Natural language processing for music information retrieval : deep analysis of lyrics structure and content

Michael Fell

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Traitement Automatique des Langues pour la Recherche d'Information Musicale: Analyse Profonde de la Structure et du Contenu des Paroles de Chansons

Michael FELL

Wimmics, CNRS, I3S, Inria

Présentée en vue de l'obtention du grade de docteur en Informatique **de l'**Université Côte d'Azur
Dirigée par
Elena Cabrio et Fabien Gandon
Soutenue le 29.04.2020

Devant le jury, composé de:
Claire Gardent, Senior Researcher, CNRS
Carlo Strapparava, Senior Researcher, FBK
Philipp Cimiano, PR, Universität Bielefeld
Frédéric Precioso, PR, Université Côte d'Azur
Elena Cabrio, MCF, Université Côte d'Azur
Fabien Gandon, DR, Inria

cnrs i3s sophia antipolis Inria inventeurs du monde numérique wimmics

Traitement Automatique des Langues pour la Recherche d'Information Musicale: Analyse Profonde de la Structure et du Contenu des Paroles de Chansons

COMPOSITION DU JURY

Rapporteurs
Claire Gardent, Senior Researcher, CNRS/LOREA
Carlo Strapparava, Senior Researcher, FBK-IRST

Examinateurs
Philipp Cimiano, PR, Universität Bielefeld
Frédéric Precioso, PR, Université Côte d'Azur

Directeurs de thèse
Elena Cabrio, MCF, Université Côte d'Azur, Inria, CNRS, I3S
Fabien Gandon, DR, Inria, Université Côte d'Azur, CNRS, I3S

ABSTRACT

Applications in Music Information Retrieval and Computational Musicology have traditionally relied on features extracted from the music content in the form of audio, but mostly ignored the song lyrics. More recently, improvements in fields such as music recommendation have been made by taking into account external metadata related to the song. In this thesis, we argue that extracting knowledge from the song lyrics is the next step to improve the user's experience when interacting with music. To extract knowledge from vast amounts of song lyrics, we show for different textual aspects (their structure, content and perception) how Natural Language Processing methods can be adapted and successfully applied to lyrics. For the structural aspect of lyrics, we derive a structural description of it by introducing a model that efficiently segments the lyrics into its characteristic parts (e.g. intro, verse, chorus). In a second stage, we represent the content of lyrics by means of summarizing the lyrics in a way that respects the characteristic lyrics structure. Finally, on the perception of lyrics we investigate the problem of detecting explicit content in a song text. This task proves to be very hard and we show that the difficulty partially arises from the subjective nature of perceiving lyrics in one way or another depending on the context. Furthermore, we touch on another problem of lyrics perception by presenting our preliminary results on Emotion Recognition. As a result, during the course of this thesis we have created the annotated WASABI Song Corpus, a dataset of two million songs with NLP lyrics annotations on various levels.

KEYWORDS

Natural Language Processing, Text Segmentation, Multimodality, Text Summarization, Emotion Recognition, Text Classification, Lyrics Corpus

RÉSUMÉ

Les applications en Recherche d'Information Musicale et en musicologie computationnelle reposent traditionnellement sur des fonctionnalités extraites du contenu musical sous forme audio, mais ignorent la plupart du temps les paroles des chansons. Plus récemment, des améliorations dans des domaines tels que la recommandation de musique ont été apportées en tenant compte des métadonnées externes liées à la chanson. Dans cette thèse, nous soutenons que l'extraction des connaissances à partir des paroles des chansons est la prochaine étape pour améliorer l'expérience de l'utilisateur lors de l'interaction avec la musique. Pour extraire des connaissances de vastes quantités de paroles de chansons, nous montrons pour différents aspects textuels (leur structure, leur contenu et leur perception) comment les méthodes de Traitement Automatique des Langues peuvent être adaptées et appliquées avec succès aux paroles. Pour l'aspect structurel des paroles, nous en dérivons une description structurelle en introduisant un modèle qui segmente efficacement les paroles en leurs parties caractéristiques (par exemple, intro, couplet, refrain). Puis, nous représentons le contenu des paroles en résumant les paroles d'une manière qui respecte la structure caractéristique des paroles. Enfin, sur la perception des paroles, nous étudions le problème de la détection de contenu explicite dans un texte de chanson. Cette tâche s'est avèree très difficile et nous montrons que la difficulté provient en partie de la nature subjective de la perception des paroles d'une manière ou d'une autre selon le contexte. De plus, nous abordons un autre problème de perception des paroles en présentant nos résultats préliminaires sur la reconnaissance des émotions. L'un des résultats de cette thèse a été de créer un corpus annoté, le WASABI Song Corpus, un ensemble de données de deux millions de chansons avec des annotations de paroles TAL à différents niveaux.

MOTS-CLÉS
Traitement Automatique des Langues, Segmentation de Texte, Multimodalité, Résumé du Texte, Reconnaissance des Emotions, Classification de Texte, Corpus des Paroles

Dedicated to Toorum.

LIST OF PUBLISHED PAPERS

Michael Fell, Yaroslav Nechaev, Elena Cabrio, and Fabien Gandon. "Lyrics Segmentation: Textual Macrostructure Detection using Convolutions." In: *Conference on Computational Linguistics (COLING)*. Santa Fe, New Mexico, United States, 2018, pp. 2044–2054.

Diego Monti, Enrico Palumbo, Giuseppe Rizzo, Pasquale Lisena, Raphaël Troncy, Michael Fell, Elena Cabrio, and Maurizio Morisio. "An Ensemble Approach of Recurrent Neural Networks using Pre-Trained Embeddings for Playlist Completion." In: *Proceedings of the ACM Recommender Systems Challenge, RecSys Challenge 2018, Vancouver, BC, Canada, October 2, 2018*. 2018, 13:1–13:6.

Michael Fell, Elena Cabrio, Michele Corazza, and Fabien Gandon. "Comparing Automated Methods to Detect Explicit Content in Song Lyrics." In: *Recent Advances in Natural Language Processing (RANLP*. Varna, Bulgaria, 2019.

Michael Fell, Elena Cabrio, Fabien Gandon, and Alain Giboin. "Song Lyrics Summarization Inspired by Audio Thumbnailing." In: *Recent Advances in Natural Language Processing (RANLP)*. Varna, Bulgaria, 2019.

Michael Fell, Elena Cabrio, Elmahdi Korfed, Michel Buffa, and Fabien Gandon. "Love Me, Love Me, Say (and Write!) that You Love Me: Enriching the WASABI Song Corpus with Lyrics Annotations." In: *Language Resources and Evaluation Conference (LREC)*. Marseille, France, 2020.

SUBMITTED PAPERS

Michael Fell, Yaroslav Nechaev, Gabriel Meseguer-Brocal, Elena Cabrio, Fabien Gandon, and Geoffroy Peeters. "Lyrics Segmentation via Bimodal Text-audio Representation." In: *Natural Language Engineering* (2020).

ACKNOWLEDGMENTS

This Thesis has come to life under the guidance of my PhD advisors Elena Cabrio and Fabien Gandon. Elena, thank you for always being there to discuss my work, for encouraging me when I needed it most, and for providing me with unique opportunities such as the work exchange to Argentina. Thank you Fabien, for helping me prioritize my work and nurturing an amazingly productive and familial working environment in the Wimmics team. Thank you both, for the great patience and for your faith in me. I am very fortunate to have worked with you.

Claire Gardent and Carlo Strapparava, I am indebted to both of you for your hard work of reviewing my Thesis. I also want to thank Philipp Cimiano and Frédéric Precioso for being part of the examination committee. Thank you all, for having accepted my request to evaluate my work.

I want to thank everyone in the WASABI project, especially Michel Buffa for providing me with all resources I needed and Elmahdi Korfed for helping me with all my database needs. Thanks again to each and everyone of you who took part in the lyrics summary experiment.

I am thankful to all the members of the Wimmics research group for making me feel welcome from the start and for elevating my life on some level. Special mentions for Tobias and Michele for the numerous technical and non-technical discussions as well as their friendship outside of the lab.

I would like to thank the people I had the opportunity to collaborate with and learn from on my work exchanges during these three years - Laura Alonso Alemany of the Universidad Nacional de Córdoba (Argentina) and Luigi Di Caro of the Università degli Studi di Torino (Italy).

Thanks to my friends Matthias and Patrick, for your treasured friendship through the years. Thanks to my brother Christian and my parents Wolfgang and Karin for supporting me in every possible way. Thank you Candelaria, for your love and support and for rewriting the stars with me.

FUNDING
I want to thank the Agence nationale de la recherche for funding my work in the framework of the WASABI Project under the contract ANR-16-CE23-0017-01.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

In this Chapter we introduce the context and the motivation underlying the present research work, and position it in the multidisciplinary framework of the research.

CONTENTS

1.1 CONTEXT AND MOTIVATION

When a popular song plays on the radio, it is easy to remember and sing along with the melody, but it is much harder to remember the lyrics. While we intuitively and automatically process the melody, understanding and memorizing the lyrics can require an effort from us. Because of this asymmetry in perception, we can listen to a song many times and only remember small parts of the lyrics. It is only understandable then, why in everyday life the lyrics are often perceived as less important to a song than the melody.

What factors are responsible for the perception of music and how to extract and use such information, is studied in the field of **Music Information Retrieval (MIR)**. Music perception is influenced by the factors **music content**, **music context** and **listener-related factors** [103]. The music content is defined by the audio and the lyrics of the song. The music context, on the other hand, is defined by the relations of the song to the world, such as knowledge on the artist or the genre. And the listener-related factors are, for instance, his music preferences or her current mood. Building on this model of perception, MIR applications

leverage one or more of the above mentioned factors with the goal of improving or focusing the music listening experience. Example applications of automated systems in MIR are **music recommendation** and **music search engines**. To provide the listener with relevant recommendations, large databases of music are automatically analyzed by content aspects (e.g. song melody, song topic, song structure and emotions) and song context (e.g. artist and genre). Then, songs with similar content or context aspects can be recommended. For music search engines, an abundance of search criteria may be used in an advanced search interface, allowing a search for songs of a specific topic, structure, emotion and genre. Common to both of these higher level MIR applications is that they require high quality lower level automated tasks to be carried out, such as topic identification, melody estimation and emotion recognition.

Before and besides MIR, musicologists and music enthusiasts have gathered a plethora of knowledge about music over the times. Musical knowledge, however, can be unstructured (e.g. a blog post about the favorite band) and musicologists are usually not experts in MIR. As an example, imagine a journalist who prepares a radio show about the life of David Bowie. The show may require an overview of the work of the singer which is so far not readily available. Easy access to a music search engine enables the journalist to search for David Bowie's songs, grouped by, for instance, their topics, the time of their publication and the emotions they convey. An automatic historical music analysis can be provided to reveal how different musical properties have changed over time, for instance the artist may have changed his style significantly over time in an attempt to stick up with the trend or to set it. Given such deep analysis of songs, the journalist is provided with valuable knowledge, enabling him to rapidly assemble all the data needed for his show.

