to choose from. This is for instance the case in the synonym test, where the LSA is used to rank four different choices to find the one most similar to the query.

To better understand the disagreement between the observed and predicted links, we examine the links that are deemed most unlikely by the PLSA model. Evaluating $p(d|d')/p(d)$ for all pairs of articles d and d' gives a score of relevance for each of the links in Wikipedia. Some of the least likely links are shown in figure 6.9. The unlikely links are links from unknown artists to larger ones, and very many of the links are to *Frank Sinatra*. The two least likely links are for instance the ones from the death metal cover band *Ten Masked Men*, who have performed one of Sinatra's songs, and *Awaken* that is an underground rock band/indie music project from Belgium, who have also made a cover version of a Sinatra song. A brief overview of the rest of the least relevant links shows this general trend that there are a lot of spurious links from smaller artists to more well-established artists.

Investigating the links deemed most likely by the PLSA-model reveals the most interesting application of the model, namely for annotating the *context* of links. The most links deemed most relevant are for instance:

Link from	To
ed manion	the miami horns
kikki danielsson	elisabeth andreassen
alcazar	andreas lundstedt

1. **50 cent**
2. Dr. Dre
3. Busta Rhymes
4. Q-Tip
5. Talib Kweli
6. Kanye West
7. John Legend
8. Common
9. Pino Palladino
10. Tony Bennett
11. Bing Crosby
12. Louis Armstrong
13. Ella Fitzgerald
14. Billie Holiday
15. Benny Goodman

1. **Pearl Jam**
2. Led Zeppelin
3. P. J. Proby
4. Jackie Deshannon
5. Dolly Parton
6. Carrie Underwood
7. Kellie Pickler
8. Paris Bennett
9. Queen
10. Queen + Paul Rodgers
11. Bob Geldof
12. Ian Dury
13. Madness
14. No Doubt
15. Unwritten Law

Figure 6.10: Playlists based on random walks in Wikipedia.

The relevance of the link from the saxophone player *Ed Manion* to the horn section *The Miami Horns* is primarily based on the PLSA-context described by the keywords *sax*, *baritone*, and *juke*. The two other links are between Swedish pop stars, who have participated in the Eurovision Song Contest, and are primarily described by two PLSA-contexts with the keywords *contest*, *eurovision*, and *sweden*, *swedish*. The PLSA-model can thus be used to qualify relations between articles, and thus extends the relational knowledge that we described in section 2.1 with the context of the relation. The ability to describe why there is a relation between artists would be very useful for a music browsing tool, where links are described using the semantic contexts.

6.8 Artist based playlists

Given the inter-linked nature of Wikipedia it is natural to ask whether it can be used for playlist generation. Although, many songs have Wikipedia articles, we here focus for simplicity on the artist level. We compare here the properties of the random surfing in the observed Wikipedia links with PLSA directed surfing.

To illustrate we start the self-avoiding random walk using Wikipedia links from the Rap artist '50 cent', and 'Pearl Jam' respectively and obtained the playlists shown in figure 5. These playlists exhibit the "small world" effect that is seen in many networks, i.e., the average path length between any two nodes is very

1. **50 Cent**
2. The Notorious B.I.G.
3. Tupac Shakur
4. The Game
5. G-Unit
6. Ja Rule
7. Jadakiss
8. D-Block
9. Lloyd Banks
10. Shady Records
11. Mobb Deep
12. Young Buck
13. Cam'Ron
14. Jim Jones
15. Joe Budden

1. **Pearl Jam**
2. Nirvana
3. Mudhoney
4. Soundgarden
5. Alice In Chains
6. Mad Season
7. Layne Staley
8. Eddie Vedder
9. Jack Irons
10. Neil Young
11. Kurt Cobain
12. Led Zeppelin
13. The Who
14. Zak Starkey
15. The Waterboys

Figure 6.11: Playlists based on PLSI-directed walk in Wikipedia.

short and therefore it only takes a very few hops to get essentially to a random location. Thus the playlist generation must be guided in some way.

We propose to use the semantic knowledge of the PLSA , i.e., instead of following a random link, the link to the article d that has the highest relevance, $\log p(d|d')/p(d)$ given the current article d'[10]. The viability of the playlists produced using this setup are shown in figure 6.8. Clearly the directed playlists are much more focused than the 'random walks'. There are many ways of generalizing the approach, e.g., which songs to play and instead of following the most likely link we could create a random walk by selecting links with a probability derived from relevance etc. Finally, we note that the 'semantic' playlists could be viewed as an artist similarity metric and thus be evaluated against any of the many other available metrics.

6.9 Discussion

Wikipedia has significant potential as a common sense database for music information retrieval research. However, we face hard challenges to realise this potential due to the heterogeneity of content and overlinked hypertext structure.

[10] Note, this is different from the so-called intelligent surfer generalization of PageRank [93]

The semantic analysis showed some of the relevance of links made by Wikipedia editors. We found a number of less relevant links, e.g., between very high and low profile artists, that would have been more appropriately given as a separate list. The context given by the components in the semantic mixture model was shown to provide a context to artist-artist links. Together, these results point to applications, e.g., that the semantic analysis may be used for cleaning the Wikipedia link structure, or in potential modifications for 'hypermnesia' browsers that present links based on context.

Finally, we investigated the properties of a virtual 'stream of consciousness' following Wikipedia links. A simple self-avoiding random walk was shown to quickly get lost in the graph, while a semantically directed walk produced a more focused playlist.

Chapter 7

Multiway latent factor modelling of temporal music information

The latent semantics extracted in the previous approach was done without any knowledge of any of the obvious underlying mechanisms of musical understanding. The link structure analysed in the previous chapter is one way to extract musical knowledge from Wikipedia. But as was also mentioned the markup language used in Wikipedia can also be used to infer a range of more or less structured data that is more accessible than general Web data.

Our organisation of music is often based on using an ontology of genres. This organisation is often based on a notion of inheritance forming a sort of hierarchy, such that lower levels in the hierarchy are subgenres of the upper levels. This evolution of musical genres clearly has a temporal part which should be modelled. Is the rock by the Rolling Stones related to The Beatles and AC/DC in the same way? Using the knowledge of the era the music was created in could give some insights into these temporal influences.

Tensor methods in the context of text mining have recently received some attention using higher-order decomposition methods such as the Parallel Factors model (PARAFAC) [12] that to some extent can be seen as a generalization

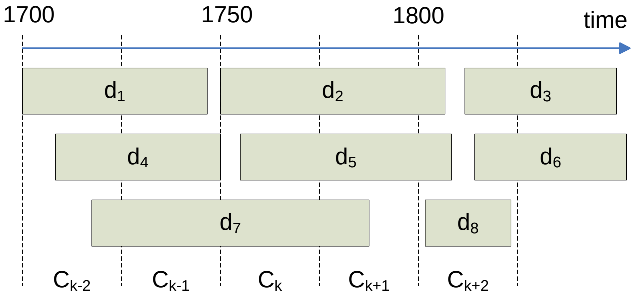


Figure 7.1: An example of assigning a collection of documents d_i based on the time intervals the documents belong to. The assignment produces a document collection C_k for each time interval

of Singular Value Decomposition in higher dimensional arrays. The article [12] applies tensor decomposition methods successfully for topic detection in e-mail correspondence over a 12 month period. The article also employs a non-negatively constrained PARAFAC model forming a Nonnegative Tensor Factorization which can be seen as a straightforward extension of the NMF-algorithm by Lee and Seung to tensors.

This chapter presents the application of temporal topic detection on musical data extracted from Wikipedia. The work presented in this chapter was in part published in the paper C.

7.1 Temporal topic detection

Detecting latent factors or topics in text using NMF and PLSA has assumed an unstructured and static collection of documents.

Extracting topics from a temporally changing text collection has received some attention lately, for instance by [78] and also touched by [16]. These works investigate text streams that contain documents that can be assigned a timestamp y. The timestamp may for instance be the time a news story was released, or in the case of articles describing artists it can be a timespan indicating the active years of the artist. Finding the evolution of topics over time requires assigning documents $d_1, d_2, ..., d_m$ in the collection to time intervals $y_1, y_2, ..., y_l$, as illustrated in figure 7.1. In contrast to the temporal topic detection approach in [78], we can assign documents to multiple time intervals, e.g. if the active years of an artist spans more than one of the chosen time intervals. The assignment of documents then provides l sub-collections $C_1, C_2, ..., C_l$ of documents.

The next step is to extract topics and track their evolution over time.

7.1.1 Stepwise temporal PLSA

The approaches to temporal topic detection presented in [78] and [16] employ latent factor methods to extract distinct topics for each time interval, and then compare the found topics at succeeding time intervals to link the topics over time to form temporal topics.

We extract topics from each sub-collection C_k using a PLSA-model [48]. The model assumes that documents are represented as a bags-of-words where each document d_i is represented by an n-dimensional vector of counts of the terms in the vocabulary, forming an $n \times m$ term by document matrix for each sub-collection C_k. PLSA is defined as a latent topic model, where documents and terms are assumed independent conditionally over topics z:

$$P(t, d)_k = \sum_{z}^{Z} P(t|z)_k P(d|z)_k P(z)_k \tag{7.1}$$

This model can be estimated using the Expectation Maximization (EM) algorithm, cf. [48].

The topic model found for each document sub-collection C_k with parameters, $\theta_k = \{P(t|z)_k, P(d|z)_k, P(z)_k\}$, need to be stringed together with the model for the next time span θ_{k+1}. The comparison of topics is done by comparing the term profiles $P(t|z)_k$ for the topics found in the PLSA model. The similarity of two profiles is naturally measured using the KL-divergence,

$$D(\theta_{k+1}||\theta_k) = \sum_t p(t|z)_{k+1} \log \frac{p(t|z)_{k+1}}{p(t|z)_k}. \tag{7.2}$$

Determining whether a topic is continued in the next time span is quite simply chosen based on a threshold λ, such that two topics are linked if $D(\theta_{k+1}||\theta_k)$ is smaller than a fixed threshold λ. In this case the asymmetric KL-divergence is used in accordance with [78]. The choice of the threshold must be tuned to find the temporal links that are relevant.

7.1.2 PARAFAC decomposition

The method presented above has been shown to be useful, but does not fully utilize the time information that is contained in the data. Some approaches have used the temporal aspect more directly, e.g. [27] where an incrementally trainable NMF-model is used to detect topics. This approach does include some of the temporal knowledge but still lacks global view of the important topics viewed over the whole corpus of texts.

We therefore investigate the use of multi-way arrays, representing the document collection as a three-way array $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$, where each observation x_{ijk} is the count of a specific term i in document j assigned to time slot k. The multi-way arrays or tensors are denoted using the boldface Euler script ($\boldsymbol{\mathcal{X}}$) in the following.

The extraction of latent topics using this tensor representation can be done using the PARAFAC method that is analogous to the SVD method used for two-way LSA modelling. The PARAFACdecomposition expresses the tensor as a sum of mode-1 tensors. The decomposition is usually written using the Kruskal operator, $\boldsymbol{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, with $\mathbf{A} \in \mathbb{R}^{T \times Z}$, $\mathbf{B} \in \mathbb{R}^{D \times Z}$, and $\mathbf{C} \in \mathbb{R}^{Y \times Z}$. $\boldsymbol{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is thus shorthand for:

$$\forall(t, d, y), x_{tdy} = \sum_{z=1}^{Z} a_{tz} b_{dz} c_{yz} \tag{7.3}$$

This decomposition can be estimated by minimising the following objective:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \left\| \boldsymbol{\mathcal{X}} - \sum_{r=1}^{R} a_{tz} b_{dz} c_{yz} \right\|^2 \tag{7.4}$$

to find the best possible $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The minimisation of the function can be done very efficiently using an alternating least squares method, such as described in [12], and implemented in the Tensor Toolbox [57].

The PARAFAC model can be seen as an expansion of the SVD to three dimensions, but it must be noted that the rank-1 decomposition matrices are not required to be orthogonal. Another widely used model is the Higher Order Singular Value Decomposition (HOSVD)[60], which does impose orthogonality on the components by matricising in each mode and calculating the singular values for each mode.

7.1.3 Multiway PLSA

The 2-way PLSA model in 7.1 can be extended to a 3-way model by also conditioning the topics over years y, as follows:

$$P(t, d, y) = \sum_z P(t|z)P(d|z)P(y|z)P(z) \tag{7.5}$$

The model parameters are estimated using maximum likelihood using the EM-algorithm, e.g. as in [116]. The derivation of the EM updates are analogous to the derivation of PLSA presented in section 4.2. The log-likelihood of this model is thus just an extension of the regular PLSA model, as follows:

$$\begin{aligned} \log p(X|model) &= \sum_d \sum_t \sum_y n(t, d, y) \log p(t, d, y) && (7.6)\\ &= \sum_d \sum_t \sum_y n(t, d, y) \log \sum_z p(z)p(t|z)p(d|z)p(y|z) && (7.7) \end{aligned}$$

The expectation step evaluates $P(z|t, d, y)$ using the estimated parameters at step t.

(E-step):
$$P(z|t, d, y) = \frac{p(t|z)p(d|z)p(y|z)p(z)}{\sum_{z'} p(t|z')p(d|z')p(y|z')p(z')} \tag{7.8}$$

The M-step then updates the parameter estimates.

(M-step):
$$\begin{aligned} P(z) =& \frac{1}{N} \sum_{tdy} x_{tdy} P(z|t, d, y) && (7.9)\\ P(t|z) =& \frac{\sum_{dy} x_{tdy} P(z|t, d, y)}{\sum_{tdy} x_{tdy} P(z|t, d, y)} && (7.10)\\ P(d|z) =& \frac{\sum_{ty} x_{tdy} P(z|t, d, y)}{\sum_{tdy} x_{tdy} P(z|t, d, y)} && (7.11)\\ P(y|z) =& \frac{\sum_{td} x_{tdy} P(z|t, d, y)}{\sum_{tdy} x_{tdy} P(z|t, d, y)} && (7.12) \end{aligned}$$

The EM algorithm is guaranteed to converge to a local maximum of the likelihood, but the EM algorithm is sensitive to initial conditions, so a number of methods to stabilize the estimation have been devised, e.g. Deterministic Anealing [48]. We have not employed these but instead rely on restarting the training procedure a number of times to find a good solution.

7.1.4 Topic model interpretation

The interpretation of the topics for the mwPLSA is of course analogous to the 2-way PLSA when it comes to keyword extraction, as $P(t|z)$ represents the probabilities of terms for the topic z, thus providing a way to find words that are representative of the topic. The most straightforward method to find these keywords is to use the words with the highest probability $P(t|z)$. This approach unfortunately seems somewhat flawed when we use a low number of components, as the histogram reflects the overall frequency of words, which means that generally common words tend to dominate the $P(t|z)$.

This effect can be neutralized by measuring the relevance of words in a topic relative to the probability in the other topics. Measuring the difference between the histograms for each topic can be measured by use of the symmetrized Kullback-Leibler divergence:

$$KL(z, \neg z) = \sum_t \underbrace{(P(t|z) - P(t|\neg z)) \log \frac{P(t|z)}{P(t|\neg z)}}_{w_t} \tag{7.13}$$

This quantity is a sum of contributions from each term t, w_t. The terms that contribute with a large value of w_t are those that are relatively more special for the topic z. w_t can thus be used to choose the keywords. The keywords should be chosen from the terms that have a positive value of $P(t|z) - P(t|\neg z)$ and with the largest w_t.

The time profiles $p(y|z)$ show the temporal evolution of the components, which might also be very useful to describe the contents. We can for instance see which topics overlap in time as the relevance of a topic in each time slot.

7.2 Wikipedia data

In this experiment we investigated the description of composers in Wikipedia. This should provide us with a dataset that spans a number of years, and provides a wide range of topics. We performed the analysis on the Wikipedia data dump saved 27th of July 2008, retrieving all documents that Wikipedia editors assigned to composer categories such as "Baroque composers" and "American composers". This produced a collection of 7358 documents, that were parsed so that only the running text was kept.