Traditionally, MIR has focused only on one part of the music content: the audio track. More recent approaches [85] showed that considering the context of the song is beneficial. For instance, they show that performance in the task of retrieving similar artists is improved by enriching

the artist representation by semantically linking them to a knowledge base [86]. While these approaches succeed in leveraging the music context, they still put little weight on the lyrics. As the lyrics are not integrated into modern music recommendation systems, recommending songs with lyrics that convey similar topics or have a similar writing style is not possible. We argue that, just as the music, the lyrics are an integral part of a song's content. Despite that, song lyrics have seen only low attention compared to audio.

While MIR has enabled more and more useful applications and has put vast knowledge into the hands of musicologists, we believe that putting more emphasis on the inclusion of lyrics into the MIR paradigm will improve the music listening experience even further. For instance, this will allow for music recommendations based on the topics and core ideas expressed in the lyrics. Today, discovering playlists of a given topic is only possible when curated playlists are available, as topics are inherently encoded in the lyrics but not in the audio. Furthermore, the lower level modules that the recommender relies on perform better when lyrics are taken into account. For instance, genre classification and emotion detection systems have been shown to perform better when the lyrics content is considered aside the audio content.

To approach MIR with lyrics in mind, we need to automatically process large amounts of text and our solution to this is **Natural Language Processing (NLP)**. In the framework of Artificial Intelligence, the field of NLP is located at the intersection of linguistics and computer science and its goal is to automatically process natural languages. Methods in NLP have been developed, for example, to extract knowledge from texts, such as the topics, the emotion or the author's style. We believe that a textual content analysis of the lyrics via NLP will enable novel searches like *find songs where the chorus talks about hope, but the verse talks about struggle* and allow the user to find songs that are on a currently hot topic. Our decision to apply NLP to song lyrics makes us face some core challenges. Song lyrics have a distinctively different language than other domains which poses problems for standard NLP methods. This is why we

have developed NLP methods to deal with the peculiarities of lyrics. Some relevant differences are the following. Lyrics are structurally special in the way that they do not consist of sentences and paragraphs, but of lines and segments where line breaks can occur in the middle of a sentence. This can prohibit the use of off-the-shelf NLP models such as POS taggers, parsers and consequently downstream tasks such as information extraction. Furthermore, lyrics are fundamentally highly repetitive texts. Ignoring this fact can for instance lead to highly redundant summaries. Finally, lyrics are tightly related to poems and as such can contain an abundance of figurative and poetic language. Leaving large parts of it to the interpretation of the listener is fine. Because of this, tasks such as the detection of explicit language (full of profanities) in lyrics is inherently subjective.

1.2 THESIS CONTRIBUTIONS

Since (i) song lyrics are texts that are inherently connected to a piece of music and as (ii) we develop automated methods to study song lyrics, this Thesis is located at the crossroad of (i) the musicological field of Music Information Retrieval and (ii) the computer science field of Natural Language Processing.

To cope with the different levels of analysis required to face the challenges raised in our work, the overall structure of this Thesis is in analogy to the levels of abstraction in NLP theory as explained in the following. Human language can be thought to work on different levels of abstraction, where the lower abstraction level of **Syntax** deals with questions such as *what kinds of words are there?* (e.g. verb, noun, adjective) and *in which order are words arranged in sentences?* (e.g. subject-predicate-object). The higher abstraction level of **Semantics** deals with the question *what is the meaning of this word?* and *what is the meaning of this text?*. The highest abstraction level, **Pragmatics**, is concerned with the questions *how is language used and perceived?*. In analogy to the questions asked in these three levels of abstraction, we stip-

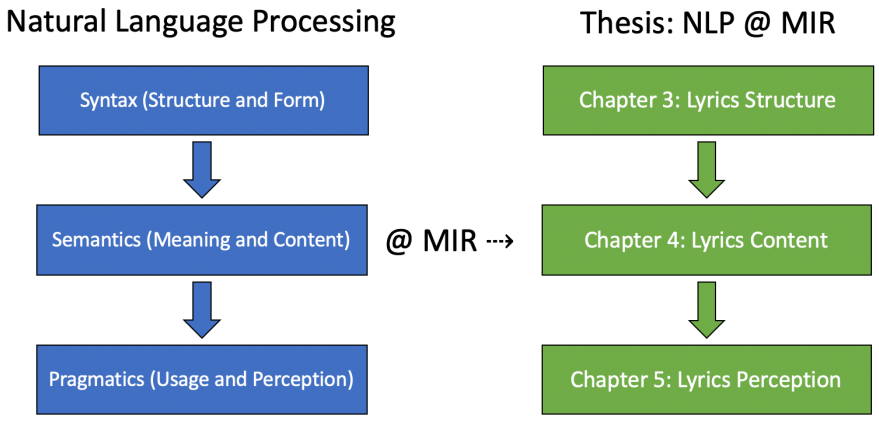


Figure 1.1: Typical NLP pipeline view (left) and the analogy we follow when applying NLP to MIR in this thesis (right).

ulate three corresponding levels of lyrics analysis and make a contribution to each one of them: the **Lyrics Structure**, the **Lyrics Content** and the **Lyrics Perception**. Figure 1.1 visualizes the different levels of analysis and the structure of this Thesis. We further develop NLP methods which overcome the previously described limitations caused by the anomalous nature of lyrics as texts.

Lyrics Structure: We **segment lyrics** into their structural building blocks (such as intro, verse and chorus). Our segmentation method draws from audio structure analysis and is fundamentally based on the repetitive nature of the lyrics. Furthermore, we show that using both the text and the audio of a song improves on segmentation performance. This work was partially published at the conference *COLING 2018* (unimodal text-based part) and is under review in the journal *Natural Langue Engineering* (bimodal text-audio-based part).

Lyrics Content: Our method for **lyrics summarization** takes into account their audio nature. First, we adopt the generic text summarization view to produce summaries that contain the central sentences of the text. Then we incorporate the lyrics thumbnail perspective into the summary by weighting the more repetitive and representative parts of the lyrics higher. With this we draw an analogy to audio

summarization and we can show that summaries created in such a way are perceived as of higher quality. This work was published at RANLP 2019.

Lyrics Perception: We show the limitations of existing NLP methods to deal with subjective and artistic text genres such as lyrics. We compare different methods to **detect explicit content in lyrics**. Our findings show that the task is highly subjective and therefore very hard. The figurative and poetic language used in lyrics together with contexts such as music genre impedes reaching a consensus on what qualifies the language in lyrics to be "explicit". This work was published at RANLP 2019.

Integrating our work, we **release the large-scale WASABI Song Corpus** enriched by different NLP annotations, partially devised from our methods developed in this thesis. The dataset of 2 million songs is introduced in Section 2.3 and the result of the annotation is described in Chapter 6. This work has been accepted at LREC 2020.

1.3 STRUCTURE OF THE THESIS

The Thesis is structured as follows:

Chapter 2 introduces the WASABI Project in which this thesis has been written and introduces the reader to the WASABI Song Corpus, the central dataset we use for experimentation throughout this thesis.

Chapter 3 deals with the structural aspect of lyrics. Given a song text, we derive a structural description of it. We introduce a model that efficiently segments the lyrics into its characteristic parts. We finally discuss an estimation of labelling the segments.

Chapter 4 deals with the problem of representing the content of lyrics. We initially explore different possible representations based on topic models and information extraction. We then introduce our content representation by means of summarizing the lyrics in a way that respects

the characteristic lyrics structure. We end with an outlook on a more abstractive summarization.

Chapter 5 deals with the perception of lyrics in the world. As an instantiation we discuss the problem of detecting explicit content in a song text. This task proves to be very hard and we show that the difficulty partially arises from the subjective nature of perceiving lyrics in one way or another depending on the context. Furthermore, we touch on another problem of lyrics perception by presenting our preliminary results on Emotion Recognition.

Chapter 6 describes the annotated WASABI Song Corpus, a dataset we have created by enriching the dataset described in Section 2.3 with NLP annotations of different levels based on methods we developed in the previous Chapters.

Chapter 7 concludes the Thesis drawing final remarks and suggesting directions for future improvements.

2

THE WASABI PROJECT

In this Chapter we give a brief overview over the WASABI Project in the context of which this thesis has been written. We clarify the differences to similar projects and introduce the reader to the WASABI Song Corpus, the central dataset we use for experimentation throughout this work.

CONTENTS

2.1 MOTIVATION AND GOALS

The **WASABI project**[1] (**W**eb **A**udio **S**emantic **A**ggregated in the **B**rowser for **I**ndexation)[75] is a research project conducted from early 2017 until mid 2020 which is founded by the French National Agency for Research (ANR) under the contract ANR-16-CE23-0017-01. Its goal is to enable scenarios in Music Information Retrieval and Musicology such as described in Section 1.1. The multidisciplinary project assembles partners from various backgrounds: computational linguists, computer scientists and software engineers from the I3S laboratory of Université Côte d'Azur, IRCAM, Deezer, and the Parisson company. Other collaborators are journalists and archivists from Radio France, as well as music composers, musicologists, music schools and sound engineering schools. The first of the two goals of the WASABI Project is the construction of a large-scale song knowledge base, which we call the **WASABI Song Corpus**. It was specified to combine metadata collected from music databases

1 http://wasabihome.i3s.unice.fr/

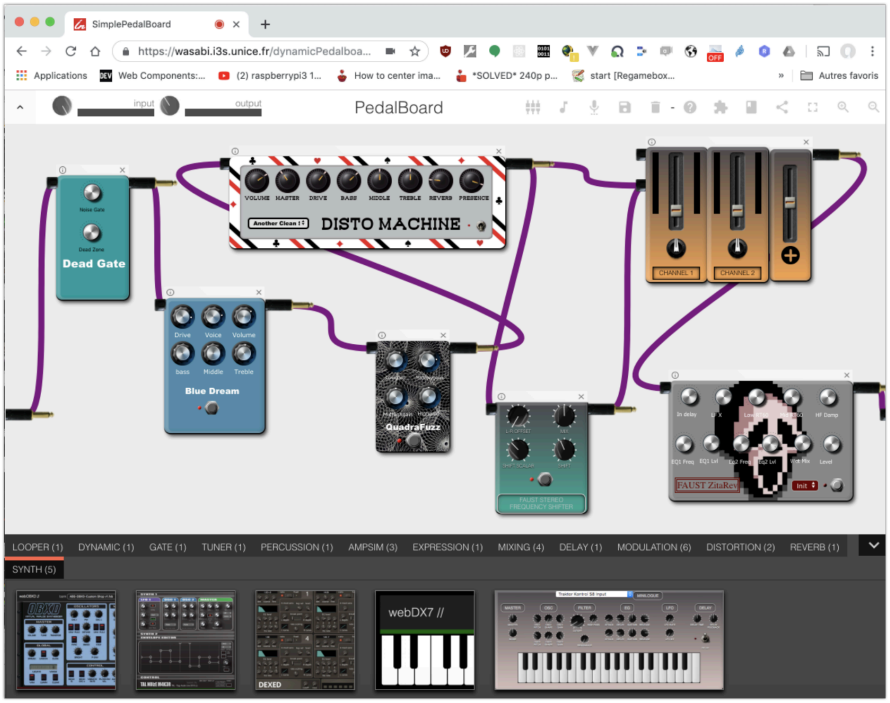


Figure 2.1: The pedalboard with loaded plugins. Illustration taken from [21].

on the Web (e.g. artists, discography, producers, year of production), metadata resulting from the audio analysis (e.g. beat, loudness, chords, structure, cover detection, source separation) and metadata resulting from the analysis of song lyrics to answer questions auch as *What topics is this song about?*, *Which emotions are conveyed?* and *What is the structure of the song lyrics?*. The second project goal is the development of semantic applications that add high value by exploiting the semantic database. Apps such as an online mixing table, guitar amp simulations with a virtual pedalboard (see Figure 2.1), audio analysis visualization tools, annotation tools and a similarity search tool that works by uploading audio extracts or playing some melody using a MIDI device.

While similar attempts as ours have been made (see next Section), our project is uniquely built on a broader scope than other projects and mixes a wider set of metadata. The WASABI Project is based on the collaboration between the Semantic Web models and algorithms to obtain semantic

metadata from the web, the Natural Language Processing algorithms that extract information from the song lyrics and the algorithms from the Music Information Retrieval domain that work on the audio, altogether producing a richer and more consistent knowledge base.

2.2 RELATED WORK

The Million Song Dataset (MSD) project[2] [12] is a collection of audio features and metadata for a million contemporary popular music tracks. The metadata is extracted from Web resources (e.g. artist names, tags, years) and the audio. Given that MSD mainly focuses on audio data, the complementary the musiXmatch dataset[3] has been released, associating MSD songs with their lyrics in bag-of-words (BOW) representation. Contrarily, in the WASABI Project we extract knowledge from the full lyrics.

MusicWeb and its successor MusicLynx [2] link music artists within a Web-based application for discovering connections between them and provide a browsing experience using extra-musical relations. The project shares some ideas with WASABI, but works on the artist level, and does not perform analyses on the audio and lyrics content itself. It reuses, for example, MIR metadata from AcousticBrainz.

Companies such as Spotify, GraceNote, Pandora, or Apple Music have sophisticated private knowledge bases of songs and lyrics to feed their search and recommendation algorithms, but such data is not available publicly, and they mainly rely on audio features.

2.3 THE WASABI SONG CORPUS

In the context of the WASABI research project, a two million song database has been built, the **WASABI Song Corpus**. It contains metadata on 77k artists, 208k albums, and 2.10M songs. The metadata has been *i)* aggregated, merged and curated from different data sources on the Web, and *ii)*

2 http://millionsongdataset.com
3 http://millionsongdataset.com/musixmatch/

enriched by pre-computed or on-demand analyses of the lyrics and audio data. The partners in the WASABI Project have performed various levels of analysis and built interactive Web Audio applications on top of the output. For example, the TimeSide analysis and annotation framework have been linked [47] to make on-demand audio analysis possible. In connection with the FAST project[4], an offline chord analysis of 442k songs has been performed, and both an online enhanced audio player [94] and chord search engine [95] have been built around it. A rich set of Web Audio applications and plugins has been proposed [17–19], that allow, for example, songs to be played along with sounds similar to those used by artists.

All these metadata, computational analyses and Web Audio applications have now been gathered in one easy-to-use web interface, the **WASABI Interactive Navigator**[5], illustrated in Figure 2.2.

The partners in the WASABI Project started building the WASABI Song Corpus by collecting for each artist the complete discography, band members with their instruments, time line, equipment they use, and so on. For each song they collected its lyrics from LyricWiki[6], the synchronized lyrics when available[7], the DBpedia abstracts and the categories the song belongs to: genre, label, writer, release date, awards, producers, artist and band members, the stereo audio track from Deezer, the unmixed audio tracks of the song, its ISRC, bpm and duration. Then, they matched the song identifiers from the WASABI Song Corpus with the identifiers from MusicBrainz, iTunes, Discogs, Spotify, Amazon, AllMusic, GoHear and YouTube. Figure 2.3 shows all the data sources we have used to create the WASABI Song Corpus. We have also aligned the WASABI Song Corpus with the publicly available LastFM dataset[8], resulting in 327k tracks in our corpus having a LastFM id.

4 http://www.semanticaudio.ac.uk
5 http://wasabi.i3s.unice.fr/
6 http://lyrics.wikia.com/
7 from http://usdb.animux.de/
8 http://millionsongdataset.com/lastfm/

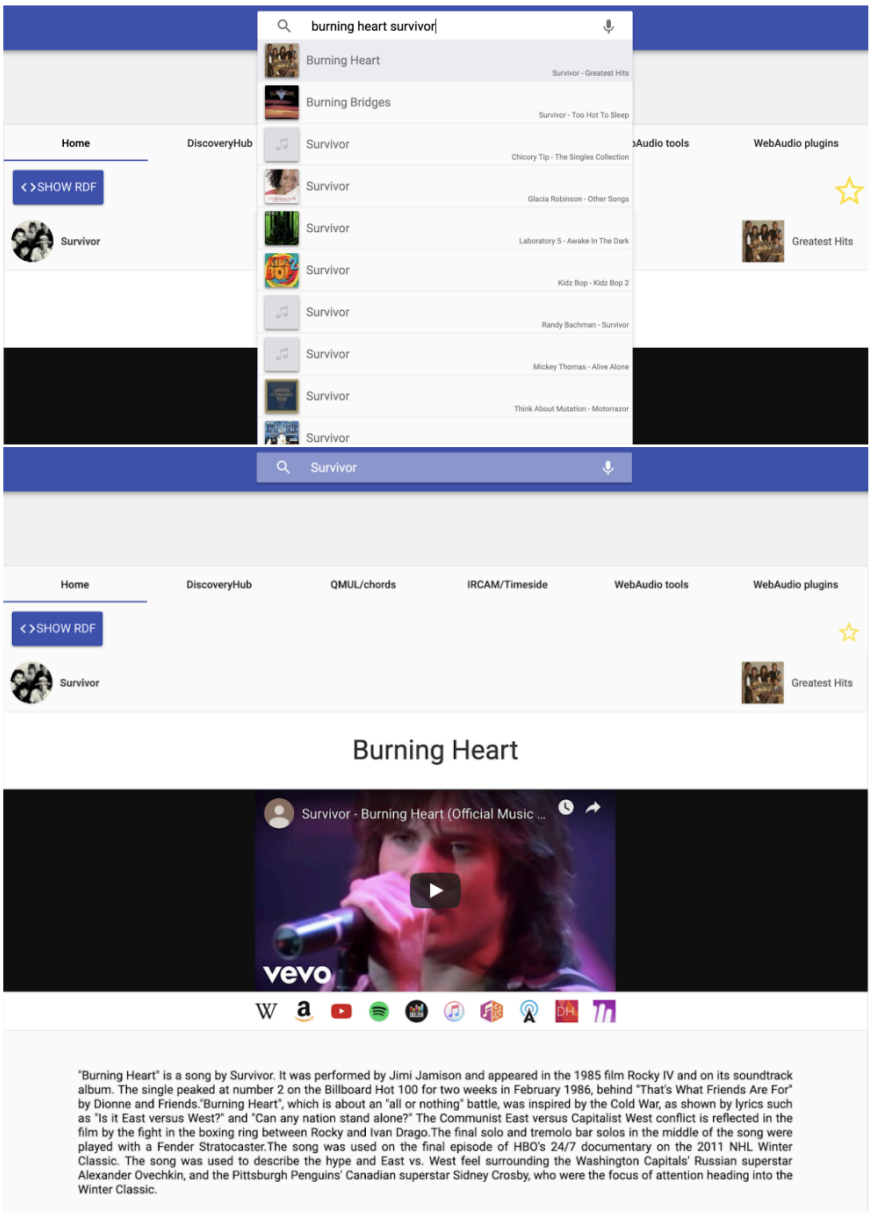


Figure 2.2: The WASABI Interactive Navigator. Illustration taken from [20].

Figure 2.3: The datasources connected to the WASABI Song Corpus. Illustration taken from [21].

As of today, the corpus contains 1.73M songs with lyrics (1.41M unique lyrics). 73k songs have at least an abstract on DBpedia, and 11k have been identified as *classic songs*, meaning they have been number one, got a Grammy award or have lots of cover versions. About 2k songs have a multi-track audio version, and on-demand source separation using Open-Unmix [108] or Spleeter [54] is provided as a TimeSide plugin.

In the remainder of this Chapter, we first present key statistics on the **initial corpus** (before NLP annotations). We then introduce the NLP annotations we have added to obtain the **annotated WASABI Song Corpus**, which is described in detail in Chapter 6. We close the current Chapter with the technical details on the accessibility of the annotated WASABI Song Corpus.

LANGUAGE DISTRIBUTION

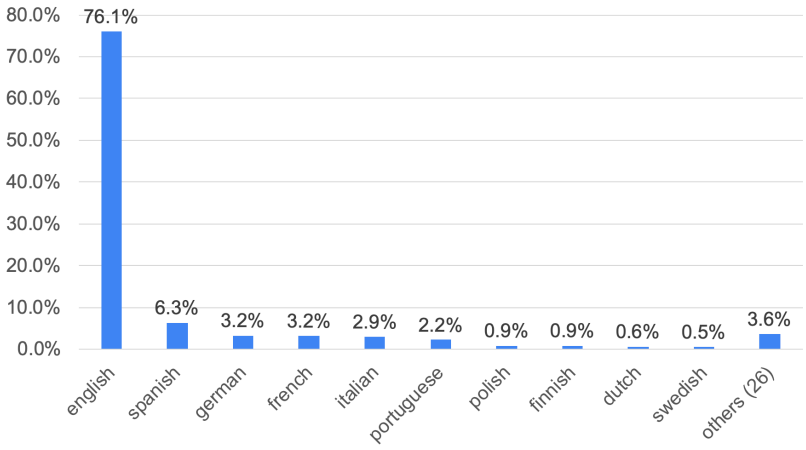
Figure 2.4a shows the distribution of the ten most frequent languages in the WASABI Song Corpus[9]. In total, the corpus contains songs of 36 different languages. The vast majority (76.1%) is English, followed by Spanish (6.3%) and by four languages in the 2-3% range (German, French, Italian, Portugese). On the bottom end, Swahili and Latin amount to 0.1% (around 2k songs) each.