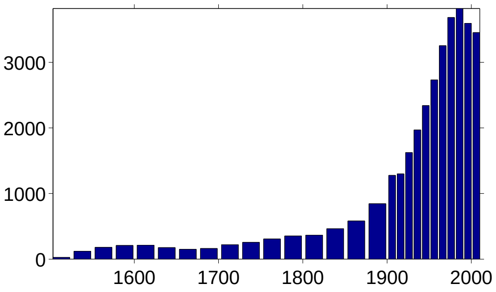


Figure 7.2: Number of composer documents assigned to each of the chosen time spans.

Initial investigations in music information Web mining showed that artist names can heavily bias the results. Therefore words occurring in titles of documents, such as *Wolfgang Amadeus Mozart*, are removed from the text corpus, i.e. occurrences of the terms 'wolfgang', 'amadeus', and 'mozart' were removed from all documents in the corpus. Furthermore we removed irrelevant stopwords based on a list of 551 words. Finally terms that occurred fewer than 3 times counted over the whole dataset and terms not occurring in at least 3 different documents were removed.

The document collection was then represented using a bag-of-words representation forming a term-document matrix $\mathbf{X}$ where each element x_{td} represents the count of term t in document d. The vector $\mathbf{x}_d$ thus represents the term histogram for document d.

To place the documents temporally the documents were parsed to find the birth and death dates. These data are supplied in Wikipedia as documents are assigned to categories such as "1928 births" and "2007 deaths". The dataset contains active composers from around 1500 until today. The next step was then to choose the time spans to use. Inspection of the data revealed that the number of composers before 1900 is quite limited so the artists were assigned to time intervals of 25 years, giving a first time interval of [1501-1525]. After 1900 the time intervals were set to 10 years, for instance [1901-1910]. Composers were assigned to time intervals if they were alive in some of the years. We estimated the years composers were active by removing the first 20 years of their lifetime. The resulting distribution of documents on the resulting 27 time intervals is seen in figure 7.2.

The term by document matrix was extended with the time information by assigning the term-vector for each composer document to each era, thus forming a 3-way tensor containing terms × documents × years. The tensor was further normalized over years, such that the weight of the document summed over years is the same as in the initial term doc-matrix. I.e. $P(d) = \sum_{t,y} X_{tdy} = \sum_t X_{td}$. This was done to avoid long-lived composers dominating the resulting topics.

The resulting tensor $\mathbf{X} \in \mathbb{R}^{m \times n \times l}$ contains 18536 terms x 7358 documents x 27 time slots with 4,038,752 non-zero entries (0.11% non-zero entries).

7.2.1 Term weighting

The performance of machine learning approaches in text mining often depends heavily on the preprocessing steps that are taken. Term weighting for LSA-like methods and NMF have thus shown to be paramount in getting interpretable results. We applied the well-known $tfidf$ weighting scheme, using $tf = \log(1+x_{tdy})$ and the log-entropy document weighting, $idf = 1+\sum_{d=1}^{D} \frac{h_{td} \log h_{td}}{logD}$, where $h_{td} = \frac{\sum_y x_{tdy}}{\sum_{dy} x_{tdy}}$. The log local weighting minimizes the effect of very frequent words, while the entropy global weight tries to discriminate important terms from common ones. The documents in Wikipedia differ quite a lot in length, therefore we employ document normalization to avoid that long articles dominate the modeled topics.

7.3 Experiments

We performed experiments on the Wikipedia composer data using the stepwise temporal PLSA method, PARAFAC and the multiway-PLSA methods. This section first presents some general observations on the topics extracted using the three methods. We show that the

7.3.1 Stepwise temporal PLSA

The step-by-step method was employed training 5 and 16 component models for each of the sub-collections of documents described above. The number of components was chosen to have one model that describes more general topics and a model with a higher number of components that can detect more specific topics. Using the higher number of topics makes it possible to leave out some of the less salient topics at each time step. The PLSA models for each time span was trained using a stopping criterion of 10^{-5} relative change of the cost

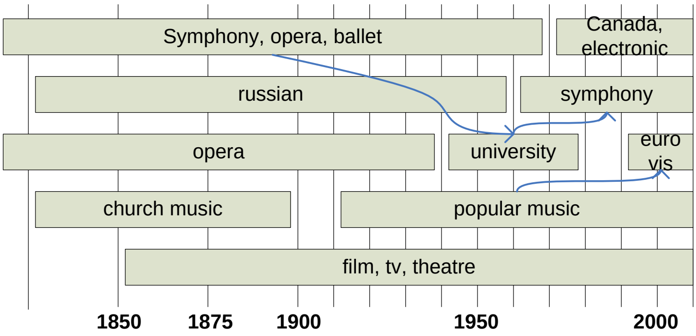


Figure 7.3: Topics detected using step-by-step PLSA. The topics are depicted as connected boxes, but are the results of the KL-divergence-based linking between time slots

function. We restarted the training a number of times for each model choosing the model minimizing the likelihood.

The topics extracted were then coupled together over time, picking topics that have a KL-divergence between the topic term distributions, $D(\theta_{k+1}|\theta_k)$, below the threshold λ. This choice of threshold produces a number of topics that stretch over several time spans. Tuning this threshold to identify the relevant links between topics in different time spans, makes this model somewhat difficult to tune. A low setting for λ may thus leave out some of the more subtle relations, while a higher setting produces too many links to be interpretable. Figure 7.4 shows the temporal linking of topics for the 5 component model. This representation does show that we are able to extract temporally linked topics from the model. Inspecting the keywords extracted for the PLSA components, we can identify semantics of the temporal topics, as shown in figure 7.3 where the 20th century topics are shown. There are clearly 4 topics that are present throughout the whole period. The topics are film and TV music composers, which in the beginning contains Broadway/theater composers. The other dominant topic describes hit music composers. Quite interestingly this topic forks off a topic describing Eurovision song contest composers in the last decades.

Even though the descriptions of artists in Wikipedia contain a lot of bibliographical information it seems that the latent topics have musically meaningful keywords. As there was no use of a special vocabulary in the preprocessing phase, it is not obvious that these musically relevant phrases would be found.

The stepwise temporal PLSA approach has two basic shortcomings. The first

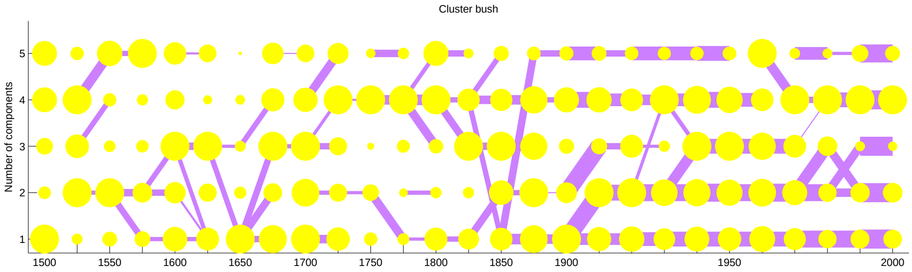


Figure 7.4: Plot of clusters for each time step for the 5 component PLSA model. The temporal linking is done using a threshold on the KL-similarity of term profiles of the PLSA topics.

being the problem of adjusting the threshold λ to find the meaningful topics over time. The second is to choose the number of components to use in the PLSA in each time span. The 5 topics that are used above do give some interpretable latent topics in the last decade as shown in figure 7.3. On the other hand the results for the earlier time spans that contain less data, means that the PLSA model finds some quite specific topics at these time spans. As an example the period 1626-1650 has the following topics:

1626-1650 41%	34%	15%	8.7%	1.4%
keyboard	madrigal	viol	baroque	anglican
organ	baroque	consort	italy	liturgi
surviv	motet	lute	poppea	prayer
italy	continuo	england	italian	respons
church	monodi	charles	lincoronazion	durham
nuremberg	renaissance	royalist	opera	english
choral	venetian	masqu	finta	chiefli
baroque	style	fretwork	era	england
germani	cappella	charless	venice	church
collect	itali	court	teatro	choral

The topics found here are quite meaningful in describing the baroque period, as the first topic describes the church music, and the second seems to find the musical styles, such as madrigals and motets. The last topic on the other hand only has a topic weight of $P(z) = 1.4\%$ suggesting that it might be somewhat specific.

The topics extracted for the last century seem semantically reasonable, but are very broad. If we add more topics to the models more specific topics will be extracted. The graph showing the temporal topics is shown in figure 7.5.

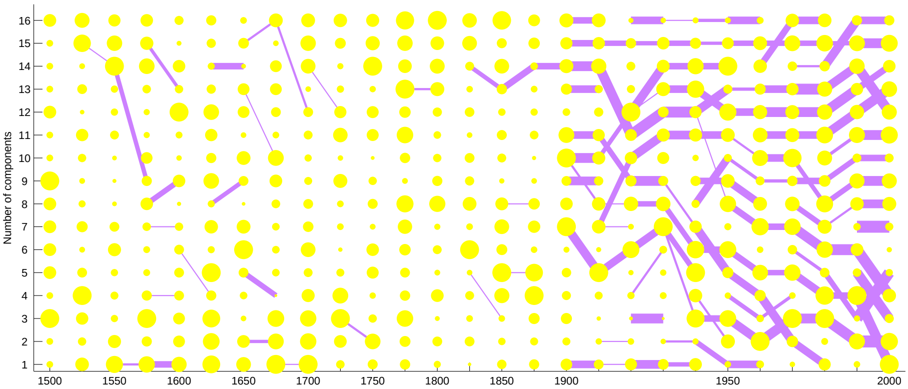


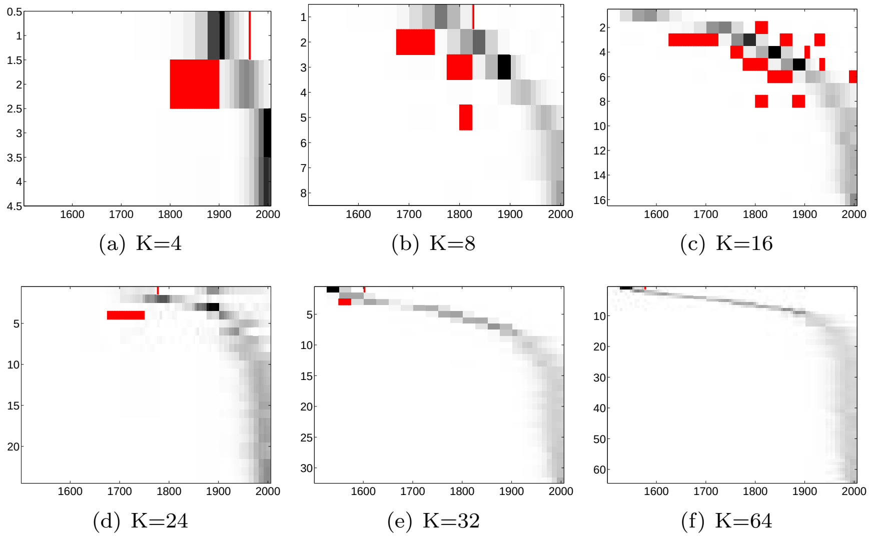
Figure 7.5: Plot of clusters for each time step for the 16 component PLSA model. The temporal linking is done using a threshold on the KL-similarity of term profiles of the PLSA topics. The stepwise PLSA approach finds clear temporal topics in the last century but clearly a lower threshold is needed at the earlier time steps.

7.3.2 PARAFAC

The main advantage of the tensor decomposition methods is that the temporal linking of topics is accomplished directly through the model estimation. This flexibility allows us to find temporal topics by fitting models using different numbers of components to find the inherent structure of the data. We first investigate the topics found using the PARAFAC and mwPLSA models.

The PARAFAC model was trained on the entropy weighted tensor until a change in relative fit of 10^{-5} from one iteration to the next was reached, which was the standard used in the Tensor Toolbox. The models were chosen from 5 runs of the algorithm with random initialisations, picking the model with the best fit.

The interpretation of the PARAFAC model is given by inspecting the decomposition $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where the columns of the $\mathbf{A}$ and $\mathbf{B}$ matrices gives the weights for terms and documents, respectively, while the $\mathbf{C}$ matrix gives a profile over time, showing at which times the topics are active. Figure 7.6 shows a visualisation of $\mathbf{C}$ for 4, 8, 16, and 32 component models. The weights of $\mathbf{C}$ are plotted as a "heatmap", where darker colours correspond to larger positive values, while a red color denotes that the weights are negative.



(a) K=4

(b) K=8

(c) K=16

(d) K=24

(e) K=32

(f) K=64

Figure 7.6: Time view of PARAFAC components

The topics are generally unimodally distributed over time so the model only finds topics that increase in relevance and then dies off without reappearing. The skewed distribution of documents over time which we described earlier emerges clearly in the distribution of topics, as most of the topics are placed in the last century.

As an example of the components that are generated from the PARAFAC model we show the terms that have the highest values for each component for the 32 component model.

7.3.3 Multi-way PLSA

Modeling using the MWPLSA model was also performed on the tensor described in section 7.2. Analogously with the stepwise temporal PLSA model we stopped training reaching a change of less than 10^{-5} of the cost function.

The time evolution of topics can be visualized using the parameter $P(y|z)$ that gives the weight of a component for each time span. Figure 7.7 shows the result for 4, 8, 16, 32 and 64 components using the same heatmap-representation as used above, where darker colors correspond to higher values of $P(y|z)$. The topics are generally look very like the ones extracted by PARAFAC .

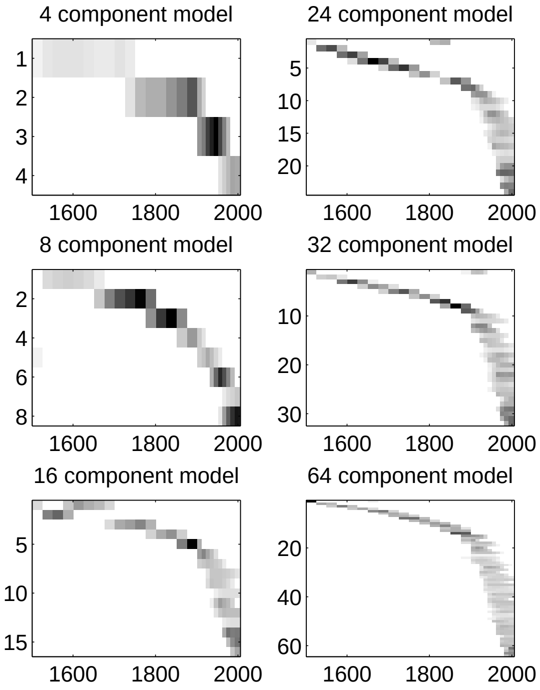


Figure 7.7: Time view of components extracted using MWPLSA, showing the time profiles $P(y|z)$ as a heatmap. A dark color corresponds higher values of $P(y|z)$

Adding more topics to the model has two effects considering the plots. Firstly the topics seem to be sparser in time, finding topics that span fewer years, the difference is quite clear comparing the 64 component model of PARAFAC and mwPLSA . The mwPLSA finds topics that are constrained to the first half of the 1900's, while the PARAFAC topics span the full century.

Using more topics decomposes the last century into more topics that must be semantically different as they almost all span the same years. The keywords of some of the topics for the 32 component model are shown in table 7.1, including the time spans that they belong to.