GENRE DISTRIBUTION

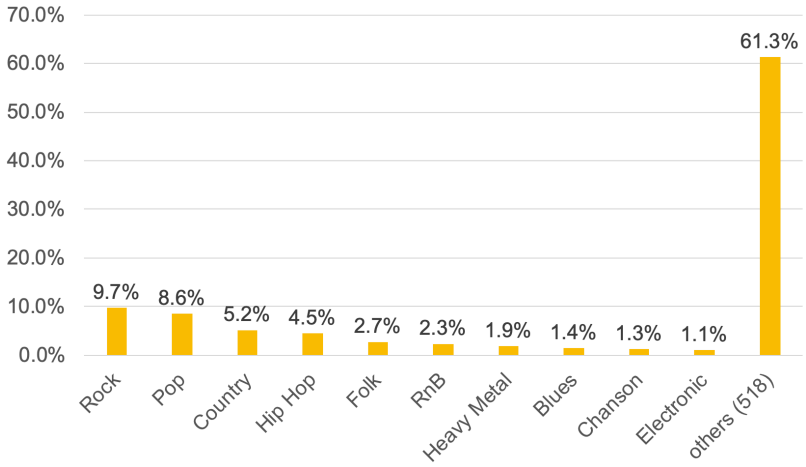
In Figure 2.4b we depict the distribution of the ten most frequent genres in the WASABI Song Corpus[10]. In total, 1.06M of the titles are tagged with a genre. It should be noted that the genres are very sparse with a total of 528 different ones. This high number is partially due to many subgenres such as Alternative Rock, Indie Rock, Pop Rock, etc. which we omitted in Figure 2.4b for clarity. The most common genres are Rock (9.7%), Pop (8.6%), Country (5.2%), Hip Hop (4.5%) and Folk (2.7%).

9 Based on language detection performed on the lyrics.

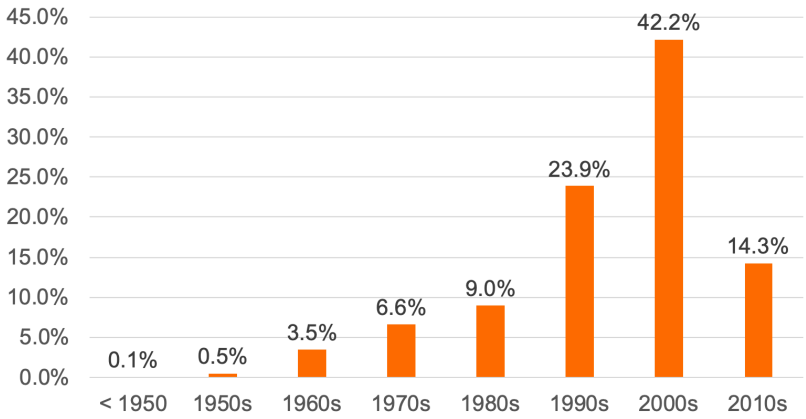
10 We take the genre of the album as ground truth since song-wise genres are much rarer.



(a) Language distribution (100% = 1.73M)



(b) Genre distribution (100% = 1.06M)



(c) Decade of publication distribution (100% = 1.70M)

Figure 2.4: Statistics on the WASABI Song Corpus

PUBLICATION YEAR

Figure 2.4c shows the number of songs published by decade[11]. We find that over 50% of all songs in the WASABI Song Corpus are from the 2000s or later and only around 10% are from the seventies or earlier.

NLP ANNOTATIONS

Several Natural Language Processing methods have been applied to the lyrics of the songs included in the WASABI Song Corpus, as well as various analyses of the extracted information have been carried out. As we develop these NLP methods in the following Chapters, we will finally describe the different annotations we have added to the song lyrics in the dataset in Section 6.2. Based on the research we have conducted, we add the following lyrics annotations: lyrical structure, lyrics summary, explicit content in lyrics, emotions contained in lyrics, and topics in lyrics. We conclude with a diachronic analysis of prominent topics in the WASABI Song Corpus in Section 6.3.

ACCESSIBILITY OF THE WASABI SONG CORPUS

The WASABI Interactive Navigator relies on multiple database engines: it runs on a MongoDB server altogether with an indexation by Elasticsearch and also on a Virtuoso triple store as a RDF graph database. It comes with a REST API[12] and an upcoming SPARQL endpoint. All the database metadata is publicly available[13] under a CC licence through the WASABI Interactive Navigator as well as programmatically through the WASABI REST API. We provide the files of the current version of the WASABI Song Corpus, the models we have built on it as well as updates here: `https://github.com/micbuffa/WasabiDataset`.

11 We take the album publication date as proxy since song-wise labels are too sparse.

12 `https://wasabi.i3s.unice.fr/apidoc/`

13 There is no public access to copyrighted data such as lyrics and full length audio files. Instructions on how to obtain lyrics are nevertheless provided and audio extracts of 30s length are available for nearly all songs.

3

LYRICS STRUCTURE

In this Chapter we deal with the problem of detecting the structure in lyrics. We reduce the problem to the subtasks lyrics segmentation and segment labelling. We introduce a model that efficiently segments the lyrics[1]*. We further discuss segment labelling.*

CONTENTS

3.1 INTRODUCTION

As we all know, lyrics are texts that accompany a piece of music, and just like music they come in all shapes and sizes. For instance, consider the three lyrics depicted in

1 This work has been published at the conference *COLING 2018* (unimodal text-based part) and is under review in the journal *Natural Language Engineering* (bimodal text-audio-based part).

Figure 3.1. The green and yellow lyrics are both from the Rock band Guns N' Roses while the violet one is from the Rappers Wu-Tang Clan. These examples illustrate a number of different structural properties of lyrics. First, note that all lyrics consist of **text lines** which in turn consist of words. As our illustration shows, these text lines come in different lengths - ranging from short (green) to very long (violet) - and can be single words, phrases or even full sentences. As these lyrics are associated with a song, matching the melody, rhythm and beat with the lyrics line is more important to the composer than forming a grammatically correct sentence. Second, a song text typically is formed of different **text segments** consisting of text lines. The green text consists of two segments while the yellow one is made of six segments. The violet lyric is basically a single giant segment, which is a typical property of lyrics from the genre of Rap. Looking a bit closer we find that in the yellow and the green lyrics certain segments are repeated, constituting a typical structure of a large range of songs. More precisely, the repeated elements are chorus, which we indicate with the green and yellow boxes in Figure 3.1. The chorus usually is one of the most important parts of a song and is the part people remember best of a song. While **repetition** is a key element in song lyrics, it is not always there, as illustrated by the violet text.

In many lyrics, characteristic text segments are present and can be labelled with terms such as *intro*, *verse*, *bridge*, *chorus* and *outro* [16]. These labels have a tradition in musicology as descriptions of song structure, but we can also use them to describe the lyrics since their structure tends to mirror the song structure.

Accurately describing the structure of a song text is a non-trivial task that requires diverse knowledge. Algorithms that aim to automatically detect the structure usually operate in two steps: a **lyrics segmentation** stage that divides lyrics into segments, and a **segment labelling** stage that labels each segment with a structure type (e.g. intro, verse, chorus).

Although a few works have addressed the task of finding chorus or repeated parts in music [6, 72], full structure

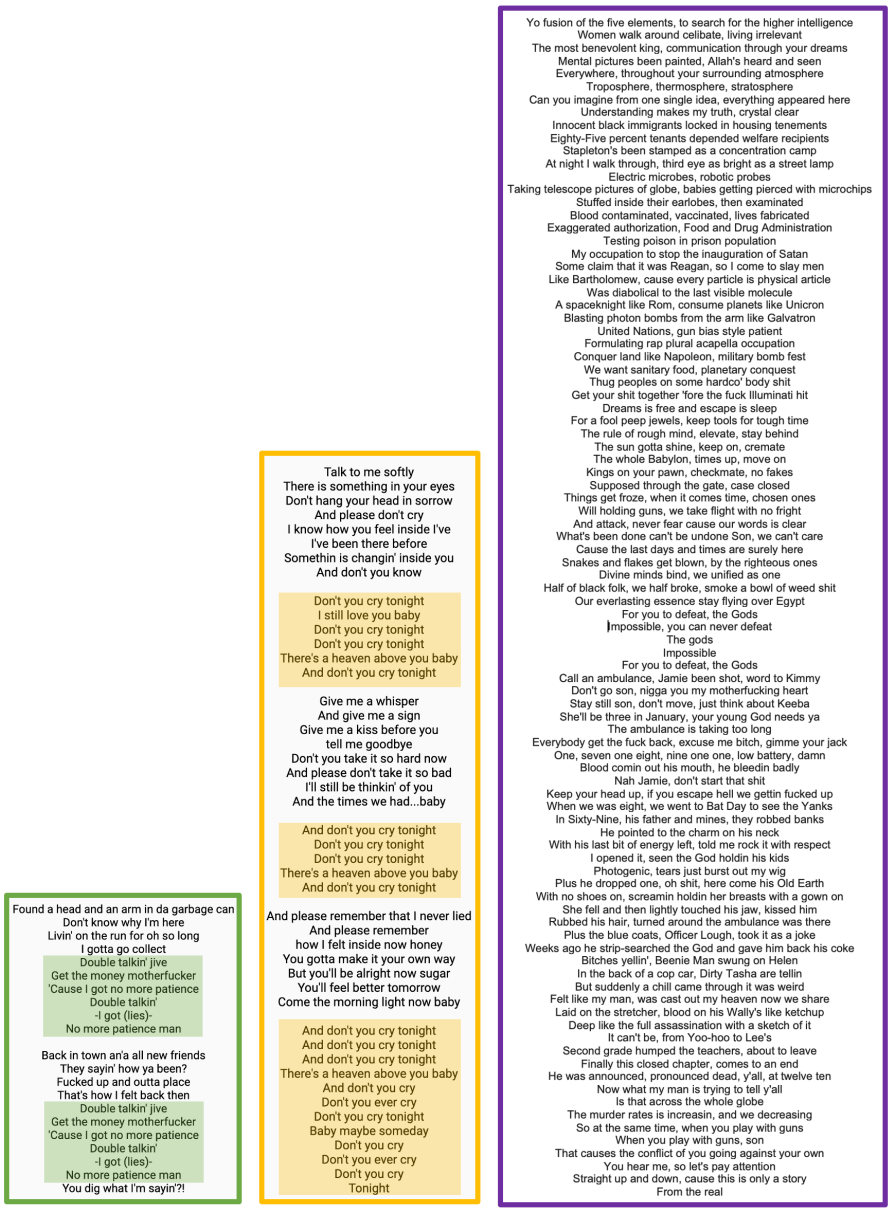


Figure 3.1: Visualization of different lyrics structures of two Rock songs (green and yellow) and one Rap song (violet). The transparent boxes in the green and yellow lyrics indicate the chorus. The green song is *Double Talkin' Jive*, the yellow is *Don't Cry* - both by Guns 'N Roses. The violet one is called *Impossible* by the Wu-Tang Clan.

detection remains challenging unless some complexity reduction strategies are applied - such as selecting a subset of songs belonging to musical genres characterized by repeating patterns (e.g. Country or Pop songs). Given the variability in the set of structure types provided in the literature according to different genres [16, 109] and the lack of large annotated datasets, attempts to achieve segment labelling have been rare. Furthermore, as the accuracy of lyrics segmentation in the state of the art is not fully satisfying yet [115], we focus on improving the performance in lyrics segmentation and leave the task of semantic labeling to the discussion (see Section 3.6).