The first topic in table 7.1 is one of the two topics that accounts for the years 1626-1650, and it is clear that the keywords summarize the five topics found using the multiway PLSA. The last four topics are some of the 24 topics that describe the last century. The second topic has the keywords ragtime and rag, placed in the years 1876-1921, which aligns remarkably well with the genre

1601-1700	1876-1921	1921-1981	1921-1981	1971-2008
2.10%	2.40%	4.70%	4.80%	6.40%
baroque	ragtime	concerto	broadway	single
continuo	sheet	nazi	hit	chart
motet	rag	war	songwriter	album
viol	weltemignon	symphony	film	release
survive	nunc	piano	mgm	hit
basso	ysa	ballet	vaudeville	track
italian	schottisch	neoclassical	lyricist	sold
court	dimitti	neoclassic	benni	demo
church	blanch	choir	bing	fan
cathedral	parri	hochschul	inter	pop

Table 7.1: Keywords for 5 of the 32 components in a mwPLSA model. The assignment of years is given from $P(y|z)$ and percentages placed at each column are the corresponding component weights, $P(z)$

description on Wikipedia: *”Ragtime [...] is an originally American musical genre which enjoyed its peak popularity between 1897 and 1918.”*[1] Comparing to the stepwise temporal PLSA approach ragtime also appeared as keywords in the 16 component model, appearing as a topic from 1901-1920. The next topic seems to describe World War II, but also contains the neoclassical movement in classical music. The 16 component stepwise temporal PLSA approach finds a number of topics from 1921-1940 that describe the war, such as a topic in 1921-1930 with keywords: *war, time, year, life, influence* and two topics in 1931-1941, *1. time, war, year, life, style* and *2: theresienstadt, camp, auschwitz, deport, concentration, nazi.* These are quite unrelated to music, so it is evident that the global view of topics employed in the mwPLSA-model identifies neoclassicism to be the important keywords compared to topics from other time spans.

7.4 Model selection

A crucial issue in both tensor decomposition and mixture modelling is to select the appropriate number of components, i.e. the value of K. Observing visualisations of the data using different number of components may yield some insights and the use of ground truth data to qualify the choice of K. In this section we investigate some ways of estimating the ”right” number of components for our mixture model.

We will first apply the well-known Akaike’s information criterion (AIC). The second part investigates a more heuristic method which does not rely on the assumptions used by the information criteria.

[1] http://en.wikipedia.org/wiki/Ragtime

7.4.1 AIC model selection

The probabilistic view of mwPLSA has the advantage that we can rely on techniques earlier proposed for model selection. The most well-known measure is Akaike's Information Criterion[6].

As the model complexity increases, the Maximum of the likelihood also increases because of the additional degrees of freedom in the model. Information criteria have been developed using various arguments, forming different penalty terms on the likelihood function. The AIC takes the very simple form, given by:

$$AIC = -2L(\hat{\theta}) + 2P, \tag{7.14}$$

where $L(\hat{\theta})$ is the log-likelihood of our ML parameters $\hat{\theta}$ and P is the number of parameters in the model. The assumptions for the derivation of the AIC do not hold for the mixture of multinomials model we employ here, so the results here can only be used as an indication. We therefore just investigate how the results align with the empirical observations below. The number of parameters P could be calculated as $T * Z + D * Z + Y * Z + Z$. As the number non-zero parameters, and therefore the number effective parameters, is much smaller than this number, we use the number of non-zero parameters for P.

We trained the models for a varying number of components and calculated the AIC-penalised likelihood plotted in figure 7.8. It shows that the number of components should be in the range 6-12.

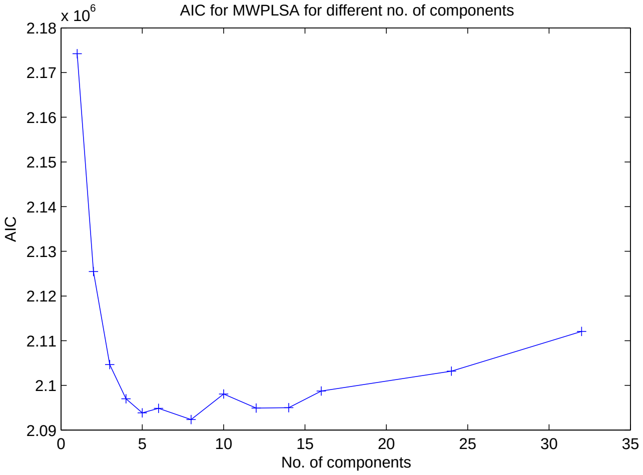


Figure 7.8: Plot showing the AIC for a varying number of components

7.4.2 Consensus clustering model selection

In this section we investigate a method to select the number of components based on a heuristic method called consensus clustering [80].

The basic idea of this approach is to investigate how stable a clustering using resampling of the data. If the clustering assigns documents to the same clusters under different conditions then this is the correct number of clusters, i.e. if there is consensus.

The method was initially devised for clustering algorithms, so in the following we will treat the decompositions as clusterings by assigning each document to one of the components of the models.

The clustering assignments for the PARAFAC model is based on the document loadings in $\mathbf{B}$, weighted with the component weight λ,

$$\arg\max_z |\lambda \mathbf{B}|. \tag{7.15}$$

This approach for clustering may be a little crude because of the negative and positive numbers that occur in the matrices.

Clustering using the PLSA models is more well-founded. Given the document profiles $P(d|z)$ and the mixture weights we can simply calculate the posterior probability of a document d being assigned to a topic z using Bayes' theorem;

$$\arg\max_z P(z|d) = \frac{P(d|z)P(z)}{\sum_z P(d|z)P(z)} \tag{7.16}$$

This both applies for both the two- and three-way PLSA models. Selecting the number of components in the step-wise PLSA model was not considered as the selection of relevant topics in part comes from which topics linked at different time steps, which is somewhat more difficult to model.

Consensus matrix

The basic idea of investigating how stable a clustering is under different clusterings was used in [80] to devise the so-called consensus clustering. The algorithm is based on resampling or subsampling a dataset, to form H different datasets $D^{(h)}$. The clustering algorithm is then run on each dataset, and for each resulting clustering a co-occurrence matrix $M^{(h)}$ is formed. The matrix $M^{(h)}$ simply indicates whether two documents are clustered together:

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if items i and j belong to same cluster} \\ 0 & \text{else} \end{cases}$$

Finally, let $I^{(h)}$ be the $(N \times N)$ indicator matrix such that its (i,j)-th entry is equal to 1 if both items i and j are present in the dataset $D^{(h)}$, and 0 otherwise. The need for the indicator matrix is due to the use of subsampling. The indicator matrix keeps track of the number of iterations in which two items are both included in the subsampled dataset.

The consensus matrix M can then be defined as a properly normalized sum of the connectivity matrices of all the perturbed datasets $D^{(h)} : h = 1, 2, ..., H$:

$$M(i,j) = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)} \tag{7.17}$$

That is, the entry (i,j) in the consensus matrix records the number of times items i and j are assigned to the same cluster divided by the total number of times both items are selected. The entries of M range from 0 to 1 and reflect the probability that samples i and j cluster together. The number of subsampling runs is chosen by continuing until the matrix M^h seems to stabilize. This took around 40-50 runs in our case.

By using the off-diagonal entries of M as a measure of similarity among samples, we use average linkage hierarchical clustering to reorder the samples and thus the rows and columns of M. Plotting the matrix using the ordering shows how stable the clustering is. A perfect clustering will be represented with a number blocks of 1's surrounded by 0's. If there is confusion between the clusters then the matrix will have a number of values that are between 0 and 1.

Visual inspection of the reordered matrix $\bar{C}$ provides a very good insight into the clustering structure, as we will see below, it is also important to have a quantitative measure of the stability of the clusters for each value of K. For this we calculate the cophenetic correlation coefficient (as suggested in [24])

The cophenetic correlation is calculated given the clustering tree calculated using the linkage algorithm, we used to sort the documents for visualisation, and measures how well the dendrogram built in the linkage algorithm preserves the pairwise distances between the original unmodelled data points.

The cophenetic distance between two observations is represented in a dendrogram by the height of the link at which those two observations are first joined.

That height is the distance between the two subclusters that are merged by that link.

The actual cophenetic distance is calculated as the linear correlation between the distances, i.e.

- $x(i,j) = |Xi - Xj|^2$, the original distance measured as the Euclidean distance between the ith and jth observations.
- $t(i,j) =$ the cophenetic distance between the model points T_i and T_j.

Then, letting $\bar{x}$ be the average of the $x(i,j)$-distances, and letting $\bar{t}$ be the average of the $t(i,j)$, the cophenetic correlation coefficient c is given by

$$cc = \frac{\sum_{i<j}(x(i,j) - \bar{x})(t(i,j) - \bar{t})}{\sqrt{\left[\sum_{i<j}(x(i,j) - \bar{x})^2\right]\left[\sum_{i<j}(t(i,j) - \bar{t})^2\right]}} \tag{7.18}$$

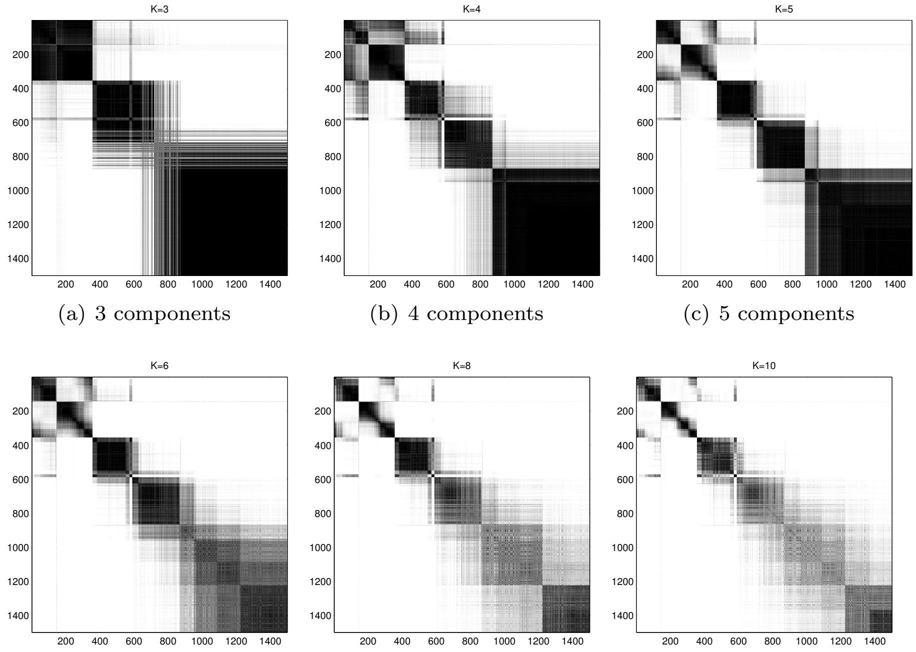
This measure should be close to 1 for a very good division of points into clusters, and can thus be used to compare different clusterings of points to quantitatively produces the best.

7.4.3 Evaluation of consensus clustering

Consensus matrices were constructed for PARAFAC and mwPLSA models using z=2,3,4,5,6,8,10 and using 2-way PLSA, that does not use the time information. The consensus matrices are shown in figure 7.9, 7.10, and 7.11 for mwPLSA, PARAFAC and the 2-way PLSA respectively.

The figures show that both mwPLSA and PARAFAC find a clustered structure in the data. The 2 component model provides a very clear clustering but this may not be very informative of the inherent topics in the data. Overall the five topic model provides a nice clustering, while the models with a higher number topics seems to confuse documents between the clusters.

The main difference between the two tensor decomposition methods is seen in the model for 5 topics. mwPLSA finds 5 clusters, while the PARAFAC model only reveals 4 clusters. The 5th component in the PARAFAC decomposition seems to be a very small cluster that can be seen between the 2nd and 3rd large block on the diagonal. These documents seem to belong to both the 1st and 2nd cluster.



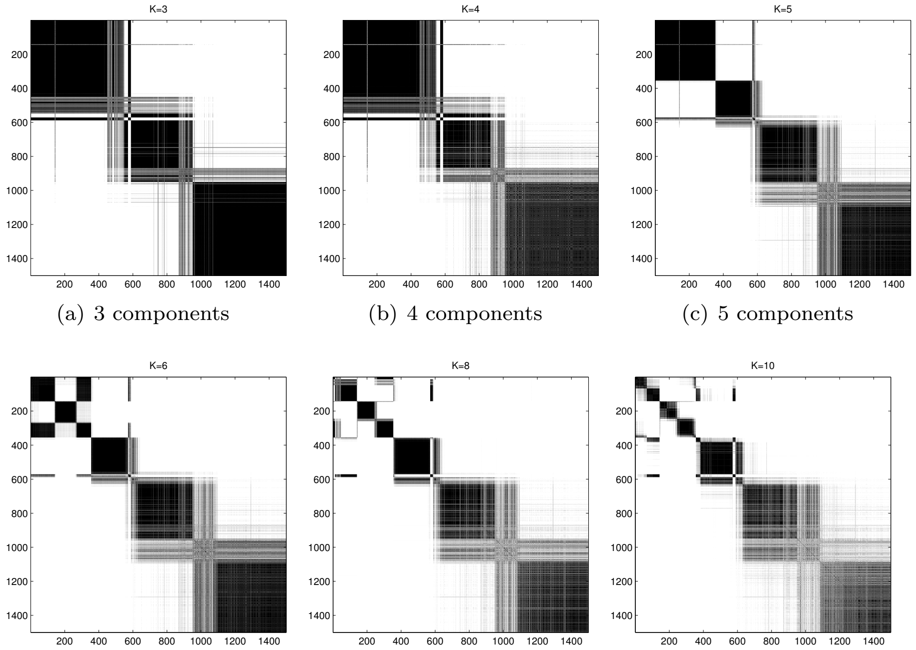
(a) 3 components (b) 4 components (c) 5 components

(d) 6 components (e) 8 components (f) 10 components

Figure 7.9: Consensus matrices for MWPLSA using a range of no. of clusters. The clear black blocks on the diagonal are evidence of stable clusters.

Comparing the tensor models to the 2-way PLSA model shows that the lack of time information in the 2-way model means that no clear clustering is obtainable.

The three different methods can also be compared more directly using the consensus clustering scheme. Figure 7.12 shows the clustering found by mwPLSA, PARAFAC, and 2-way PLSA. The documents used in the consensus matrix are all ordered according to the ordering used for the mwPLSA decomposition. It is evident that the PLSA model only finds one of the clusters found by the algorithms that use the time information. The PARAFAC decomposition on the other hand seems to miss the first and second cluster found by MWPLSA, as seen by the grouping of those two clusters in figure 7.12(b).



(a) 3 components (b) 4 components (c) 5 components

(d) 6 components (e) 8 components (f) 10 components

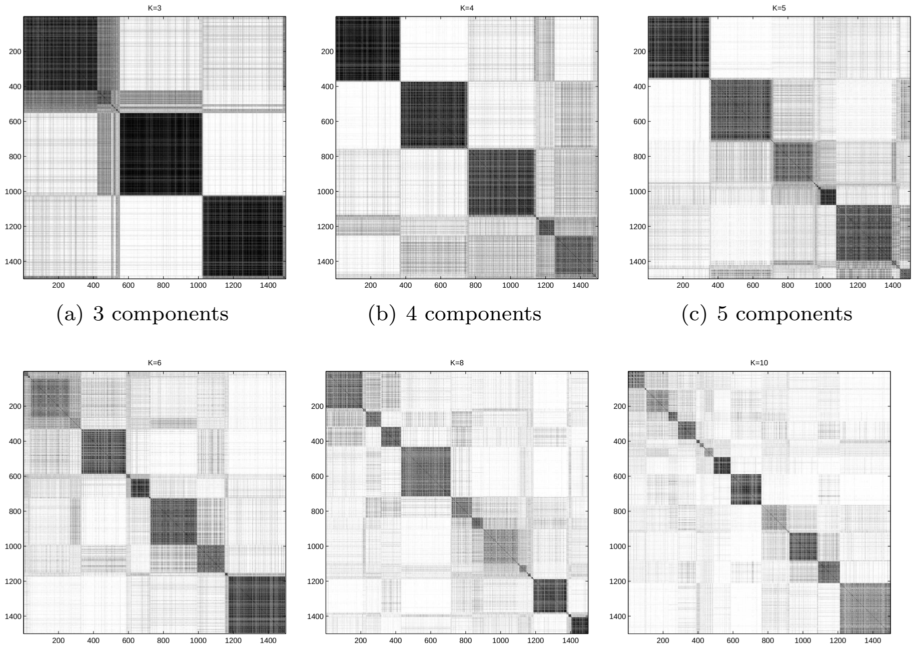
Figure 7.10: Consensus matrices for PARAFAC using a range of no. of clusters. The clear black blocks on the diagonal are evidence of stable clusters.