The remainder of this Chapter is structured as follows: In Section 3.2 we detail our approach to lyrics segmentation: we explain why we made it **multimodal** and formulate our research questions and constributions. In Section 3.3 we setup and discuss the two experiments we conducted using either **unimodal** (text-based) lyrics representations or **bimodal** (text-audio-based) lyrics representation. We follow up with an error analysis in Section 3.4. We then position our work in the current state of the art in Section 3.5 and discuss the task of lyrics **segment labelling** in Section 3.6. Finally, in Section 3.7 we conclude with future research directions.

3.2 OUR APPROACH TO LYRICS SEGMENTATION

The task of **lyrics segmentation** is fundamental for full structure detection of song lyrics. While the final goal lies in detecting the building blocks (e.g. intro, verse, chorus) of a song text, this first step is a prerequisite to segment labelling when segment borders are not known. Thus, a method to automatically segment unsegmented song texts is needed to automate that first step.

Many heuristics can be imagined to find the segment borders. In our example (see Figure 3.2), separating the lyrics into segments of a constant length of four lines gives the correct segmentation. However, in another example, the segments can be of different length. This is to say that

enumerating heuristic rules is an open-ended task. Among previous works in the literature on lyrics structure analysis, [115] heavily exploited repeated patterns present in the lyrics to address this task, and it shows that this general class of pattern is very helpful with segment border detection.

For this reason, in this work we follow [115] by casting the lyrics segmentation task as **binary classification**. Let $L = \{a_1, a_2, ..., a_n\}$ be the lyrics of a song composed of n **lyrics lines** and $seg \subseteq (L, \mathbb{B})$ be a function that returns for each line $a_i \in L$ if it is the end of a segment. The task is to learn a classifier that approximates seg. At the learning stage, the ground truth segment borders are observed from segmented text as double line breaks. At the testing stage the classifier has to predict the now hidden segment borders.

In order to infer the lyrics structure, we develop a **Convolutional Neural Network-based model**. Our model architecture is detailed in Section 3.2.4. It detects segment boundaries by leveraging the repeated patterns in a song text that are conveyed by the Self-Similarity Matrices.

3.2.1 *The Need for Multimodality*

We fundamentally base our approach on (i) repeated patterns in lyrics and (ii) the intimate relation between lyrics and music. We first introduce a method relying on purely textual features which we call **unimodal lyrics segmentation**. While the method provides good results, it falls short in capturing the structure of the song in case there is no clear structure in the lyrics - when sentences are never repeated, or in the opposite case when they are always repeated. In such cases however, the structure may arise from the acoustic/audio content of the song, often from the melody representation. Therefore, as a second step, we extend our method by complementing the textual analysis with acoustic aspects. We perform lyrics segmentation on a synchronized text-audio representation of a song to benefit from both textual and audio features. We call this

method **bimodal lyrics segmentation**. Consequently, in our model, each lyrics line is naturally associated to a segment of audio. We define a **bimodal lyrics line** $a_i = (l_i, s_i)$ as a pair containing both the i-th text line l_i, and its associated audio segment s_i. In the case we only use the textual information, we model this as **unimodal lyrics lines**, i.e. $a_i = (l_i)$. This definition can be straightforwardly extended to more modalities, a_i then becomes a tuple containing time-synchronized information.

DETAILED EXAMPLE

To better understand the rational underlying the proposed multimodal approach, consider the segmentation of the Pop song depicted in Figure 3.2. The left side shows the lyrics and its segmentation into its structural parts: the horizontal green lines indicate the segment borders between the different lyrics segments. We can summarize the segmentation as follows: Verse_1-Verse_2-Bridge_1-Chorus_1-Verse_3-Bridge_2-Chorus_2-Chorus_3-Chorus_4-Outro. The middle of Figure 3.2 shows the repetitive structure of the lyrics. The exact nature of this structure representation is introduced later and is not needed to understand this introductory example. The crucial point is that the segment borders in the song text (green lines) coincide with highlighted rectangles in the chorus (the C_i) of the lyrics structure (middle). We find that in the verses (the V_i) and bridges (the B_i) highlighted rectangles are only found in the melody structure (right). The reason is that these verses have different lyrics, but share the same melody (analogous for the bridges). While the repetitive structure of the lyrics is an effective representation for lyrics segmentation, we believe that an enriched segment representation that also takes into account the audio of a song can improve segmentation models. While previous approaches relied on purely textual features for lyrics segmentation, showing the discussed limitations, we propose to perform lyrics segmentation on a synchronized text-audio representation of a song to benefit from both textual and audio features.

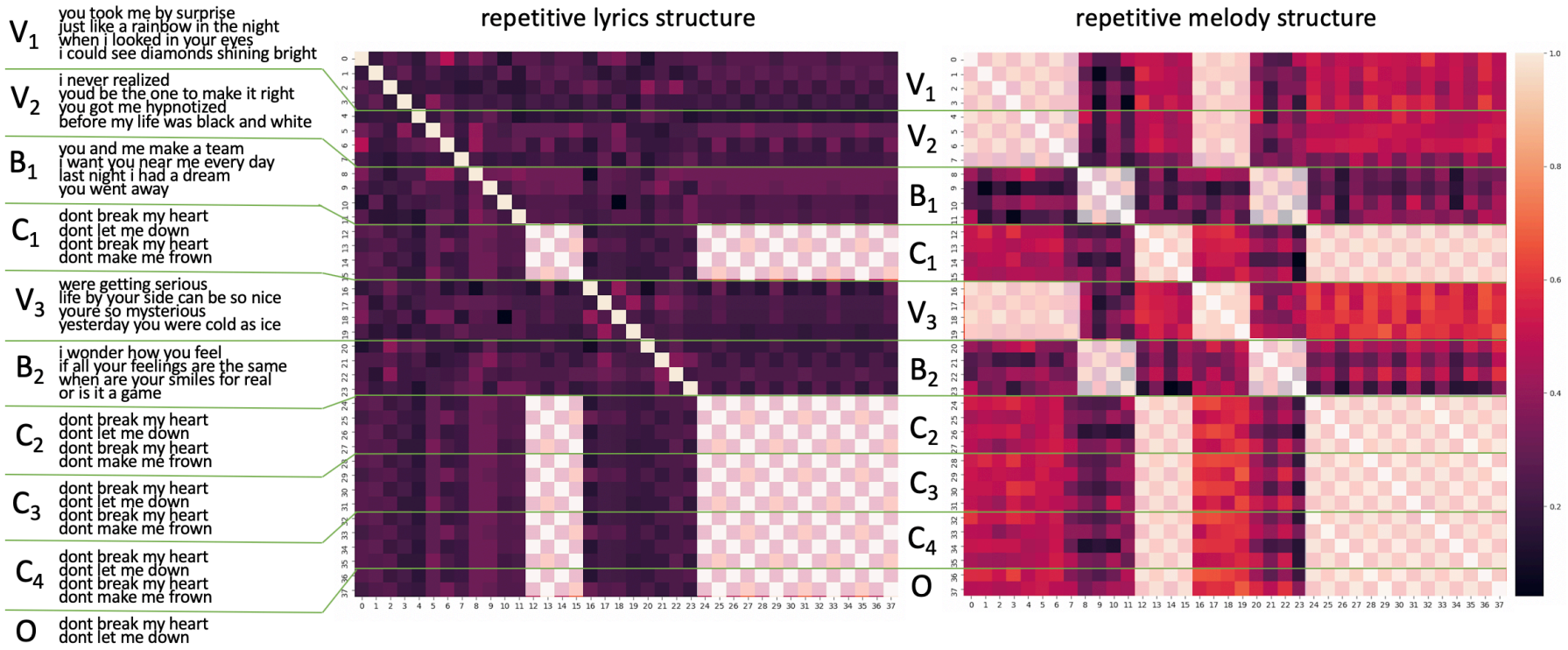


Figure 3.2: Lyrics (left) of a Pop song, the repetitive structure of the lyrics (middle), and the repetitive structure of the song melody (right). Lyrics segment borders (green lines) coincide with highlighted rectangles in lyrics structure and melody structure. (“Don‘t Break My Heart” by Den Harrow)

3.2.2 *Research Questions and Contributions*

We aim to answer the following research question: *given the text and audio of a song, can we learn to detect the lines delimiting segments in the song text?* This question is broken down into two sub questions: *1) given solely the song text, can we learn to detect the lines delimiting segments in the song?* and *2) do audio features - in addition to the text - boost the model performance on the lyrics segmentation task?*

To address these questions, this Chapter contains the following contributions.

- We introduce a convolutional neural network-based model that *i)* efficiently exploits the Self-Similarity Matrix representations (SSM) used in the state-of-the-art [115], and *ii)* can utilize traditional features alongside the SSMs (see this Section).
- We experiment with novel features that aim at revealing different properties of a song text, such as its phonetics and syntax. We evaluate this **unimodal** text-based approach on two standard datasets of English lyrics, the Music Lyrics Database and the WASABI corpus (see Section 3.3.1). We show that our proposed method can effectively detect the boundaries of music segments outperforming the state of the art, and is portable across collections of song lyrics of heterogeneous musical genre (see Section 3.3).
- We experiment with a **bimodal** lyrics representation (see Section 3.2.5) that incorporates audio features into our model. For this, we use a novel bimodal corpus (DALI, see Section 3.3.1) in which each song text is time-aligned to its associated audio. Our bimodal lyrics segmentation performs significantly better than the unimodal approach. We investigate which text and audio features are the most relevant to detect lyrics segments and show that the text and audio modalities complement each other. We perform an ablation test to find out to what extent our method relies on

the alignment quality of the lyrics-audio segment representations (see Section 3.3).

3.2.3 *Self-Similarity Matrices*

We produce Self-Similarity Matrices (SSMs) based on bimodal lyrics lines $a_i = (l_i, s_i)$ in order to capture repeated patterns in the text line l_i as well as its associated audio segment s_i. SSMs have been previously used in the literature to estimate the structure of music [31, 49] and lyrics [44, 115]. Given a song consisting of bimodal lines $\{a_1, a_2, ..., a_n\}$, a Self-Similarity Matrix $SSM_M \in \mathbb{R}^{n \times n}$ is constructed, where each element is set by computing a similarity measure between the two corresponding elements $(SSM_{\mathrm{M}})_{ij} = \mathrm{sim}_{\mathrm{M}}(x_i, x_j)$. We choose x_i, x_j to be elements from the same modality, i.e. they are either both lyrics lines (l_i) or both audio segments (s_i) associated to lyrics lines. $\mathrm{sim}_{\mathrm{M}}$ is a similarity measures that compares two elements of the same modality to each other. In our experiments, this is either a text-based or an audio-based similarity (see Section 3.3.2). As a result, SSMs constructed from a text-based similarity highlight distinct patterns of the text, revealing the underlying structure (see Figure 3.2, middle). Analogously, SSMs constructed from an audio-based similarity highlight distinct patterns of the audio (see Figure 3.2, right). In the unimodal case, we compute SSMs from only one modality: either text lines l_i or audio segments s_i.