Cophenetic correlation

Comparing the results for mwPLSA for different number of components shows that the most robust clustering is obtained by using 5 components. The cophenetic correlation coefficient described in section 7.4.2, was evaluated for the mwPLSA-model. The result is shown in figure 7.13. As long as the cophenetic correlation increases the extra components provide more information, so we should choose the number of components that has the highest score, which incidentally is 5 components.

7.4.4 Example of clustering

Using 5 components may seem like a low number, considering the many musical directions that have been through the times. We therefore inspect the 5 component model to see whether the extracted topics are meaningful. The



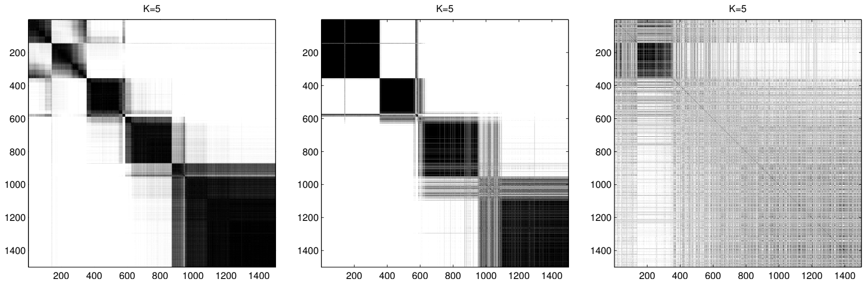
(a) 3 components (b) 4 components (c) 5 components

(d) 6 components (e) 8 components (f) 10 components

Figure 7.11: Consensus matrices for NMF using a range of no. of clusters. The clear black blocks on the diagonal are evidence of stable clusters.

topics are interpreted by extracting the keywords as described in 7.1.4. The keywords for the MWPLSA-model are shown below, and corresponding plot of the components over time is shown in figure 7.4.4.

1525-1800 11%	1800-1900 10%	1876-1941 17%	1921-1981 23%	1951-2008 38%
major	opera	piano	record	album
keyboard	paris	war	jazz	release
madrigal	vienna	ja	film	award
court	op	conservatory	mingus	band
motet	piano	js	arrange	film
italian	dagoult	conductor	serial	ensemble
flat	petersburg	symphony	radio	rock
survive	die	pour	sinatra	festival
venice	concert	die	york	universal
lost	leipzig	conduct	state	record



(a) MWPLSA (b) PARAFAC (c) NMF

Figure 7.12: Consensus matrices for 5 component models

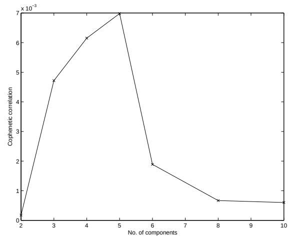


Figure 7.13: Cophenetic coefficient for MWPLSA using different number of components.

The keywords show that the topics seem to be described by musically meaningful keywords, although the words seem to be quite broad descriptors of music. The last two topics are somewhat larger than the first three, which reflects the skewed distribution of documents in the data. The topics can therefore not be very specific as they represent a number of genres. So in case we want to get a meaningful clustering of the last century, the model needs more topics.

7.5 Multi-resolution topics

The use of different number of components in the mwPLSA model, as seen in figure 7.7, shows that the addition of topics to the model shrinks the number of years they span. The higher specificity of the topics when using more components gives a possibility to "zoom" in on interesting topic, while the low complexity models can provide the long lines in the data.

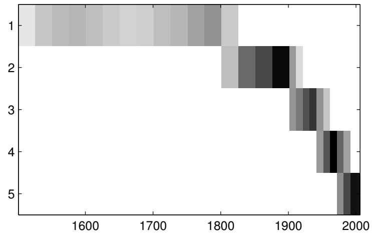


Figure 7.14: Time view of the 5 component MWPLSA-model

To illustrate how the clusters are related as we add topics to the model, we can generate a so-called clusterbush, as proposed in [83]. The result for the MWPLSA-based clustering is shown in figure 7.15. The clusters are sorted such that the clusters placed earliest in time are placed left. It is evident the clusters related to composers from the earlier centuries form small clusters that are very stable, while the later components are somewhat more ambiguous. The clusterbush could therefore be good tool for exploring the topics at different timespans to get an estimate of the number of components needed to describe the data. It is for instance quite evident that the years before 1700, are quite adequately described using two clusters, while further topics should be assigned to the 20th century.

7.6 Discussion

We have investigated the use of time information in analysis musical text in Wikipedia. It was shown that the use of time information produces meaningful latent topics, which are not readily extractable from this text collection without any prior knowledge.

The multiway PLSA shows some definite advantages in finding topics that are temporally defined. The stepwise temporal PLSA approach is quite fast to train and processing for each time span can readily be processed in parallel. But it has the practical drawback that it requires a manual tuning of the linking threshold, and the lack of a global view of time in the training of PLSA models misses some topics as shown above. The training of multiway PLSA is somewhat slower than the step-by-step approach but the more flexible representation that the model gives is a definite advantage, for instance when data has a skewed distribution as in the work resented here. The global model would also make it possible to do model selection over all time steps directly.

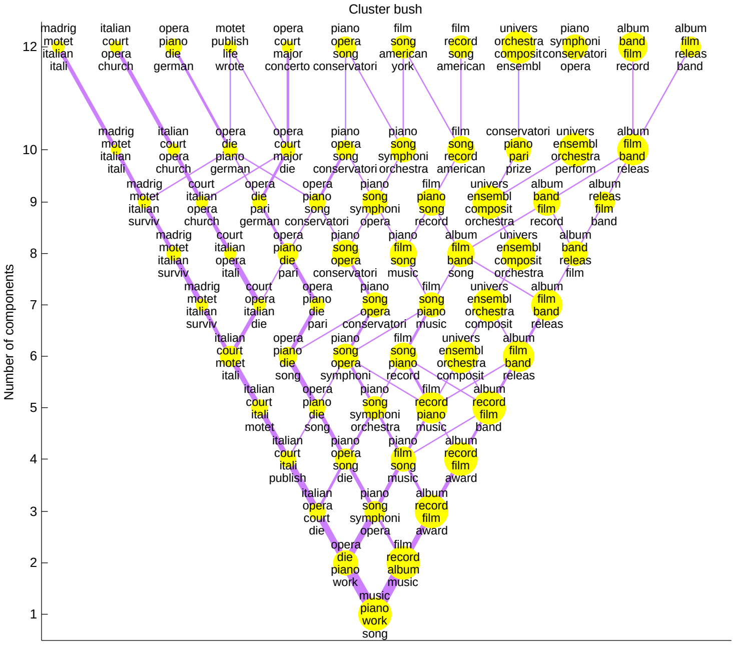


Figure 7.15: "Cluster bush" visualisation of the results of the MWPLSA clustering of composers. The size of the circles correspond to the weight of the cluster, and the thickness of the line between circles how related the clusters are. Each cluster is represented by the keywords and is placed according to time from left to right.

The use of the structural data from Wikipedia data also seems to be a very useful resource for MIR for instance for temporal browsing of artists, and maybe even more relevant as a tool for finding artists that have influenced others.

Chapter 8

Conclusion

The application of automatical indexing of multimedia data, and especially sound is still in its infancy. Image retrieval has been researched for a number of years with resulting big advances. The understanding of sound and music on the other hand still leaves room for much work. Especially in the domain of music we have seen that an understanding of the cultural background is an important factor for automatic recommendations and navigation. This thesis has investigated methods that can be useful for identifying the context of sound signals which may be used to improve retrieval and navigation of sound collections.

Latent semantics of text describing multimedia have been investigated as a method to extract some of the background knowledge for sound retrieval.

We showed how the latent semantic approach could be used for automatically transcribed spoken document retrieval. The probabilistic semantic analysis was approximated using a least squares cost non-negative matrix factorisation, which can be seen as a gaussian approximation to the multinomial distribution of words. The count-based PLSA model, was relaxed using the NMF model, such that the entries of the term-document matrix are seen as real values. The latent topic representation of the spoken document database was used to perform query-expansion, which we suggest to alleviate the problems of retrieving erroneously transcribed documents. The method was demonstrated in a Web-based demo for retrieval of CNN broadcast news.

The second part of the work has investigated how to collect knowledge about and model the context of music. We identified the principal methods for extracting the cultural background of music through general Web mining or social methods, such as tagging or collaborative filtering. The unstructured and noisy data available in the Web requires very focused crawling and extensive filtering to produce useful data for subsequent learning. The social media seem to have a strong popularity bias, which on the positive side means that well-known artists can be described very reliably, on the downside it means that niche-artists and genres may be very ill-represented in these systems. We instead investigate the compromise between the two domains in the form of the articles in the musical Wikipedia.

We investigated the linking structure of Wikipedia for use as a sort of semantic relatedness engine. Using methods from network analysis we found that the linking structure reasonably reflects the what we expect in the form of background knowledge of music. The semantics of the links in Wikipedia was further compared to the latent semantic-based similarity of artists, derived from the text-parts of the articles. We propose to use the linking structure of Wikipedia in combination with the PLSA similarity to produce playlists.

Last we use one aspect of the structure in Wikipedia data to incorporate time knowledge into the semantic modelling of musical composers. We investigated the use of stepwise temporal PLSA, as well as tensor-methods in the form of PARAFAC and multiway PLSA, for topic extraction. We demonstrated that the global view of time employed by the tensor methods makes it possible to detect topics that are not identifiable using the stepwise approach. The tensor methods also make it possible to do model selection for the full problem in one go. Model selection was investigated using consensus clustering to find the appropriate number of topics. The consensus clustering shows that the multiway PLSA produces a clearer clustering than PARAFAC. Even though the model selection criteria produce suggestions for a specific number of topics, we observe that the interpretability of topics at different model sizes produces interesting ways to examine the data. This observation led to the use of the cluster bush representation to investigate how different models with different model orders split the data into topics.

Further directions

Use of Wikipedia as a common knowledge resource for music retrieval points to a number of further research directions.

The relational meaning we extract would definitely benefit from a systematic human evaluation to assess if the extracted meaning is useful for users. This could for instance be implemented in a system for music browsing and playlisting.

The analysis of Wikipedia showed that the linking structure is an interesting resource which could be analysed further, for instance by extracting more local knowledge, for instance by finding community structure of the artist network. This knowledge could be used to find structures similar to the latent topics based on the linking structure. The availability of a hypertext such as Wikipedia also lends itself naturally to a combined modeling of text and links along the lines of [34, 82].

The retrieval of contextual knowledge for multimedia is still a field that needs further attention, as it will be necessary to investigate the sorts of meaning that I outlined in section 2.1 further. Harnessing the boundless amounts of digital media will depend on the success of these methods.

Appendix A

Artist similarity ground truth

The following tables show the 195 artists used as a ground truth for similarity measurements. The collection was initially gathered by Schedl et al. [97], and is given here for reference. The original collection of artists consisted the classical genre, but this was not useful in our experiments and were therefore not included. Some of the artists from the remaining genres were not present in our Wikipedia data, and were not used, giving an active set of 172 artists. These artists are marked with asterisks (*) in the tables.

country	folk	jazz	blues
johnny cash	bob dylan	thelonious monk	john lee hooker
willie nelson	joni mitchell	dave brubeck	muddy waters
dolly parton	crosby stills nash	billie holiday	taj mahal
hank williams	joan baez	duke ellington	john mayall
faith hill	townes van zandt	django reinhardt	big bill broonzy
dixie chicks	don mclean	glenn miller	b. b. King
garth brooks	suzanne vega	ella fitzgerald	howlin wolf
kenny rogers	tracy chapman	louis armstrong	albert king
tim mcgraw	tim buckley	cannonball adderley	blind lemon jefferson
hank snow	steeleye span	herbie hancock	blind willie mctell
brooks and dunn	woody guthrie	nina simone	mississippi john hurt
lee hazlewood	donovan	john coltrane	otis rush
kenny chesney	cat stevens	charlie parker*	etta james
jim reeves	john denver	count basie*	lightnin hopkins
roger miller	pete seeger*	miles davis*	t-bone walker*
kris kristofferson	leonard cohen*	nat king cole*	willie dixon*

rnbsoul	heavy	altindie	punk
james brown	iron maiden	nirvana	patti smith
marvin gaye	megadeth	beck	sex pistols
otis redding	slayer	the smashing pumpkins	pennywise
solomon burke	sepultura	radiohead	ramones
sam cooke	black sabbath	belle and sebastian	bad religion
aretha franklin	anthrax	jane's addiction	the clash
al green	alice cooper	echo and the bunnymen	nofx
the temptations	deep purple	sonic youth	dead kennedys
the drifters	def leppard	weezer	buzzcocks
fats domino	ac-dc	pearl jam	green day
the supremes	judas priest	foo fighters	blink-182
isaac hayes	kiss	hole	sum 41
alicia keys	metallica	bush	the misfits
erykah badu	pantera	the smiths	rancid
india.arie	queensryche	depeche mode	screeching weasel
jill scott	skid row	alice in chains*	sid vicious*

rap hiphop	electro	reggae	rocknroll
eminem	aphex twin	bob marley	the rolling stones
dr dre	daft punk	jimmy cliff	the animals
public enemy	kraftwerk	peter tosh	faces
missy elliot	chemical brothers	ziggy marley	the kinks
cypress hill	fatboy slim	sean paul	gene vincent
50 cent	basement jaxx	alpha blondie	elvis presley
mystikal	carl cox	shaggy	chuck berry
grandmaster flash	mouse on mars	maxi priest	little richard
2pac	paul oakenfold	shabba ranks	jerry lee lewis
snoop dogg	prodigy	ub40	buddy holly
jay-z	armand van helden	inner circle	bo diddley
busta rhymes	moby	eddy grant	bill haley
ll cool j	massive attack	bounty killer	the yardbirds
dmx	moloko*	capleton*	chubby checker*
ice cube	jimi tenor*	desmond dekker*	carl perkins*
run dmc*	underworld*	black uhuru*	the who*

pop
madonna
britney spears
the animals
michael jackson
janet jackson
spice girls
christina aguilera
robbie williams
nelly furtado
avril lavigne
jennifer lopez
o-town
n sync*
justin timberlake*
prince*
abba*
shakira*

Appendix B

Castsearch - Context-aware search in spoken document database

CASTSEARCH - CONTEXT BASED SPOKEN DOCUMENT RETRIEVAL

Lasse Lohilahti Mølgaard, Kasper Winther Jørgensen, and Lars Kai Hansen

Informatics and Mathematical Modelling
Technical University of Denmark Richard Petersens Plads
Building 321, DK-2800 Kongens Lyngby, Denmark

ABSTRACT

The paper describes our work on the development of a system for retrieval of relevant stories from broadcast news. The system utilizes a combination of audio processing and text mining. The audio processing consists of a segmentation step that partitions the audio into speech and music. The speech is further segmented into speaker segments and then transcribed using an automatic speech recognition system, to yield text input for clustering using non-negative matrix factorization (NMF). We find semantic topics that are used to evaluate the performance for topic detection. Based on these topics we show that a novel query expansion can be performed to return more intelligent search results. We also show that the query expansion helps overcome errors inferred by the automatic transcription.

***Index Terms*—** Audio Retrieval, Document Clustering, Non-negative Matrix Factorization, Text Mining

1. INTRODUCTION

The rapidly increasing availability of audio data via the Internet has created a need for automatic sound indexing. However, broadcast news and other podcasts often include multiple speakers in widely different environments which makes indexing hard, combining challenges in both audio signal analysis and text segmentation.