There are two common patterns that were investigated in the literature: diagonals and rectangles. Diagonals parallel to the main diagonal indicate sequences that repeat and are typically found in a chorus. Rectangles, on the other hand, indicate sequences in which all the lines are highly similar to one another. Both of these patterns were found to be indicators of segment borders.

3.2.4 *Convolutional Neural Network-based Model*

Lyrics segments manifest themselves in the form of distinct patterns in the SSM. In order to detect these patterns efficiently, we introduce the Convolutional Neural Network (CNN) architecture which is illustrated in Figure 3.3. The model predicts for each lyrics line if it is segment ending. For each of the n lines of a song text the model receives patches (see Figure 3.3, step A) extracted from SSMs $\in \mathbb{R}^{n \times n}$ and centered around the line: $\text{input}_i = \{P_i^1, P_i^2, ..., P_i^c\} \in \mathbb{R}^{2w \times n \times c}$, where c is the number of SSMs or number of channels and w is the window size. To ensure the model captures the segment-indicating patterns regardless of their location and relative size, the input patches go through two convolutional layers (see Figure 3.3, step B) [51], using filter sizes of $(w+1) \times (w+1)$ and $1 \times w$, respectively. By applying max pooling after both convolutions each feature is downsampled to a scalar. After the convolutions, the resulting feature vector is concatenated with the line-based features (see Figure 3.3, step C) and goes through a series of densely connected layers. Finally, the $softmax$ is applied to produce probabilities for each class (border/not border) (see Figure 3.3, step D). The model is trained with supervision using binary cross-entropy loss between predicted and ground truth segment border labels (see Figure 3.3, step E). Note that while the patch extraction is a local process, the SSM representation captures global relationships, namely the similarity of a line to all other lines in the lyrics.

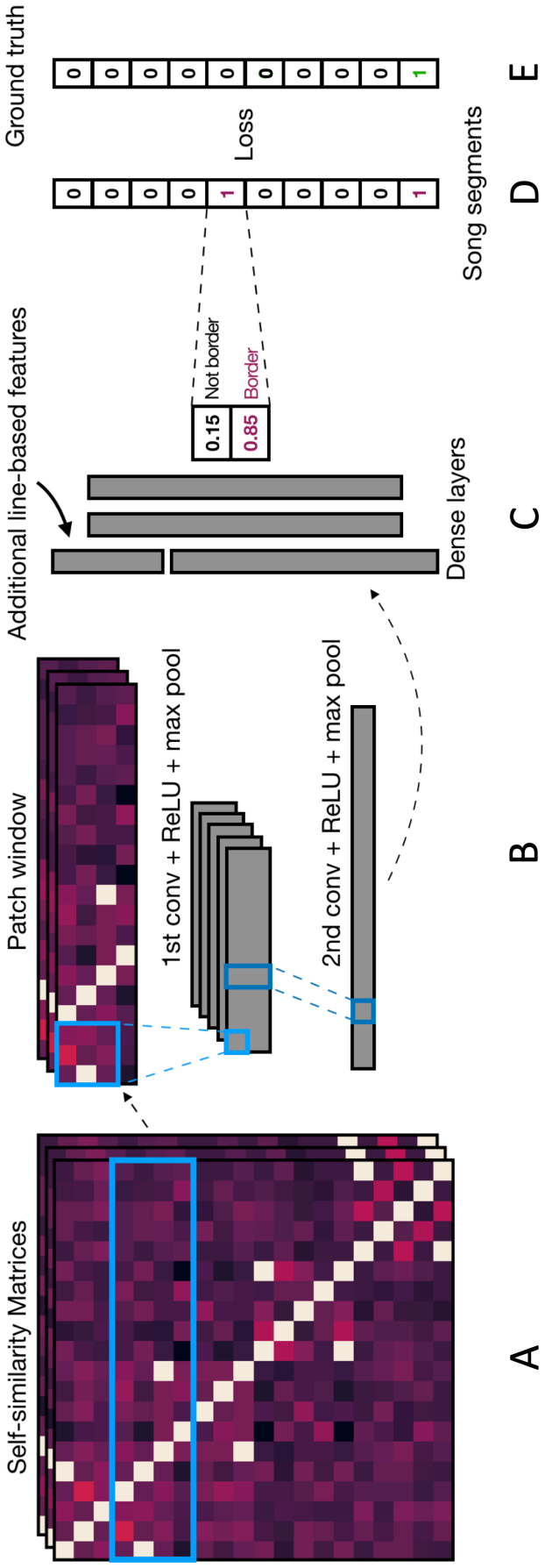


Figure 3.3: Convolutional Neural Network-based model inferring lyrics segmentation.

3.2.5 *Bimodal Lyrics Lines*

To perform lyrics segmentation on a bimodal text-audio representation of a song to benefit from both textual and audio features, we use a corpus where the annotated lyrics ground truth (segment borders) is synchronized with the audio. This bimodal dataset is described in Section 3.3.1. We focus solely on the audio extracts that have singing voice, as only they are associated to the lyrics. For that let t_i be the time interval of the (singing event of) text line l_i in our synchronized text-audio corpus. Then, a bimodal lyrics line $a_i = (l_i, s_i)$ consists of both a text line l_i (the text line during t_i) and its associated audio segment s_i (the audio segment during t_i). As a result, we have the same number of text lines and audio segments. While the textual information l_i can be used directly to produce SSMs, the complexity of the raw audio signal prevents it from being used as direct input of our system. Instead, it is common to extract features from the audio that highlight some aspects of the signal that are correlated with the different musical dimensions. Therefore, we describe each audio segment s_i as set of different time vectors. Each frame of a vector contains information of a precise and small time interval. The size of each audio frame depends on the configuration of each audio feature. We call an audio segment s_i *featurized* by a feature f if f is applied to all frames of s_i. For our bimodal segment representation we featurize each s_i with one of the following features:

- **Mel-frequency cepstral coefficients ($\boldsymbol{mfcc} \in \mathbb{R}^{14}$):** these coefficients [32] emphasize parts of the signal that are related with our understanding of the musical timbre. They have proven to be very efficient in a large range of audio applications.
- **Chroma feature ($\boldsymbol{chr} \in \mathbb{R}^{12}$):** this feature [50] describes the harmonic information of each frame by computing the *presence* of the twelve different notes.

3.3 EXPERIMENTS

In this Section we describe the setup we have used to experiment our approach. We first define the datasets, we then lay out the similarity measures for the SSM computation. Finally, we specify the parameters of our different experimental models.

3.3.1 *Datasets*

We used two kinds of corpora in our segmentation experiments. First, the WASABI Song Corpus (see Section 2.3) and the Music Lyrics Database (introduced below) contain **textual representations** of the lyrics. Second, to effectively test our bimodal approach we experiment on DALI, a dataset which contains **bimodal representations** of the lyrics where text and audio are synchronized.

The Music Lyrics Database (MLDB) V.1.2.7[2] is a proprietary lyrics corpus of popular songs of diverse genres. We use MLDB as it has been used before by the state of the art [115]. To facilitate a close comparison with their work, we also use the same configuration, considering only the 103k English song texts that have five or more segments[3] and using the same training, development and test indices (60%-20%-20% split). From the WASABI Song Corpus we sample the English song texts that contain at least five segments, resulting in 744k lyrics.

The DALI corpus[4] [74] contains synchronized lyrics-audio representations on different levels of granularity: syllables, words, lines and segments. The alignment quality of the text-audio representations differs and we partition the corpus according to the alignment quality. This way, we can test the influence of the alignment quality on the outcome of our segmentation experiment. We describe the partitioning process in the following.

2 http://www.odditysoftware.com/page-datasales1.htm
3 92% of the remaining song texts count between six and twelve segments
4 https://github.com/gabolsgabs/DALI

PARTITIONING THE DALI DATASET

DALI was created by joining two datasets: (1) a corpus for karaoke singing[5] (Animux) which contains alignments between lyrics and audio on the syllable level and (2) a subset of the WASABI Song Corpus of lyrics that belong to the same songs than the lyrics in Animux. Note that corresponding lyrics in WASABI Song Corpus can differ from those in Animux to some extent. Also, in Animux there is no annotation of segments. DALI provides estimated segments for Animux lyrics, projected from the ground truth segments from WASABI Song Corpus. For example, Figure 3.4 shows on the left side the lyrics lines as given in Animux. The right side shows the lyrics lines given in WASABI Song Corpus as well as the ground truth lyrics segments. The left side shows the estimated lyrics segments in Animux. Note how the lyrics in WASABI Song Corpus have one segment more, as the segment W_3 has no counter part in Animux.

Based on the requirements for our task, we derive a measure to assess how well the estimated Animux segments correspond / align to the ground truth WASABI Song Corpus segments. Since we will use the WASABI Song Corpus segments as ground truth labels for supervised learning, we need to make sure, the Animux lines (and hence audio information) actually belong to the aligned segment. As only for the Animux lyrics segments we have aligned audio features and we want to consistently use audio features in our segment representations, we make sure that every Animux segment has a counterpart WASABI Song Corpus segment (see Figure 3.4, $A_0 \sim W_0$, $A_1 \sim W_1$, $A_2 \sim W_2$, $A_3 \sim W_4$). On the other hand, we allow WASABI Song Corpus segments to have no corresponding Animux segments (see Figure 3.4, W_3). We further do not impose constraints on the order of appearance of segments in Animux segmentations vs. WASABI Song Corpus segmentations, to allow for possible rearrangements in the order of corresponding segments. With these considerations, we formulate a measure of alignment quality that is tailored to our task of

5 from http://usdb.animux.de/

Corpus name	Alignment quality	Song count
Q^+	high (90-100%)	1048
Q^0	med (52-90%)	1868
Q^-	low (0-52%)	1868
full dataset	-	4784

Table 3.1: The DALI dataset partitioned by alignment quality

bimodal lyrics segmentation. Let A, W be segmentations, where $A = A_0A_1...A_x$ and the A_i are Animux segments and $W = W_0W_1...W_y$ with WASABI Song Corpus lyrics segments W_i. Then the alignment quality between the segmentations A, W is composed from the similarities of the best-matching segments. Using string similarity sim_{str} as defined in Section 3.3.2, we define the alignment quality $Qual$ as follows:

$$\begin{aligned} Qual(A, W) &= Qual(A_0A_1...A_x,\ W_0W_1...W_y) \\ &= \min_{0 \le i \le x} \{\ \max_{0 \le j \le y} \{\ \text{sim}_{\text{str}}(A_i, W_j)\ \}\ \} \end{aligned}$$

In order to test the impact of $Qual$ on the performance of our lyrics segmentation algorithm, we partition the DALI corpus into parts with different $Qual$. Initially, DALI consists of 5358 lyrics that are synchronized to their audio track. Like in previous publications [44, 115], we ensure that all song texts contain at least 5 segments. This constraint reduces the number of tracks used by us to 4784. We partition the 4784 tracks based on their $Qual$ into high (Q^+), med (Q^0), and low (Q^-) alignment quality datasets. Table 3.1 gives an overview over the resulting dataset partitions. The Q^+ dataset consists of 50842 lines and 7985 segment borders and has the following language distribution: 72% English, 11% German, 4% French, 3% Spanish, 3% Dutch, 7% other languages.