Access to broadcast news can be aided by topic detection methods to retrieve only relevant parts of the broadcasts. Efficient indexing of such audio data will have many applications in search and information retrieval. The spoken document indexing issue has been approached in different systems notably in the 'Speechfind' project described in [1]. This project utilizes audio segmentation as well as automatic speech transcription to retrieve relevant sub-segments.

Segmentation of broadcast news to find topic boundaries can be approached at two different levels. Starting from analysis of the audio, locating parts that contain the same speaker in the same environment can indicate story boundaries and may be used to improve automatic speech recognition performance. We have investigated this approach in [2].

The speaker segments generated through the audio analysis can then be processed through an automatic speech-to-text system to generate transcripts. Utilizing these transcripts enables a top-down segmentation based on the semantic content of the news stories. The area of topic detection in text has been widely researched for the last decade, see e.g., [3] for a presentation. Though, automatically transcribed text poses additional difficulties than manually transcripts, due to imperfect transcriptions. We utilize non-negative matrix factorization (NMF) for document topic detection. NMF has earlier shown to yield good results for this purpose, see e.g., [4, 5].

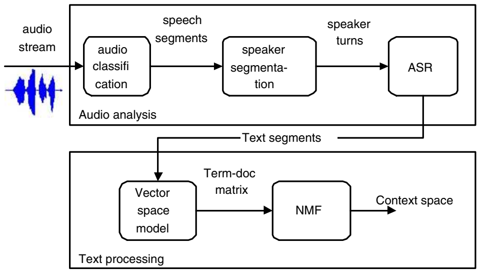


Fig. 1. The system setup. The audio stream is first processed using audio segmentation. Segments are then using an automatic speech recognition (ASR) system to produce text segments. The text is then processed using a vector representation of text and apply non-negative matrix factorization (NMF) to find a topic space.

2. SYSTEM OVERVIEW

The system presented here operates in two domains combining audio based processing and text based topic detection. The overall system pipeline is shown in figure 1. The audio segmentation part was described recently in [2] and will only be briefly outlined here. The text processing step will be covered in section 3.

2.1. Audio Segmentation

The basic sound representation is a feature set consisting on 12-dimensional MFCC coefficients as well as MPEG type features such as the zero-crossing rate, short-time energy, and spectral flux. The first step is to separate speech parts, excluding 'jingles'. The step is performed in a supervised classification step using a trained linear classifier. The classifier operates on 1 sec windows and detects the two classes speech and music. The audio classification step was shown to have a correct classification rate of 97.8%, see [6] for additional details and feature set evaluation.

To aid topic spotting and locate story boundaries in the news stream we use additional audio cues to segment the audio into sub-segments containing only one speaker in a given environment. Broadcast news shows typically have unknown number and identities of speakers, including news anchors and guests appearing both in the radio studio and in the field. Therefore we invoke an unsupervised speaker change detection algorithm [2]. The algorithm is based on a 12-dimensional MFCC feature set, that are statistically summarized by vector quantization in sliding windows on both sides of a hypothesized change-point. The similarity of these windows is then measured using the vector quantization distortion (VQD) and possible speaker changes are detected by simple thresholding. A second step is invoked for removal of false alarms inferred by the change detection algorithm. This step uses larger windows around a hypothesized change-point to yield more precise statistically models. An overall F-measure of 0.85 is found using this algorithm [6].

2.2. Automatic Speech Recognition

Speech-to-text transcriptions are based on the Sphinx4[7] speech recognition system. Sphinx4 is a large vocabulary speaker independent open source speech recognition system from Carnegie Mellon University. Sphinx4 is capable of a word accuracy of $50 - 80\%$ depending on the speaking style of the speaker and the background noise level.

3. TOPIC DETECTION

By audio segmentation and subsequent speech recognition we obtain a speaker document database. The database can be approached using standard indexing methods. To further increase the user friendliness we propose to use text modeling tools to spot relevant contexts of the documents. Probabilistic latent semantic analysis has shown very powerful for high-level statistical modeling of text [8]. PLSI was defined as a conditional model, which complicates generalization to news documents. We note that by modeling the joint probability the topic detection problem can be solved by non-negative matrix factorization (NMF). Xu et al. [4] have already shown that NMF outperforms SVD- and eigen decomposition clustering methods, see also [5].

3.1. Document Representation

Given the speaker documents $D = \{d_1, d_2, ..., d_m\}$ and the vocabulary set $T = \{t_1, t_2, ..., t_n\}$, the $n \times m$ term-document matrix $\mathbf{X}$ is created, where $\mathbf{X}_{i,j}$ is the count of term t_i in the speaker document d_j.

NMF factorizes the non-negative $n \times m$ matrix $\mathbf{X}$ into the non-negative $n \times r$ matrix $\mathbf{W}$ and the non-negative $r \times m$ matrix $\mathbf{H}$. This is done to minimize the objective function $J = \frac{1}{2}\|\mathbf{X} - \mathbf{WH}\|$, where $\| \cdot \|$ denotes the squared sum of the elements of the matrix.

The columns of $\mathbf{W}$ forms a r-dimensional semantic space, where each column can be interpreted as a context vocabulary of the given document corpus. Each document, columns of $\mathbf{H}$, is hence formed as a linear sum of the contexts. Usually r is chosen to be smaller than n and m.

3.2. NMF for Document Retrieval

Let N be the sum of all elements in $\mathbf{X}$. Then $\widetilde{\mathbf{X}} = \frac{\mathbf{X}}{N}$ form a frequency table approximating the joint probability of terms t and documents d

$$\widetilde{\mathbf{X}} \equiv \frac{\mathbf{X}}{N} \approx p(t, d). \tag{1}$$

Expanding on a complete set of disjoint contexts k we can write $p(t, d)$ as the mixture

$$p(t, d) = \sum_{k=1}^{K} p(t|k)p(d|k)p(k), \tag{2}$$

where $p(t|k)$ and $p(d|k)$ are identified as $\mathbf{W}$ and $\mathbf{H}$ of the NMF respectively, if columns of $\mathbf{W}$ and rows of $\mathbf{H}$ are normalized as probability distributions

$$\begin{aligned} p(t, d) &= \sum_{k=1}^{K} \mathbf{W}_{t,k}\mathbf{H}_{k,d} && (3) \\ &= \sum_{k=1}^{K} \frac{\mathbf{W}_{t,k}}{\alpha_k}\frac{\mathbf{H}_{k,d}}{\beta_k}\alpha_k\beta_k, && (4) \end{aligned}$$

where $\alpha_k = \sum_t \mathbf{W}_{t,k}$, $\beta_k = \sum_d \mathbf{H}_{k,d}$, and $p(k) = \alpha_k\beta_k$. Thus, the normalized $\mathbf{W}$ is the probability of term t given a context k, while the normalized $\mathbf{H}$ is the probability of document d in a context k. $p(k)$ can be interpreted as the prior probability of context k.

The relevance (probability) of context k given a query string d^* is estimated as

$$p(k|d^*) = \sum_t p(k|t)p(t|d^*), \tag{5}$$

where $p(t|d^*)$ is the normalized histogram of (known) terms in the query string d^*, while $p(k|t)$ is found using Bayes theorem using the quantities estimated by the NMF step

$$\begin{aligned} p(k|t) &= \frac{p(t|k)p(k)}{\sum_{k'} p(k|t')p(k')} && (6) \\ &= \frac{\mathbf{W}_{t,k}p(k)}{\sum_{k'} \mathbf{W}_{t,k'}p(k')}. && (7) \end{aligned}$$

The relevance (probability) of document d given a query d^* is then

$$\begin{aligned} p(d|d^*) &= \sum_{k=1}^{K} p(d|k)p(k|d^*) && (8) \\ &= \sum_{k=1}^{K} \mathbf{H}_{k,d}p(k|d^*). && (9) \end{aligned}$$

The relevance is used for ranking in retrieval. Importantly we note that high relevance documents need not contain any of the search terms present in the query string. If the query string invokes a given subset of contexts the most central documents for these context are retrieved. Thus, the NMF based retrieval mechanism acts as a kind of association engine: "These are documents that are highly relevant for your query".

4. EVALUATION

In the following section we evaluate the use of NMF for topic detection and document retrieval.

To form a database 2099 CNN-News podcast shows have been automatically transcribed and segmented into 30977 speaker documents, yielding a vocabulary set of 37791 words after stop-word removal. The news show database was acquired during the period 2006-04-04 to 2006-08-09.

Based on the the database a term-document matrix was created and subjected to NMF decomposition using $K = 70$ contexts producing matrices $\mathbf{W}$ and $\mathbf{H}$. The implementation of the NMF-algorithm was done using the approach of [9]. For each context the ten most probable terms in the corresponding column of $\mathbf{W}$ were extracted as keywords. Based on the keyword list each context was manually labeled with a short descriptive text string (one-two words).

Label	No. segments
Crisis in Lebanon	8
War in Iraq	7
Heatwave	7
Crime	5
Wildfires	1
Hurricane season	2
Other	30
Total	60

Table 1. The specific contexts used for evaluation by manual topic delineation.

> ... california governor arnold's *fortson agar* inspected the california mexico border by helicopter wednesday to see ...
>
> ... the past days president bush asking california's governor for fifteen hundred more national guard troops to help patrol the mexican border but governor orville *schwartz wicker* denying the request saying...

Fig. 2. Two examples of the retrieved text for a query on 'schwarzenegger'.

For evaluation of the topic detector eight CNN-News shows were manually segmented and labeled in a subset of six contexts out of the $K = 70$ contexts identified by NMF. Segments that were not found to fall into any of six topics were labeled as 'other'. The six labels and the number of segments for each label can be seen in table 1.

4.1. NMF for Query Expansion and Segmentation

As described above the probabilistic interpretation of the NMF factorization allows query expansion by 'association'.

To illustrate the usefulness of this system let us consider a specific example. We query the system with the single term 'schwarzenegger', the governor of the state of California in USA.

The query expansion first uses eq. (5) to evaluate probabilities of the contexts given the query string. The result of the 'schwarzenegger' query produces the following three most probable contexts that were hand-labeled from the automaticly generated keyword list:

- 'California Politics' $p(k|d^*) = 0.38$
- 'Mexico border' $p(k|d^*) = 0.32$
- 'Politics' $p(k|d^*) = 0.17$

Illustrating that the system indeed is able to find associate relevant topics from broadcasts in the database, which consists of data from the summer of 2006.

The retrieval of documents using our method is based on evaluating eq. (8), returning documents with the highest relevance. The results of traditional text indexing returns documents including the exact term 'schwarzenegger', which might be sufficient. However, the advantage as mentioned, by expanding the query on 'topic space', allow the system to return documents that do not directly include the word 'schwarzenegger' but are related through the topics. This is indeed happening in two of the top ten documents returned from our database, which are shown in figure 2. The problem here is that the automatically generated transcription was imperfect, missing to detect the word 'schwarzenegger'. Thus we here see an example of the top-down healing of mistakes made by speech-to-text transcription by using a global topic representation.

To perform a more quantitative evaluation we use eq. (5) to calculate the posterior probabilities of each context given short query strings d^*. This can be used to segment a news cast into homogenous topic segments. In particular we treat a short sequence of ten words as a query string, and associate topics to these queries based on the probability $p(k|d^*)$. 'Sliding' d^* along the news cast, topic changes can thereby be found when $\arg\max_k p(k|d^*)$ changes.

For evaluating the segmentation task we use the recall (RCL) and precision (PRC) measures defined as:

$$\text{RCL} = \frac{\textit{no. of correctly found change-points}}{\textit{no. of manually found change-points}} \tag{10}$$

$$\text{PRC} = \frac{\textit{no. of correctly found change-points}}{\textit{no. of estimated change-points}}, \tag{11}$$

where an estimated change-point is defined as correct if a manually found change-point is located within ±5 words.

In the test we concatenated the eight manually segmented news shows and removed stop-words. Running the topic detection algorithm resulted in a recall of 0.88 with a precision of 0.44. It shows that almost every manually found change-points are found by our algorithm. On the other hand the method infers a number of false alarms. This is mostly due to that some of the manually found segments are quite long and include subsegments. For instance some of the segments about 'crisis in Lebanon' contains segments where the speaker is speaking about the US Secretary of State Condoleezza Rice's relationship to the crisis. These subsegments have a larger probability with other 'US politic' contexts, so the system will infer a change-point, i.e., infer an off-topic association induced by a single prominent name. If such events are unwanted, we probably need to go beyond mere probabilistic arguments, hence, invoke a 'loss' matrix penalizing certain association types.

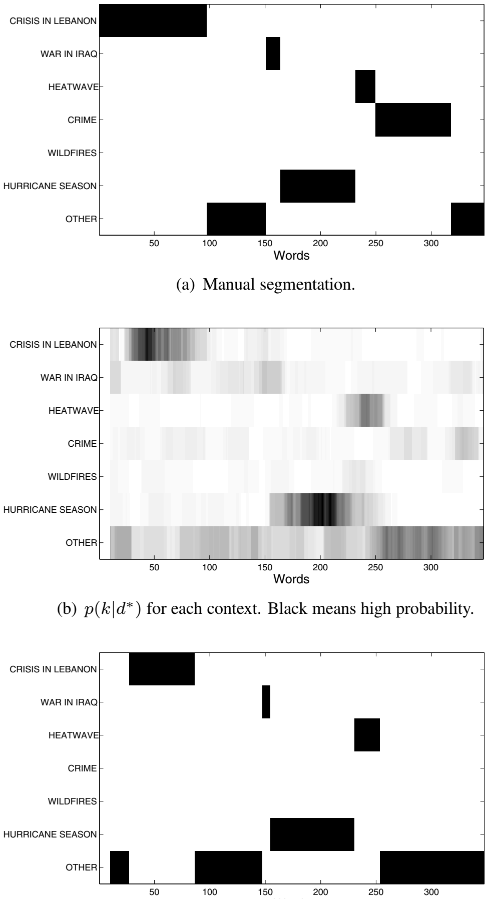
Figure 3 shows an example of the segmentation process for one of the news shows. Figure 3(a) shows the manual segmentation, while figure 3(b) shows the $p(k|d^*)$ distribution forming the basis of figure 3(c). The NMF-segmentation is consistent with the manual segmentation, with exceptions, such as a segment which is manually segmented as 'crime' but missed by the NMF-segmentation.

4.2. Topic Classification

The segmentation procedure described above provides labels for all instances of d^*. As in [4, 5] we use the accuracy (AC), defined as $\text{AC} = \frac{1}{n}\sum_{i=1}^{n}\delta\big(c_m(i), c_s(i)\big)$ to quantitatively evaluate the classification performance. $\delta(x, y)$ is 1 if $x = y$, 0 otherwise. c_m and c_s denotes the manually and system labels respectively and n is the number of test strings. Using the same data as in the segmentation task we achieve an overall AC of 0.65.

The confusion matrix for the experiment is shown in table 2. The table shows that most of the errors are when the system are classifying c1, c2, c3, and c4 as c7 ('other'). The system here output the class 'other' when none of the six selected topic classes has the highest relevance $p(k|d^*)$.

In the above classification the query string d^* is based on a sequence of ten words. If instead we use the 60 manually found segments as query string we are able to detect 53 correctly, which gives an accuracy of $\text{AC} = 0.88$.



(a) Manual segmentation.

(b) $p(k|d^*)$ for each context. Black means high probability.

(c) The segmentation based on $p(k|d^*)$.