The central input features used by our Convolutional Neural Network-based model are the different SSMs. Therefore, the choice of similarities used to produce the SSMs is essential to the approach. We experiment with both text-

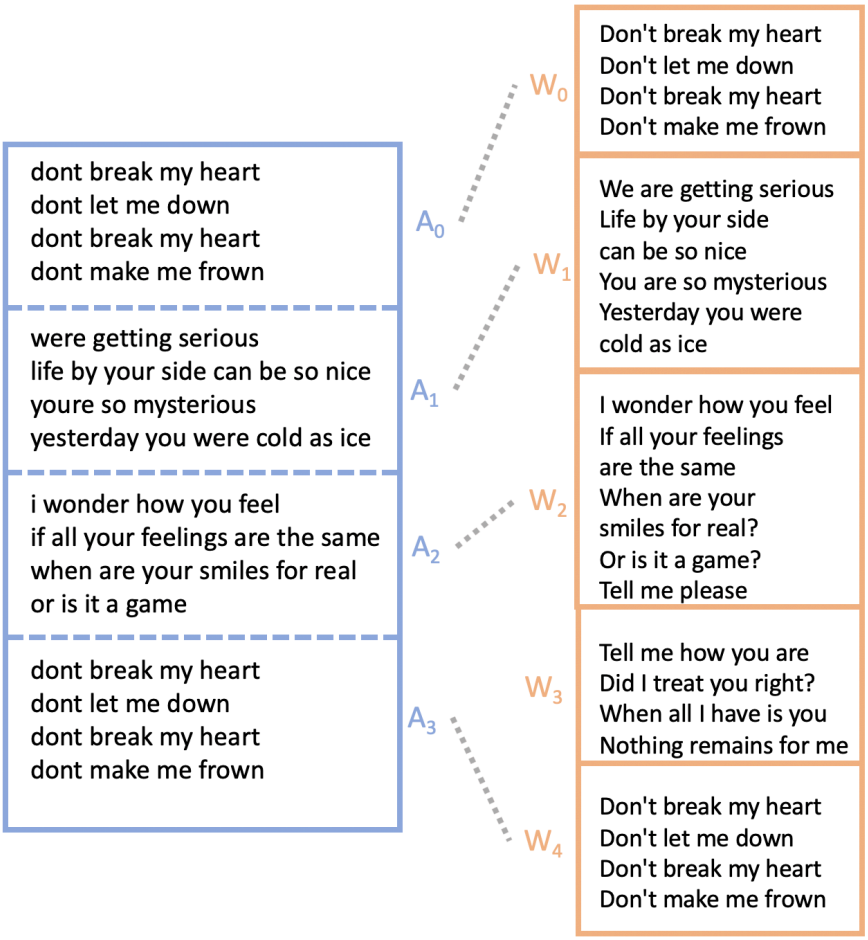


Figure 3.4: Lyrics lines and estimated lyrics segments in Animux (left). Lyrics lines and ground truth lyrics segments in WASABI (right) for the song ("Don't Break My Heart" by Den Harrow)

based and audio-based similarities, these are defined in Section 3.3.2. We then present the experiments using unimodal lyrics lines (text only) in Section 3.3.3 and bimodal lyrics lines (text and audio) in Section 3.3.4, respectively.

3.3.2 *Similarity Measures*

We produce SSMs based on three line-based text similarity measures. We further add audio-based similarities - the crucial ingredient that makes our approach multimodal. In the following, we define the text-based and audio-based similarities used to compute the SSMs.

Text similarities: given the text lines of the lyrics, we compute different similarity measures, based on either their characters, their phonetics or their syntax.

- **String similarity ($\mathbf{sim_{str}}$):** a normalized Levenshtein string edit similarity between the characters of two lines of text [65]. This has been widely used - e.g. [44, 115].
- **Phonetic similarity ($\mathbf{sim_{phon}}$):** a simplified phonetic representation of the lines computed using the *Double Metaphone Search Algorithm* [99]. When applied to the textual snippets "i love you very much" and "i'l off you vary match" it returns the same result: "ALFFRMX". This algorithm was developed to capture the similarity of similar sounding words even with possibly very dissimilar orthography. We translate the text lines into this "phonetic language" and then compute sim_{str} between them.
- **Lexico-syntactical similarity ($\mathbf{sim_{lsyn}}$):** this measure, initially proposed in [43], combines lexical with syntactical similarity. sim_{lex} captures the similarity between text lines such as "Look into my eyes" and "I look into your eyes": these are partially similar on a lexical level and partially similar on a syntactical level. Given two lines x, y lexico-syntactical similarity is defined as: $sim_{lsyn}(x, y) = sim^2_{lex}(x, y) + (1 - sim_{syn}) \cdot sim_{syn}(\hat{x}, \hat{y})$,

where sim_{lex} is the overlap of the bigrams of words in x and y, and sim_{syn} is the overlap of the bigrams of pos tags in $\hat{x}, \hat{y}$, the remaining tokens that did not overlap on a word level.

Audio similarities: There are several alternatives to measure the similarity between two audio sequences (e.g. mfcc sequences) of possibly different lengths, among which Dynamic Time Warping T_d is the most popular one in the Music Information Retrieval community. Given bimodal lyrics lines a_u, a_v (as defined in Section 3.2.5), we compare two audio segments s_u and s_v that are featurized by a particular audio feature (mfcc, chr) using T_d:

$$T_d(i,j) = d(s_u(i), s_v(j)) + \min \left\{ \begin{array}{c} T_d(i-1,j), \\ T_d(i-1,j-1), \\ T_d(i,j-1) \end{array} \right\}$$

T_d must be parametrized by an inner distance d to measure the distance between the frame i of s_u and the frame j of s_v. Depending on the particular audio feature s_u and s_v are featurized with, we employ a different inner distance as defined below. Let m be the length of the vector s_u and n be the length of s_v. Then, we compute the minimal distance between the two audio sequences as $T_d(m,n)$ and normalize this by the length r of the shortest alignment path between s_u and s_v to obtain values in [0,1] that are comparable to each other. We finally apply $\lambda x.(1-x)$ to turn the distance T_d into a similarity measure S_d:

$$S_d(s_u, s_v) = 1 - T_d(m,n) \cdot r^{-1}$$

Given bimodal lyrics lines a_i, we now define similarity measures between audio segments s_i that are featurized by a particular audio feature presented previously (mfcc, chr) based on our similarity measure S_d:

- **MFCC similarity ($\mathbf{sim_{mfcc}}$):** S_d between two audio segments featurized by the mfcc feature. As inner distance we use the cosine distance:
  $d(x,y) = x \cdot y \cdot (\|x\| \cdot \|y\|)^{-1}$

- **Chroma similarity ($\mathbf{sim_{chr}}$):** S_d between two audio segments featurized by the chroma feature; using cosine distance as inner distance

3.3.3 *Unimodal Lyrics Segmentation*

In our first experiment we represent song texts via unimodal lyrics lines (textual representation) and experiment on the Music Lyrics Database and the WASABI Song Corpus. We compare to the state of the art [115] and successfully reproduce their best features to validate their approach. Two groups of features are used in the replication: repeated pattern features (RPF) extracted from SSMs and n-grams extracted from text lines. The RPF basically act as hand-crafted image filters that aim to detect the edges and the insides of diagonals and rectangles in the SSM.

Then, our own models are neural networks as described in Section 3.2.4, that use as features SSMs and two line-based features: the line length and n-grams. For the line length, we extracted the character count from each line, a simple proxy of the orthographic shape of the song text. Intuitively, segments that belong together tend to have similar shapes. Similarly to [115]'s term features we extracted those n-grams from each line that are most indicative for segment borders: using the tf-idf weighting scheme, we extracted n-grams that are typically found left or right from the segment border, varied n-gram lengths and also included indicative part-of-speech tag n-grams. This resulted in 240 term features in total. The most indicative words at the start of a segment were: {ok, lately, okay, yo, excuse, dear, well, hey}. As segment-initial phrases we found: {Been a long, I've been, There's a, Won't you, Na na na, Hey, hey}. Typical words ending a segment were: {..., .., !, ., yeah, ohh, woah. c'mon, wonderland}. And as segment-final phrases we found as most indicative: {yeah!, come on!, love you., !!!, to you., with you., check it out, at all., let's go, ...}

In this experiment we consider only SSMs made from text-based similarities; we note this in the model name as CNN_{text}. We further name a CNN model by the set of SSMs

that it uses as features. For example, the model $\text{CNN}_{\text{text}}\{\text{str}\}$ uses as only feature the SSM made from string similarity sim_{str}, while the model $\text{CNN}_{\text{text}}\{\text{str, phon, lsyn}\}$ uses three SSMs in parallel (as different input channels), one from each similarity.

For convolutional layers we empirically set $w_{\text{size}} = 2$ and the amount of features extracted after each convolution to 128. Dense layers have 512 hidden units. We have also tuned the learning rate (negative degrees of 10), the dropout probability with increments of 0.1. The batch size was selected from the beginning to be 256 to better saturate our GPU. The CNN models were implemented using Tensorflow.

In addition to our reimplementation of the RPF method described previously, we implement two baselines. The random baseline guesses for each line independently if it is a segment border (with a probability of 50%) or not. The line length baseline uses as only feature the line length in characters and is trained using a logistic regression classifier.

For comparison with the state of the art, we use as a first dataset the same they used, the MLDB (see Section 3.3.1). To test the system portability to bigger and more heterogeneous data sources, we further experimented our method on the WASABI corpus (see Section 3.3.1). In order to test the influence of genre on classification performance, we aligned MLDB to WASABI as the latter provides genre information. Song texts that had the exact same title and artist names (ignoring case) in both data sets were aligned. This rather strict filter resulted in an amount of 58567 (57%) song texts with genre information in MLDB. Table 3.3 shows the distribution of the genres in MLDB song texts. We then tested our method on each genre separately, to test our hypothesis that classification is harder for some genres in which almost no repeated patterns can be detected (as Rap songs). To the best of our knowledge, previous work did not report on genre-specific results.

In this work we did not normalize the lyrics in order to rigorously compare our results to [115]. We estimate the proportion of lyrics containing tags such as *Chorus* to be marginal (0.1-0.5%) in the MLDB corpus. When applying

our methods for lyrics segmentation to lyrics found online, an appropriate normalization method should be applied as a pre-processing step. For details on such a normalization procedure we refer the reader to [43], Section 2.1.