Fig. 3. Figure 3(a) shows the manual segmentation of the news show into 7 classes. Figure 3(b) shows the distribution $p(k|d^*)$ used to do the actual segmentation shown in figure 3(c). The NMF-segmentation is in general consistent with the manual segmentation. Though, the segment that is manually segmented as 'crime' is labeled 'other' by the NMF-segmentation

5. CONCLUSION

We have presented a system capable of retrieving relevant segments of audio broadcast news. The system uses audio processing to segment the audio stream into speech segments. A speech-to-text system is utilized to generate transcriptions of the speech. Furthermore a strategy for application of non-negative matrix factorization of joint probability tables to retrieve relevant spoken documents in a broadcast news database. We have demonstrated retrieval of documents in which query terms did not appear because of transcription errors, hence shown that global topic models can assist interpretation of speech-to-text data. The system is fully implemented as a web demo available from the URL: http://castsearch.imm.dtu.dk.

	c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**
c1	**370**	32	10	0	8	0	380
c2	0	**131**	1	2	0	8	52
c3	0	0	**105**	7	0	8	88
c4	0	16	2	**9**	0	0	112
c5	0	0	0	0	**13**	0	0
c6	0	0	29	0	0	**88**	0
c7	3	29	8	0	6	0	**759**

Table 2. Classification confusion matrix, where rows are manual labels and columns are estimated labels. The used classes are: (c1) crisis in Lebanon, (c2) war in Iraq, (c3) heatwave, (c4) crime, (c5) wildfires, (c6) hurricane season, and (c7) other.

ACKNOWLEDGEMENTS

This work is supported by the Danish Technical Research Council, through the framework project 'Intelligent Sound',
www.intelligentsound.org (STVF No. 26-04-0092).

6. REFERENCES

[1] John H. L. Hansen, Rongqing Huang, Bowen Zhou, Michael Seadle, J. R. Deller, Aparna R. Gurijala, Mikko Kurimo, and Pongtep Angkititrakul, "Speechfind: Advances in spoken document retrieval for a national gallery of the spoken word," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 712–730, september 2005.

[2] K. W. Jørgensen, L. L. Mølgaard, and L. K. Hansen, "Unsupervised speaker change detection for broadcast news segmentation," in *Proc. EUSIPCO*, 2006.

[3] James Allan, Ed., *Topic Detection and Tracking: Event-Based Information Organization*, Kluwer Academic Publishers, Norwell, MA, 2002.

[4] Wei Xu, Xin Liu, and Yihong Gong, "Document clustering based on non-negative matrix factorization," in *Proc. of the 26th International ACM SIGIR*, New York, NY, USA, 2003, pp. 267–273, ACM Press.

[5] Farial Shahnaz, Michael W. Berry, V. Paul Pauca, and Robert J. Plemmons, "Document clustering using nonnegative matrix factorization," *Inf. Process. Manage.*, vol. 42, no. 2, pp. 373–386, 2006.

[6] K. W. Jørgensen and L. L. Mølgaard, "Tools for automatic audio indexing," M.S. thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 2006.

[7] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rite Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," Tech Report TR-2004-127, Sun Microsystems, 2004.

[8] Thomas Hofmann, "Probabilistic Latent Semantic Indexing," in *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, Berkeley, California, August 1999, pp. 50–57.

[9] Chih-Jen Lin, "Projected gradient methods for non-negative matrix factorization," Tech. Rep., Department of Computer Science, National Taiwan University, 2005.

Appendix C

Temporal analysis of text data using latent variable models

Submitted to Machine Learning for Signal Processing

TEMPORAL ANALYSIS OF TEXT DATA USING LATENT VARIABLE MODELS

Lasse L. Mølgaard[*], *Jan Larsen*

Section for Cognitive Systems
DTU Informatics
DK-2800 Kgs. Lyngby, Denmark
{llm, jl}@imm.dtu.dk

Cyril Goutte

Interactive Language Technologies
National Research Council of Canada
Gatineau, Canada, QC J8X 3X7
Cyril.Goutte@cnrc-nrc.gc.ca

ABSTRACT

Detecting and tracking of temporal data is an important Task in multiple applications. In this paper we study temporal text mining methods for Music Information Retrieval. We compare two ways of detecting the temporal latent semantics of a corpus extracted from Wikipedia, using a stepwise PLSA approach and a global multiway PLSA method. The analysis indicates that the global analysis method is able to identify relevant trends which are difficult to get using a step-by-step approach.

1. INTRODUCTION

Music Information Retrieval (MIR) is a multifaceted field, which until recently mostly focused on audio analysis. The use of textual descriptions, beyond using genres, has grown in popularity with the advent of different music websites, e.g. ”Myspace.com” or ”The Hype Machine”[1], where abundant data about music has become easily available. This has for instance been investigated in [1], where textual descriptions of music were retrieved World Wide Web to find similarity of artists. The question of how to use textual data in a useful way for music information is still an open one. Retrieving unstructured data using general web crawling produces a lot of data that must be cleaned to focus on web pages and terms that actually describe musical artists and concepts. Community-based music web services such as tagging based systems, e.g. Last.fm, have also shown to be a good basis for extracting latent semantics of musical track descriptions [2]. The text-based methods have so far only considered text data without any structured knowledge. In this study we investigate if the incorporation of time information in latent semantic factor helps to enhance the detection and description of topics.

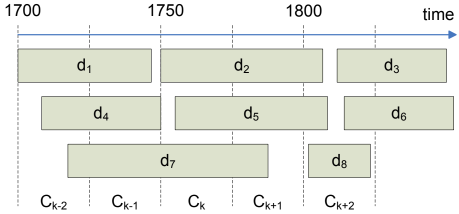


Fig. 1. An example of assigning a collection of documents d_i based on the time intervals the documents belong to. The assignment produces a document collection C_k for each time interval

Tensor methods in the context of text mining have recently received some attention using higher-order decomposition methods such as the PARAllel FACtors model [3], that can be seen as a generalization of Singular Value Decomposition in higher dimensional arrays. The article [3] applies tensor decomposition methods successfully for topic detection in e-mail correspondence over a 12 month period. The article also employs a non-negatively constrained PARAFAC model forming a Nonnegative Tensor Factorization analogous to the well-known Nonnegative Matrix Factorization (NMF) [4].

NMF and Probabilistic Latent Semantic Analysis (PLSA) [5] have successfully been applied in many text analysis tasks to find interpretable latent factors. The two methods have been shown to be equivalent [6], where PLSA has the advantage of providing a direct probabilistic interpretation of the latent factors.

2. TEMPORAL TOPIC DETECTION

Detecting latent factors or topics in text using NMF and PLSA has assumed an unstructured and static collection of documents.

Extracting topics from a temporally changing text col-

[*] First author performed the work while visiting NRC Interactive Language Technologies group.

[1] hypem.com

lection has received some attention lately, for instance by [7] and also touched by [8]. These works investigate text streams that contain documents that can be assigned a timestamp y. The timestamp may for instance be the time a news story was released, or in the case of articles describing artists it can be a timespan indicating the active years of the artist. Finding the evolution of topics over time requires assigning documents $d_1, d_2, ..., d_m$ in the collection to time intervals $y_1, y_2, ..., y_l$, as illustrated in figure 1. In contrast to the temporal topic detection approach in [7], we can assign documents to multiple time intervals, e.g. if the active years of an artist spans more than one of the chosen time intervals. The assignment of documents then provides l sub-collections $C_1, C_2, ..., C_l$ of documents.

The next step is to extract topics and track their evolution over time.

2.1. Stepwise temporal PLSA

The approaches to temporal topic detection presented in [7] and [8] employ latent factor methods to extract distinct topics for each time interval, and then compare the found topics at succeeding time intervals to link the topics over time to form temporal topics.

We extract topics from each sub-collection C_k using a PLSA-model [5]. The model assumes that documents are represented as a bags-of-words where each document d_i is represented by an n-dimensional vector of counts of the terms in the vocabulary, forming an $n \times m$ term by document matrix for each sub-collection C_k. PLSA is defined as a latent topic model, where documents and terms are assumed independent conditionally over topics z:

$$P(t,d)_k = \sum_{z}^{Z} P(t|z)_k P(d|z)_k P(z)_k \qquad (1)$$

This model can be estimated using the Expectation Maximization (EM) algorithm, cf. [5].

Having found a topic model for each document sub-collection C_k with parameters, $\{P(t|z)_k, P(d|z)_k, P(z)_k\}$, these topics need to be stringed together with the topics for the next time span $k+1$. The comparison of topics is done by comparing the term profiles $P(t|z)_k$ for the topics found in the PLSA model. The similarity of two profiles is naturally measured using the KL-divergence,

$$D(k+1||k) = \sum_t p(t|z)_{k+1} \log \frac{p(t|z)_{k+1}}{p(t|z)_k}. \qquad (2)$$

Determining whether a topic is continued in the next time span is quite simply chosen based on a threshold λ, such that two topics are linked if $D(k+1||k)$ is smaller than a fixed threshold λ. The KL-divergence is asymmetric but $D(k+1||k)$ makes more sense to use than $D(k||k+1)$ [7]. The choice of the threshold must be tuned to find the temporal links that are relevant.

2.2. Multiway PLSA

The method presented above is useful to some extent, but does not fully utilize the time information that is contained in the data. Some approaches have used the temporal aspect, e.g. [9] where an incrementally trainable NMF-model is used to detect topics. This approach does include some of the temporal knowledge but still lacks global view of the important topics viewed over the whole corpus of texts.

Using multiway models, also called tensor methods we can model the topics directly over time. The 2-way PLSA model in 1 can be extended to a 3-way model by also conditioning the topics over years y, as follows:

$$P(t,d,y) = \sum_z P(t|z)P(d|z)P(y|z)P(z) \qquad (3)$$

The model parameters are estimated through Maximum likelihood using the EM-algorithm, e.g. as in [10]. The expectation step evaluates $P(z|t,d,y)$ using the estimated parameters at step t.

$$\text{(E-step):} \qquad P(z|t,d,y) = \frac{p(t|z)p(d|z)p(y|z)p(z)}{\sum_{z'} p(t|z')p(d|z')p(y|z')p(z')} \qquad (4)$$

The subsequent M-step then updates the parameter estimates.

$$\text{(M-step):} \qquad P(z) = \frac{1}{N} \sum_{tdy} x_{tdy} P(z|t,d,y) \qquad (5)$$

$$P(t|z) = \frac{\sum_{dy} x_{tdy} P(z|t,d,y)}{\sum_{tdy} x_{tdy} P(z|t,d,y)} \qquad (6)$$

$$P(d|z) = \frac{\sum_{ty} x_{tdy} P(z|t,d,y)}{\sum_{tdy} x_{tdy} P(z|t,d,y)} \qquad (7)$$

$$P(y|z) = \frac{\sum_{td} x_{tdy} P(z|t,d,y)}{\sum_{tdy} x_{tdy} P(z|t,d,y)} \qquad (8)$$

The EM algorithm is guaranteed to converge to a local maximum of the likelihood. The EM algorithm is sensitive to initial conditions, so a number of methods to stabilize the estimation have been devised, e.g. Deterministic Annealing [5]. We have not employed these but instead rely on restarting the training procedure a number of times to find a good solution.

2.3. Topic model interpretation

The latent factors z of the model can be seen as topics that are present in the data. The parameters of each topic can be used as descriptions of the topic. $P(t|z)$ represents the probabilities of the terms for the topic z, thus providing a way to find words that are representative of the topic. The most straightforward method to find these keywords is to use the words with the highest probability $P(t|z)$. This approach unfortunately is somewhat flawed as the histogram reflects the overall frequency of words, which means that generally common words tend to dominate the $P(t|z)$.

This effect can be neutralized by measuring the relevance of words in a topic relative to the probability in the other topics. Measuring the difference between the histograms for each topic can be measured by use of the symmetrized Kullback-Leibler divergence:

$$KL(z, \neg z) = \sum_t \underbrace{(P(t|z) - P(t|\neg z)) \log \frac{P(t|z)}{P(t|\neg z)}}_{w_t} \qquad (9)$$

This quantity is a sum of contributions from each term t, w_t. The terms that contribute with a large value of w_t are those that are relatively more special for the topic z. w_t can thus be used to choose the keywords. The keywords should be chosen from the terms that have a positive value of $P(t|z) - P(t|\neg z)$ and with the largest w_t.

3. WIKIPEDIA DATA

In this experiment we investigated the description of composers in Wikipedia. This should provide us with a dataset that spans a number of years, and provides a wide range of topics. We performed the analysis on the Wikipedia data dump saved 27th of July 2008, retrieving all documents that Wikipedians assigned to composer categories such as "Baroque composers" and "American composers". This produced a collection of 7358 documents, that were parsed so that only the running text was kept.

Initial investigations in music information web mining showed that artist names can heavily bias the results. Therefore words occurring in titles of documents, e.g. *Wolfgang Amadeus Mozart*, are removed from the text corpus, i.e. all occurrences of the terms 'mozart', 'wolfgang' and 'amadeus' have been removed from all documents in the corpus. Furthermore we removed semantically irrelevant stopwords based on a stoplist of 551 words. Finally terms that occurred fewer than 3 times counted over the whole dataset and terms not occurring in at least 3 different documents were removed.

The document collection was then represented using a bag-of-features representation forming a term-document matrix $\mathbf{X}$ where each element x_{td} represents the count of term t in document d. The vector $\mathbf{x}_d$ thus represents the term histogram for document d.

To place the documents temporally the documents were parsed to find the birth and death dates. These data are supplied in Wikipedia as documents are assigned to categories such as "1928 births" and "2007 deaths". The dataset contains active composers from around 1500 until today. The next step was then to choose the time spans to use. Inspection of the data revealed that the number of composers before 1900 is quite limited so the artists were assigned to time intervals of 25 years, giving a first time interval of [1501-1525]. After 1900 the time intervals were set to 10 years, for instance [1901-1910]. Composers were assigned to time intervals if they were alive in some of the years. We estimated the years composers were active by removing the first 20 years of their lifetime. The resulting distribution of documents on the resulting 27 time intervals is seen in figure 2.

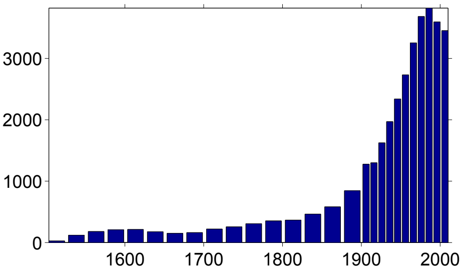


Fig. 2. Number of composer documents assigned to each of the chosen time spans.

The term by document matrix was extended with the time information by assigning the term-vector for each composer document to each era, thus forming a 3-way tensor containing terms × documents × years. The tensor was further normalized over years, such that the weight of the document summed over years is the same as in the initial term doc-matrix. I.e. $P(d) = \sum_{t,y} X_{tdy} = \sum_t X_{td}$. This was done to avoid long-lived composers dominating the resulting topics.

The resulting tensor $\mathbf{X} \in \mathbb{R}^{m \times n \times l}$ contains 18536 terms x 7358 documents x 27 time slots with 4,038,752 non-zero entries (0.11% non-zero entries).

3.1. Term weighting

The performance of machine learning approaches in text mining often depends very heavily on the preprocessing steps that are taken. Term weighting for LSA-like methods and Non-negative Matrix factorization have thus shown to be paramount in getting interpretable results. We applied the well-known $TF{\cdot}IDF$ weighting scheme, using $TF = \log(1+x_{tdy})$ and the log-entropy weighting variant of document weighting, $IDF = 1 + \sum_{d=1}^{D} \frac{h_{td} \log h_{td}}{logD}$, where $h_t d = \frac{\sum_y x_{tdy}}{\sum_{dy} x_{tdy}}$. The log local weighting minimizes the effect of

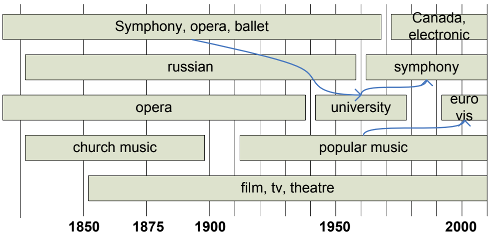


Fig. 3. Topics detected using step-by-step PLSA. The topics are depicted as connected boxes, but are the results of the KL-divergence-based linking between time slots

very frequent words, while the entropy global weight tries to discriminate important terms from common ones. The documents in Wikipedia differ quite a lot in length, therefore we employ document normalization to avoid that long articles dominate the modeled topics.

4. EXPERIMENTS

We performed experiments on the Wikipedia composer data using the stepwise temporal PLSA method and the multiway-PLSA methods.

4.1. Stepwise temporal PLSA

The step-by-step method was employed training 5 and 16 component models for each of the sub-collections of documents described above. The number of components was chosen to have one model that describes more general topics and a model with a higher number of components that can detect more specific topics. Using the higher number of topics makes it possible to leave out some of the less salient topics at each time step. The PLSA models for each time span was trained using a stopping criterion of 10^{-5} relative change of the cost function. We restarted the training a number of times for each model choosing the model minimizing the likelihood. The topics extracted were then coupled together over time, picking topics that have a KL-divergence between the topic term distributions, $D(\theta_{y+1}|\theta_y)$, below the threshold λ. This choice of threshold produces a number of topics that stretch over several time spans. Tuning this threshold to identify the relevant links between topics in different time spans, makes this model somewhat difficult to tune. A low setting for λ may thus leave out some of the more subtle relations, while a higher setting produces too many links to be interpretable. Figure 3 shows the topics found for the 20th century using the 5 component models. There are clearly 4 topics that are present throughout the whole period. The topics are film and TV music composers, which in the beginning contains Broadway/theater composers. The other dominant topic describes hit music composers. Quite interestingly this topic forks off a topic describing Eurovision song contest composers in the last decades.

Even though the descriptions of artists in Wikipedia contain a lot of bibliographical information it seems that the latent topics have musically meaningful keywords. As there was no use of a special vocabulary in the preprocessing phase, it is not obvious that these musically relevant phrases would be found.

The stepwise temporal PLSA approach has two basic shortcomings. The first is the problem of adjusting the threshold λ to find the meaningful topics over time. The other one is to choose the number of components to use in the PLSA in each time span. The 5 topics that are used above do give some interpretable latent topics in the last decade as shown in figure 3. On the other hand the results for the earlier time spans that contain less data, means that the PLSA model finds some quite specific topics at these time spans. As an example the period 1626-1650 has the following topics[2]:

1626-1650				
41%	34%	15%	8.7%	1.4%
keyboard	madrigal	viol	baroque	anglican
organ	baroque	consort	italy	liturgi
surviv	motet	lute	poppea	prayer
italy	continuo	england	italian	respons
church	monodi	charles	lincoronazion	durham
nuremberg	renaissance	royalist	opera	english
choral	venetian	masqu	finta	chiefli
baroque	style	fretwork	era	england
germani	cappella	charles's	venice	church
collect	itali	court	teatro	choral

The topics found here are quite meaningful in describing the baroque period, as the first topic describes the church music, and the second seems to find the musical styles, such as madrigals and motets. The last topic on the other hand only has a topic weight of $P(z) = 1.4\%$. This tendency was even more distinct when using 16 components in each time span.

4.2. Multi-way PLSA

Modeling using the MWPLSA model was also performed on the tensor described in section 3. Analogously with the stepwise temporal PLSA model we stopped training reaching a change of less than 10^{-5} of the cost function. The main advantage of the MWPLSA method is that the temporal linking of topics is accomplished directly through the model estimation. This flexibility allows us to find temporal topics by fitting models using different numbers of components to find the inherent structure of the data. The time evolution of topics can be visualized using the parameter $P(y|z)$ that gives the weight of a component for each

[2] The keywords for all topics are found at http://imm.dtu.dk/ llm/mlsp/

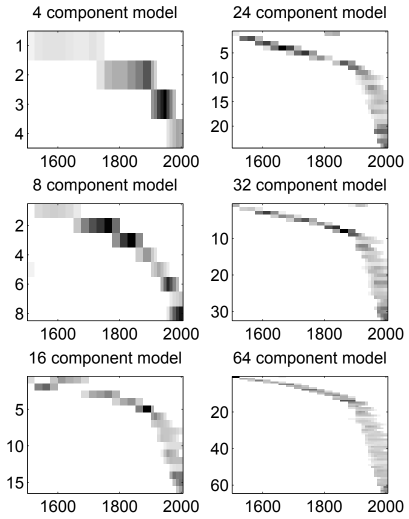


Fig. 4. Time view of components extracted using MW-PLSA, showing the time profiles $P(y|z)$ as a heatmap. A dark color corresponds higher values of $P(y|z)$

time span. Figure 4 shows the result for 4, 8, 16, 32 and 64 components as a "heatmap", where darker colors correspond to higher values of $P(y|z)$. The topics are generally unimodally distributed over time so the model only finds topics that increase in relevance and then dies off without reappearing. The skewed distribution of documents over time which we described earlier emerges clearly in the distribution of topics, as most of the topics are placed in the last century. Adding more topics to the model has two effects considering the plots. Firstly the topics seem to be sparser in time, finding topics that span fewer years. Using more topics decomposes the last century into more topics that must be semantically different as they almost all span the same years. Inspection of the different topics can show how meaningful the topics are. The keywords extracted using the method mentioned above are shown in table 1 for some of the topics extracted by the 32 component model, including the time spans that they belong to.

The first topic shown in table 1 is one of the two topics that accounts for the years 1626-1650, the keywords summarize the five topics found using the multiway PLSA. The last four topics are some of the 24 topics that describe the last century. The second topic has the keywords ragtime and rag, placed in the years 1876-1921, which aligns remarkably well with the description of the genre on Wikipedia: *"Ragtime [...] is an originally American musical genre which enjoyed its peak popularity between 1897 and 1918."*[3]. Comparing to the stepwise temporal PLSA approach ragtime also appeared as keywords in the 16 component model, appearing as a topic from 1901-1920. The next topic seems to describe World War II, but also contains the neoclassical movement in classical music. The 16 component stepwise temporal PLSA approach finds a number of topics from 1921-1940 that describe the war, such as a topic in 1921-1930 with keywords: *war, time, year, life, influence* and two topics in 1931-1941, *1. time, war, year, life, style* and *2: theresienstadt, camp, auschwitz, deport, concentration, nazi*. These are quite unrelated to music, so it is evident that the global view of topics employed in the mwplsa-model identifies neoclassicism to be the important keywords compared to topics from other time spans.

1601-1700 2.10%	1876-1921 2.40%	1921-1981 4.70%	1921-1981 4.80%	1971-2008 6.40%
baroque	ragtime	concerto	broadway	single
continuo	sheet	nazi	hit	chart
motet	rag	war	songwriter	album
viol	weltemignon	symphony	film	release
survive	nunc	piano	mgm	hit
basso	ysa	ballet	vaudeville	track
italian	schottisch	neoclassical	lyricist	sold
court	dimitti	neoclassic	benni	demo
church	blanch	choir	bing	fan
cathedral	parri	hochschul	inter	pop

Table 1. Keywords for 5 of 32 components in a MWPLSA model. The assignment of years is given from $P(y|z)$ and percentages placed at each column are the corresponding component weights, $P(z)$

5. MULTI-RESOLUTION TOPICS

The use of different number of components in the multiway PLSA model, as seen in figure 4, shows that the addition of topics to the model shrinks the number of years they span. The higher specificity of the topics when using more components gives a possibility to "zoom" in on interesting topic, while the low complexity models can provide the long lines in the data.

To illustrate how the clusters are related as we add topics to the model, we can generate a so-called clusterbush, as proposed in [11]. The result for the MWPLSA-based clustering is shown in figure 5. The clusters are sorted such that the clusters placed earliest in time are placed left. It is evident the clusters related to composers from the earlier centuries form small clusters that are very stable, while the later components are somewhat more ambiguous. The clusterbush could therefore be good tool for inspecting the topics at different timespans to get an estimate of the number of components needed to describe the data.

[3] http://en.wikipedia.org/wiki/Ragtime

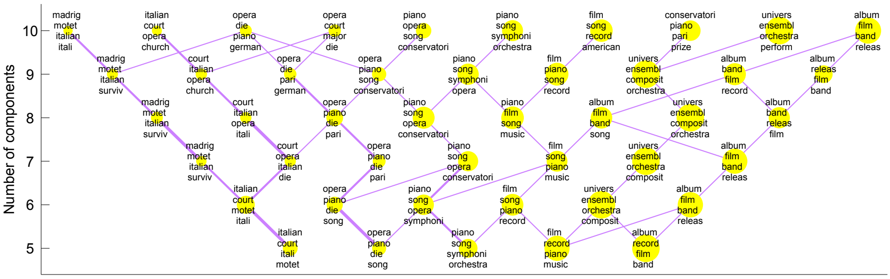


Fig. 5. ”Cluster bush” visualisation of the results of the MWPLSA clustering of composers. The size of the circles correspond to the weight of the cluster, and the thickness of the line between circles how related the clusters are. Each cluster is represented by the keywords and is placed according to time from left to right.

6. DISCUSSION

The multiway PLSA shows some definite advantages in finding topics that are temporally defined. The stepwise temporal PLSA approach is quite fast to train and processing for each time span can readily be processed in parallel. But it has the practical drawback that it requires a manual tuning of the linking threshold, and the lack of a global view of time in the training of PLSA models misses some topics as shown above. The training of multiway PLSA is somewhat slower than the step-by-step approach but the more flexible representation that the model gives is a definite advantage, for instance when data has a skewed distribution as in the work resented here. The global model would also make it possible to do model selection over all time steps directly.

7. CONCLUSION

We have investigated the use of time information in analysis musical text in Wikipedia. It was shown that the use of time information produces meaningful latent topics, which are not readily extractable from this text collection without any prior knowledge. The multiway PLSA was shown to provide a more flexible and compact representation of the temporal data than the step-by-step stepwise temporal PLSA method. The use of Wikipedia data also seems to be a very useful resource for semi-structured data for Music Information Retrieval that could be investigated further to harness the full potential of the data.

8. REFERENCES

[1] P. Knees, E. Pampalk, and G. Widmer, “Artist classification with web-based data,” in *Proceedings of ISMIR*, Barcelona, Spain, 2004.

[2] M. Levy and M. Sandler, “Learning latent semantic models for music from social tags,” *Journal of New Music Research*, vol. 37, no. 2, pp. 137–150, 2008.

[3] B. Bader, M. Berry, and M. Browne, *Survey of Text Mining II Clustering, Classification, and Retrieval*, chapter Discussion tracking in Enron email using PARAFAC, pp. 147–163, Springer, 2008.

[4] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[5] T. Hofmann, “Probabilistic Latent Semantic Indexing,” in *Proc. 22nd Annual ACM Conf. on Research and Development in Information Retrieval*, Berkeley, California, August 1999, pp. 50–57.

[6] E. Gaussier and C. Goutte, “Relation between plsa and nmf and implications,” in *Proc. 28th annual ACM SIGIR conference*, New York, NY, USA, 2005, pp. 601–602, ACM.

[7] Q. Mei and C. Zhai, “Discovering evolutionary theme patterns from text: an exploration of temporal text mining,” in *Proc. of KDD 05*. 2005, pp. 198–207, ACM Press.

[8] M. W. Berry and M. Brown, “Email surveillance using non-negative matrix factorization,” *Computational and Mathematical Organization Theory*, vol. 11, pp. 249–264, 2005.

[9] B. Cao, D. Shen, J-T Sun, X. Wang, Q. Yang, and Z. Chen, “Detect and track latent factors with online nonnegative matrix factorization,” in *Proc. of IJCAI-07*, Hyderabad, India, 2007, pp. 2689–2694.

[10] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 435–447, Feb. 2008.

[11] F. Å. Nielsen, D. Balslev, and L. K. Hansen, “Mining the posterior cingulate: Segregation between memory and pain components,” *NeuroImage*, vol. 27, no. 3, pp. 520–532, jun 2005.

Bibliography

[1] Cmu informedia digital video library project. http://www. informedia.cs.cmu.edu/.

[2] The lemur toolkit. web: http://www.lemurproject.org/.

[3] Lucene indexing engine. web, http://lucene.apache.org/.

[4] Snowball stemming algorithm. web: http://snowball.tartarus.org/.

[5] Peter Ahrendt, Anders Meng, and Jan Larsen. Decision time horizon for music genre classification using short time features. In *EUSIPCO*, pages 1293–1296, Vienna, Austria, sep 2004. URL `http://www2.imm.dtu.dk/pubdb/p.php?2981`.

[6] Hirotsugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, Dec 1974. ISSN 0018-9286.

[7] Russell Albright, James Cox, David Duling, Amy N. Langville, and Carl D. Meyer. Algorithms, initializations, and convergence for the nonnegative matrix factorization. Technical Report 81706, North Carolina State University, 2006.

[8] J. Allan, editor. *Topic Detection and Tracking: Event-Based Information Organization.* Kluwer Academic Publishers, Norwell, MA, 2002.

[9] Chris Anderson. The long tail. *Wired*, October 2004.

[10] Jean-Julien Aucouturier and Francois Pachet. Improving timbre similarity : How high's the sky? *Journal of negative results in speech and audio sciences*, 2004.

[11] Claudio Baccigalupo, Justin Donaldson, and Enric Plaza. Uncovering affinity of artists to multiple genres from social behaviour data. In *Ninth International Conference on Music Information Retrieval*, Philadelphia, Pennsylvania USA, 2008.

[12] Brett Bader, Michael Berry, and Murray Browne. *Survey of Text Mining II Clustering, Classification, and Retrieval*, chapter Discussion tracking in Enron email using PARAFAC, pages 147–163. Springer, 2008.

[13] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[14] Stephan Baumann and Oliver Hummel. Enhancing music recommendation algorithms using cultural metadata. *Journal of New Music Research*, 34 (2):161–172, 2005. doi: 10.1080/09298210500175978. URL `http://www.informaworld.com/10.1080/09298210500175978`.

[15] F. Bellomi and R. Bonato. Network analysis of wikipedia. In *Proceedings of Wikimania 2005 - First International Wikimedia Conference*, 2005. URL `http://www.fran.it/articles/wikimania_bellomi_bonato.pdf`.

[16] Michael W. Berry and Murray Brown. Email surveillance using nonnegative matrix factorization. *Computational and Mathematical Organization Theory*, 11:249–264, 2005.

[17] Michael W. Berry and Murray Browne. *Understanding Search Engines - Mathematical Modeling and Text Retrieval.* Society for Industrial and Applied Mathematics, 2nd edition edition, 2005.

[18] Michael W. Berry, Murray Browne, Amy N. Langvilleb, V. Paul Paucac, and Robert J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(Issue 1):155–173, February 2007.

[19] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information Processing & Management*, 43 (4):866 – 886, 2007. ISSN 0306-4573. doi: DOI:10.1016/j.ipm.2006.09.003. URL `http://www.sciencedirect.com/science/article/B6VC8-4M5WJ5B-3/2/156babab16dec5eb5516eb00d72543a3`.

[20] Chris M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995. ISBN 0198538642.

[21] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. ISSN 1533-7928.

[22] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7): 107–117, 1998. URL `citeseer.ist.psu.edu/brin98anatomy.html`.

[23] Rasmus Bro and Sijmen De Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1997.

[24] Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, March 2004. doi: 10.1073/pnas.0308531101.

[25] Wray Buntine. Variational extensions to em and multinomial pca. In *Machine Learning: ECML 2002*, pages 23–34, 2002. doi: 10.1007/3-540-36755-1.

[26] Wray Buntine, Jaakko Löfström, Sami Perttu, and Kimmo Valtonen. Topic-specific scoring of documents for relevant retrieval. In *Workshop on learning in web search*, pages 34–41, Bonn, Germany, 2005.

[27] Bin Cao, Dou Shen, Jian-Tao Sun, Xuanhui Wang, Qiang Yang, and Zheng Chen. Detect and track latent factors with online nonnegative matrix factorization. In *Proc. of IJCAI-07*, pages 2689–2694, Hyderabad, India, 2007.

[28] John Garofolo Cedric, Cedric G. P. Auzanne, and Ellen M. Voorhees. The trec spoken document retrieval track: A success story. In *in Text Retrieval Conference (TREC) 8*, pages 16–19, 2000.

[29] Oscar Celma and Pedro Cano. From hits to niches? or how popular artists can bias music recommendation and discovery. In *2nd Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition (ACM KDD)*, Las Vegas, USA, FebruaryApril/0August/February00August 2008. URL `http://mtg.upf.edu/files/publications/Celma-ACM-Netflix-KDD2008.pdf`.

[30] Oscar Celma and Pedro Cano. From hits to niches? or how popular artists can bias music recommendation and discovery. In *2nd Workshop on Large-Scale Rec- ommender Systems and the Net ix Prize Competition (ACM KDD)*, Las Vegas, USA, August 2008.

[31] Ya chao Hsieh, Yu tsun Huang, Chien chih Wang, and Lin shan Lee. Improved spoken document retrieval with dynamic key term lexicon and probabilistic latent semantic analysis (plsa). In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I, May 2006. doi: 10.1109/ICASSP.2006.1660182.

[32] Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, and Xuan Zhou. Extracting semantic relationships between wikipedia categories. In *1st International Workshop: "SemWiki2006 - From Wiki to Semantics" (SemWiki)*, 2006.

[33] Rudi L. Cilibrasi and Paul M. B. Vitányi. The google similarity distance. *IEEE transactions on knowledge and data engineering*, 19(3):370–383, March 2007.

[34] David Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems 13*, 2001. URL `citeseer.ist.psu.edu/cohn01missing.html`.

[35] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. doi: 10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9.

[36] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. Automatic generation of social tags for music recommendation. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 385–392. MIT Press, Cambridge, MA, 2008.

[37] Paul Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290, 1959.

[38] Christiane Fellbaum, editor. *WordNet: An electronic lexical database.* Language, speech, and communication series. The MIT Press, Cambridge, MA, USA, 1998.

[39] Eric Gaussier and Cyril Goutte. Relation between plsa and nmf and implications. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. doi: http://doi.acm.org.globalproxy.cvt.dk/10.1145/1076034.1076148.

[40] Gijs Geleijnse, Markus Schedl, and Peter Knees. The quest for ground truth in musical artist tagging in the social web era. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[41] Jim Giles. Internet encyclopaedias go head to head. *Nature*, volume 438, number 7070 (15 December):900–901, 2005.

[42] Gene H. Golub and Charles F. van Loan. *Matrix Computations.* The John Hopkins University Press, Baltimore, MD, USA, third edition, 1996.

[43] Edward F. Gonzalez and Yin Zhang. Accelerating the lee-seung algorithm for nonnegative matrix factorization. Technical Report TR05-02, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005, 2005.

[44] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29:34–54, 2005.

[45] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. ISSN 1541-1672. doi: http://doi.ieeecomputersociety.org/10.1109/MIS.2009.36.

[46] J. H. L. Hansen, R. Huang, B. Zhou, M. Seadle, J. R. Deller, A. R. Gurijala, M. Kurimo, and P. Angkititrakul. Speechfind: Advances in spoken document retrieval for a national gallery of the spoken word. *IEEE Transactions on Speech and Audio Processing*, 13(5):712–730, september 2005.

[47] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, New York, NY, USA, 2000. ACM. ISBN 1-58113-222-0. doi: http://doi.acm.org.globalproxy.cvt.dk/10.1145/358916.358995.

[48] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proc. 22nd Annual ACM Conf. on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999. URL `http://citeseer.nj.nec.com/article/hofmann99probabilistic.html`.

[49] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196, January 2001. doi: 10.1023/A:1007617005950.

[50] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, 2004. ISSN 1533-7928.

[51] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 547–554. MIT Press, Cambridge, MA, 2006.

[52] Kasper W. Jørgensen and Lasse L. Mølgaard. Tools for automatic audio indexing. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 2006.

[53] Kasper W. Jørgensen, Lasse L. Mølgaard, and Lars K. Hansen. Unsupervised speaker change detection for broadcast news segmentation. In *Proc. EUSIPCO*, 2006.

[54] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, volume 46, number 5 (September):604–632, 1999. URL `http://www.cs.cornell.edu/home/kleinber/auth.pdf`.

[55] Peter Knees, Elias Pampalk, and Gerhard Widmer. Artist classification with web-based data. In *Proceedings of ISMIR*, Barcelona, Spain, 2004.

[56] Peter Knees, Markus Schedl, and Tim Pohle. A deeper look into web-based classification of music artists. In *Proceedings of 2nd Workshop on Learning the Semantics of Audio Signals*, Paris, France, 2008.

[57] Tamara. G Kolda and Brett W. Bader. The tophits model for higher-order web link analysis. In *Workshop on Link Analysis, Counterterrorism and Security*, 2006.

[58] Thomas Kolenda, Lars K. Hansen, and Jan Larsen. Signal detection using ICA: Application to chat room topic spotting. In *Third International Conference on Independent Component Analysis and Blind Source Separation*, pages 540–545, 2001. URL `http://www2.imm.dtu.dk/pubdb/p.php?826`.

[59] Thomas K. Landauer. *Handbook of Latent Semantic Analysis*, chapter LSA as a theory of meaning, pages 3–35. Lawrence Erlbaum Associates, Mahwah, NJ, 2007.

[60] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000. doi: 10.1137/S0895479896305696. URL `http://link.aip.org/link/?SML/21/1253/1`.

[61] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[62] Lin-Shan Lee and B. Chen. Spoken document understanding and organization. *Signal Processing Magazine, IEEE*, 22(5):42–60, Sept. 2005. ISSN 1053-5888. doi: 10.1109/MSP.2005.1511823.

[63] R. B. Lehoucq. *Analysis and implementation of an implicitly restarted Arnoldi iteration.* Ph.d.-thesis, Rice University, 1995.

[64] Micheline Lesaffre, Liesbeth De Voogdt, Marc Leman, Bernard De Baets, Hans De Meyer, and Jean-Pierre Martens. How potential users of music search and retrieval systems describe the semantic quality of music. *Journal of the American Society for Information Science and Technology*, 59 (5):695–707, 2008. doi: 10.1002/asi.20731. URL `http://dx.doi.org/10.1002/asi.20731`.

[65] Mark Levy and Mark Sandler. Learning latent semantic models for music from social tags. *Journal of New Music Research*, 37(2):137–150, 2008.

[66] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[67] Chih-Jen Lin. Projected gradient methods for non-negative matrix factorization. Technical report, Department of Computer Science, National Taiwan University, 2005.

[68] Chih-Jen Lin. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19:2756–2779, 2007. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/pgradnmf.pdf`.

[69] Chih-Jen Lin. On the convergence of multiplicative update algorithms for non-negative matrix factorization. *Neural Networks, IEEE Transactions on*, 18:1589–1596, 2007.

[70] Michael Littman, Susan T. Dumais, and Thomas K. Landauer. Automatic cross-language information retrieval using latent semantic indexing. In *Cross-Language Information Retrieval, chapter 5*, pages 51–62. Kluwer Academic Publishers, 1998.

[71] Beth Logan, Andrew Kositsky, and Pedro Moreno. Semantic analysis of song lyrics. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2004)*, 2004.

[72] Rasmus Elsborg Madsen, Lars Kai Hansen, and Jan Larsen. Part-of-speech enhanced context recognition. In *Proceedings of IEEE Workshop on Machine Learning for Signal Processing (MLSP) XIV*, pages 635–644, 2004.

[73] Jose P. G. Mahedero, Álvaro MartÍnez, Pedro Cano, Markus Koppenberger, and Fabien Gouyon. Natural language processing of lyrics. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 475–478, New York, NY, USA, 2005. ACM. ISBN 1-59593-044-2. doi: http://doi.acm.org/10.1145/1101149.1101255.

[74] J. Makhoul, F. Kubala, T. Leek, Daben Liu, Long Nguyen, R. Schwartz, and A. Srivastava. Speech and language technologies for audio indexing and retrieval. *Proceedings of the IEEE*, 88(8):1338–1353, Aug 2000. ISSN 0018-9219. doi: 10.1109/5.880087.

[75] W.S. Maki. Judgments of associative memory. *Cognitive Psychology*, 54: 319–353, 2007.

[76] Michael Mandel and Daniel Ellis. Song-level features and support vector machines for music classification. In *International Symposium on Musical Information Retrieval*, 2005.

[77] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[78] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proc. of KDD '05*, pages 198–207. ACM Press, 2005.

[79] Lasse L. Mølgaard., Kasper W. Jørgensen, and Lars Kai Hansen. Castsearch - context based spoken document retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007.*, volume 4, pages IV–93–IV–96, April 2007. doi: 10.1109/ICASSP.2007.367171.

[80] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2): 91–118, July 2003. doi: 10.1023/A:1023949509487.

[81] Morten Mørup and Lars Kai Hansen. Tuning pruning in sparse non-negative matrix factorisation. In *Proceedings of European signal processing conference (EUSIPCO2009)*, Glasgow, Scotland, 2009.

[82] Ramesh Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 542–550, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: http://doi.acm.org/10.1145/1401890.1401957.

[83] Finn Å. Nielsen, Daniela Balslev, and Lars Kai Hansen. Mining the posterior cingulate: Segregation between memory and pain components. *NeuroImage*, 27(3):520–532, 2005. URL `http://www.imm.dtu.dk/~fn/Nielsen2003Posterior.html`.

[84] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error-estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[85] François Pachet and Daniel Cazaly. A taxonomy of musical genres. In *In Proc. Content-Based Multimedia Information Access (RIAO*, 2000.

[86] François Pachet, G. Westermann, and D. Laigre. Musical data mining for electronic music distribution. In *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 101–106, Nov. 2001.

[87] Elias Pampalk, Arthur Flexer, and Gerhard Widmer. Hierarchical organization and description of music collections at the artist level. In *Research and Advanced Technology for Digital Libraries*, volume 3652/2005 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 2005. doi: 10.1007/11551362.

[88] Michael Kai Petersen, Lars Kai Hansen, and Andrius Butkus. Semantic contours in tracks based on emotional tags. In *Computer Music Modeling and Retrieval: Genesis of meaning of sound and music*, 2008.

[89] Mark D. Plumbley. Algorithms for nonnegative independent component analysis. *Neural Networks, IEEE Transactions on*, 14(3):534–543, May 2003.

[90] Mark D. Plumbley and Erkki Oja. A "nonnegative pca" algorithm for independent component analysis. *Neural Networks, IEEE Transactions on*, 15(1):66–76, Jan. 2004.

[91] Tim Pohle, Peter Knees, Markus Schedl, and Gerhard Widmer. Building an interactive next-generation artist recommender based on automatically derived high-level concepts. In *Content-Based Multimedia Indexing, 2007. CBMI '07. International Workshop on*, pages 336–343, June 2007. doi: 10.1109/CBMI.2007.385431.

[92] Martin F. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980. doi: 10.1108/00330330610681286.

[93] Mathew Richardson and Pedro Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002. URL `citeseer.ist.psu.edu/article/richardson02intelligent.html`.

[94] Gerard M. Salton, A. Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of ACM*, 18(11):613–620, 1975. ISSN 0001-0782. doi: http://doi.acm.org.globalproxy.cvt.dk/10.1145/361219.361220.

[95] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl. Application of dimensionality reduction in recommender system - a case study. In *In ACM WebKDD Workshop*, 2000.

[96] Markus Schedl. *Automatically Extracting, Analyzing, and Visualizing Information on Music Artists from the World Wide Web.* PhD thesis, Johannes Kepler Universität Linz, 2008.

[97] Markus Schedl, Peter Knees, and Gerhard Widmer. A web-based approach to assessing artist similarity using co-occurrences. In *Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing (CBMI'05)*, Riga, Latvia, June 2005.

[98] Markus Schedl, Peter Knees, and Gerhard Widmer. Improving prototypical artist detection by penalizing exorbitant popularity. In *Proceedings of 3rd International Symposium on Computer Music Modeling and Retrieval*, pages 196–200, 2005.

[99] Markus Schedl, Peter Knees, Tim Pohle, and Gerhard Widmer. Towards an automatically generated music information system via web content mining. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR'08)*, Glasgow, Scotland, 2008.

[100] Farial Shahnaz, Michael W. Berry, V. Paul Pauca, and Robert J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373 – 386, 2006. ISSN 0306-4573. doi: DOI:10.1016/j.ipm.2004.11.005. URL `http://www.sciencedirect.com/science/article/B6VC8-4F6F64X-3/2/2c5da8c80deabaca36fbadad367435e8`.

[101] Suvrit Sra and Inderjit S. Dhillon. Nonnegative matrix approximation: Algorithms and applications. Technical report, University of Texas at Austin, June 2006.

[102] Sari Suomela and Jaana Kekäläinen. Ontology as a search-tool: A study of real users' query formulation with and without conceptual support. In *ECIR 2005*, volume 3408/2005 of *Lecture Notes in Computer Science (LNCS)*, pages 315–329. Springer Berlin / Heidelberg, 2005. doi: 10.1007/b107096.

[103] Kirsten Swearingen and Rashmi Sinha. Beyond algorithms: An hci perspective on recommender systems. In *Proceedings of the SIGIR 2001 Workshop on Recommender Systems*, 2001.

[104] Panagiotis Symeonidis, Maria Magdalena Ruxanda, Alexandros Nanopoulos, and Yannis Manolopoulos. Ternary semantic analysis of social tags for personalized music recommendation. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 219–224, Philadelphia, USA, 2008.

[105] Daniel Torres, Douglas Turnbull, Luke Barrington, and Gert Lanckriet. Identifying words that are musically meaningful. In *Proceedings of International Symposium on Music Information Retrieval (ISMIR)*, 2007.

[106] Karen H. L. Tso-Sutter, Leandro Balby Marinho, and Lars Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1995–1999, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-753-7. doi: http://doi.acm.org.globalproxy.cvt.dk/10.1145/1363686.1364171.

[107] J.-M. Van Thong, P.J. Moreno, B. Logan, B. Fidler, K. Maffey, and M. Moores. Speechbot: an experimental speech-based search engine for multimedia content on the web. *Multimedia, IEEE Transactions on*, 4(1): 88–96, Mar 2002. ISSN 1520-9210. doi: 10.1109/6046.985557.

[108] Vishwa Vinay, Ingemar J. Cox, Natasa Milic-Frayling, and Ken Wood. Evaluating relevance feedback algorithms for searching on small displays. In *27th European Conference on IR Research (ECIR)*, 2005.

[109] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx-4: A flexible open source framework for speech recognition. Tech Report TR-2004-127, Sun Microsystems, 2004.

[110] Brian Whitman. Semantic rank reduction of music audio. *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 135–138, Oct. 2003.

[111] Brian Whitman and Steve Lawrence. Inferring descriptions and similarity for music from community metadata. In *In Proceedings of the 2002 International Computer Music Conference*, pages 591–598, 2002.

[112] Brian A. Whitman. *Learning the meaning of music.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2005. Supervisor-Vercoe, Barry L.

[113] Steve Whittaker, Julia Hirschberg, John Choi, Don Hindle, Fernando Pereira, and Amit Singhal. Scan: designing and evaluating user interfaces to support retrieval from speech archives. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26–33, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1. doi: http://doi.acm.org.globalproxy.cvt.dk/10.1145/312624.312639.

[114] Dennis M. Wilkinson and Bernardo A. Huberman. Assessing the value of cooperation in wikipedia. *First Monday*, volume 12, number 4 (April), 2007. URL `http://www.firstmonday.org/issues/issue12_4/wilkinson/`.

[115] Wei Xu, Xing Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proc. of the 26th International ACM SIGIR*, pages 267–273, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-646-3. doi: http://doi.acm.org.globalproxy.cvt.dk/10.1145/860435.860485.

[116] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *Audio,*

Speech, and Language Processing, IEEE Transactions on, 16(2):435–447, Feb. 2008. ISSN 1558-7916. doi: 10.1109/TASL.2007.911503.

[117] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM Comput. Surv.*, 38(2):6, 2006. ISSN 0360-0300. doi: http://doi.acm.org.globalproxy.cvt.dk/10.1145/1132956.1132959.