Evaluation metrics are Precision (P), Recall (R), and f-score (F_1). Significance is tested with a permutation test [84], and the p-value is reported.

RESULTS AND DISCUSSION

The results on the MLDB dataset are shown in Table 3.2. We start by measuring the performance of our replication of [115]'s approach. This reimplementation exhibits 56.3% F_1, similar to the results reported in the original paper (57.7%). The divergence could be attributed to a different choice of hyperparameters and feature extraction code. Much weaker baselines were explored as well. The random baseline resulted in 18.6% F_1, while the usage of simple line-based features, such as the line length (character count), improves this to 25.4%.

The best CNN-based model, $\text{CNN}_{\text{text}}\{\text{str, phon, lsyn}\}$ + n-grams, outperforms all our baselines reaching 67.4% F_1, 8.2pp better than the results reported in [115]. When performing the permutation test of this model against all other models we find that, in every case, the performance difference is statistically significant ($p < .05$).

Subsequent feature analysis revealed that the model $\text{CNN}_{\text{text}}\{\text{str}\}$ is by far the most effective. The $\text{CNN}_{\text{text}}\{\text{lsyn}\}$ model exhibits much lower performance, despite using a much more complex feature. We believe the lexico-syntactical similarity is much noisier as it relies on n-grams and PoS tags, and thus propagates error from the tokenizers and PoS taggers. The $\text{CNN}_{\text{text}}\{\text{phon}\}$ exhibits a small but measurable performance decrease from $\text{CNN}_{\text{text}}\{\text{str}\}$, possibly due to phonetic features capturing similar regularities, while also depending on the quality of preprocessing tools and the rule-based phonetic algorithm being relevant for our song-based dataset. The $\text{CNN}_{\text{text}}\{\text{str, phon, lsyn}\}$ model that combines the different textual SSMs yields a performance comparable to $\text{CNN}_{\text{text}}\{\text{str}\}$.

Model	*Features*	P	R	F_1
Random baseline	n/a	18.6	18.6	18.6
Line length baseline	text line length	16.7	52.8	25.4
Handcrafted filters	RPF (our replication)	48.2	67.8	56.3
	RPF [115]	56.1	59.4	57.7
	RPF + n-grams	57.4	61.2	59.2
CNN_{text}	{str}	70.4	63.0	66.5
	{phon}	75.9	55.6	64.2
	{lsyn}	74.8	50.0	59.9
	{str, phon, lsyn}	74.1	60.5	66.6
	{str, phon, lsyn} + n-grams	72.1	63.3	**67.4**

Table 3.2: Results with unimodal lyrics lines on MLDB dataset in terms of Precision (P), Recall (R) and f-score (F_1) in %.

In addition, we test the performance of several line-based features on our dataset. Most notably, the n-grams feature provides a significant performance improvement producing the best model. Note that adding the line length feature to any CNN_{text} model does not increase performance.

To show the portability of our method to bigger and more heterogeneous datasets, we ran the CNN model on the WASABI dataset (as described in Section 2.3), obtaining results that are very close to the ones obtained for the MLDB dataset: precision: 67.4% for precision, 67.3% recall, and 67.4% f-score using the CNN_{text}{str} model.

Results differ significantly based on genre. We split the MLDB dataset with genre annotations into training and test, trained on all genres, and tested on each genre separately. In Table 3.3 we report the performances of the CNN_{text}{str} on lyrics of different genres. Songs belonging to genres such as Country, Rock or Pop, contain recurrent structures with repeating patterns, which are more easily detectable by the CNN_{text} algorithm. Therefore, they show significantly better performance. On the other hand, the performance on genres such as Hip Hop or Rap, is much worse.

Genre	*Lyrics[#]*	*P*	*R*	F_1
Rock	6011	73.8	57.7	64.8
Hip Hop	5493	71.7	43.6	<u>54.2</u>
Pop	4764	73.1	61.5	66.6
RnB	4565	71.8	60.3	65.6
Alternative Rock	4325	76.8	60.9	67.9
Country	3780	74.5	66.4	**70.2**
Hard Rock	2286	76.2	61.4	67.7
Pop Rock	2150	73.3	59.6	65.8
Indie Rock	1568	80.6	55.5	65.6
Heavy Metal	1364	79.1	52.1	63.0
Southern Hip Hop	940	73.6	34.8	<u>47.0</u>
Punk Rock	939	80.7	63.2	**70.9**
Alternative Metal	872	77.3	61.3	68.5
Pop Punk	739	77.3	68.7	**72.7**
Gangsta Rap	435	73.6	35.2	<u>47.7</u>
Soul	603	70.9	57.0	63.0

Table 3.3: Results with unimodal lyrics lines. $\text{CNN}_{\text{text}}\{str\}$ model performances across musical genres in the MLDB dataset in terms of Precision (P), Recall (R) and F_1 in %. Underlined are the performances on genres with less repetitive text. Genres with highly repetitive structure are in bold.

3.3.4 *Bimodal Lyrics Segmentation*

In the second experiment we represent song texts via bimodal lyrics lines (text+audio) and experiment on the DALI corpus. In order to test our hypotheses which text and audio features are most relevant to detect segment boundaries, and whether the text and audio modalities complement each other, we compare different types of models: baselines, text-based models, audio-based models, and finally bimodal models that use both text and audio features. We provide the following baselines: the random baseline guesses for each line independently if it is a segment border (with a probability of 50%) or not. The line length baselines use as feature only the line length in characters (text-based model) or milliseconds (audio-based model) or both, respectively. These baselines are trained using a logistic regression classifier. All other models are CNNs using the architecture described previously and use as features SSMs made from different textual or audio similarities as described in Section 3.3.2. The CNN-based models that use purely textual features (str) are named CNN_{text}, while the CNN-based models using purely audio features (mfcc, chr) are named $\text{CNN}_{\text{audio}}$. Lastly, the CNN_{mult} models are multimodal in the sense that they use combinations of textual and audio features. We name a CNN model by its modality (text, audio, mult) as well as by the set of SSMs that it uses as features. For example, the model $\text{CNN}_{\text{mult}}\{\text{str, mfcc}\}$ uses as textual feature the SSM made from string similarity sim_{str} and as audio feature the SSM made from mfcc similarity sim_{mfcc}.

As dataset we use the Q^+ partition of the DALI, i.e. the partition which has the highest alignment quality. See Section 3.3.1 for explanation of the DALI partitioning. We split the data randomly into training and test sets using the following scheme: considering that the DALI dataset is relatively small, we average over two different 5-fold cross-validations. We prefer this sampling strategy for our small dataset over a more common 10-fold cross-validation as it avoids the test set to become too small.

RESULTS AND DISCUSSION

The results are depicted in Table 3.4. The random baseline and the different line length baselines reach a performance of 15.5%-33.5% F_1. Interestingly, the audio-based line length (33.5% F_1) is more indicative of the lyrics segmentation than the text-based line length (25.0% F_1)[6].

The model $\text{CNN}_{\text{text}}\{\text{str}\}$ performs with 70.8% F_1 similarly to the $\text{CNN}_{\text{text}}\{\text{str}\}$ model from the first experiment (66.5% F_1). The models use the exact same SSM_{str} feature and hyperparameters, but another lyrics corpus (DALI instead of MLDB). We believe that as DALI was assembled from karaoke singing instances, it likely contains more repetitive song texts that are easier to segment using the employed method. Note that the DALI dataset is too small to allow a genre-wise comparison as we did in the previous experiment using the MLDB dataset.

The $\text{CNN}_{\text{audio}}$ models perform similarly well than the CNN_{text} models. $\text{CNN}_{\text{audio}}\{\text{mfcc}\}$ reaches 65.3% F_1, while $\text{CNN}_{\text{audio}}\{\text{chr}\}$ results in 63.9% F_1. The model $\text{CNN}_{\text{audio}}$ {mfcc, chr} performs with 70.4% F_1 significantly ($p < .001$) better than the models that use only one of the features. As the mfcc feature models timbre and instrumentation, whilst the chroma feature models melody and harmony, they provide complementary information to the $\text{CNN}_{\text{audio}}$ model which increases its performance.

Most importantly, the CNN_{mult} models combining text- with audio-based features constantly outperform the CNN_{text} and $\text{CNN}_{\text{audio}}$ models. $\text{CNN}_{\text{mult}}\{\text{str, mfcc}\}$ and CNN_{mult} {str, chr} achieve a performance of 73.8% F_1 and 74.5% F_1, respectively - this is significantly ($p < .001$) higher compared to the 70.8% (70.4%) F_1 of the best CNN_{text} ($\text{CNN}_{\text{audio}}$) model. Finally, the overall best performing model is a combination of the best CNN_{text} and $\text{CNN}_{\text{audio}}$ models and delivers 75.3% F_1. $\text{CNN}_{\text{mult}}\{\text{str, mfcc, chr}\}$ is the only model to significantly ($p < .05$) outperform all other models in all three evaluation metrics: precision, recall, and F_1. Note, that all CNN_{mult} models outperform all

6 Note that adding line length features to any CNN-based model does not increase performance.

Model	*Features*	P	R	F_1
Random baseline	n/a	15.7	15.7	15.7
Line length baselines	text length	16.6	51.8	25.0
	audio length	22.7	63.8	33.5
	text length + audio length	22.6	63.0	33.2
CNN_{text}	{str}	78.7	64.2	70.8
$\text{CNN}_{\text{audio}}$	{mfcc}	79.3	55.9	65.3
	{chr}	76.8	54.7	63.9
	{mfcc, chr}	79.2	63.8	70.4
CNN_{mult}	{str, mfcc}	80.6	**69.0**	73.8
	{str, chr}	82.5	**69.0**	74.5
	{str, mfcc, chr}	**82.7**	**70.3**	**75.3**

Table 3.4: Results with multimodal lyrics lines on the Q^+ dataset in terms of Precision (P), Recall (R) and F_1 in %. Note that the CNN_{text}{str} model is the same configuration as in Table 2, but trained on different dataset.

CNN_{text} and $\text{CNN}_{\text{audio}}$ models significantly ($p < .001$) in recall.

We perform an ablation test on the alignment quality. For this, we train CNN-based models with those feature sets that performed best on the Q^+ part of DALI. For each modality (text, audio, mult), i.e. CNN_{text}{str}, $\text{CNN}_{\text{audio}}${mfcc, chr}, and CNN_{mult}{str, mfcc, chr}, we train a model for each feature set on each partition of DALI (Q^+, Q^0, Q^-). We always test our models on the same alignment quality they were trained on. The alignment quality ablation results are depicted in Table 3.5. We find that independent of the modality (text, audio, mult.), all models perform significantly ($p < .001$) better with higher alignment quality. The effect of modality on segmentation performance (F_1) is as follows: on all datasets we find CNN_{mult}{str, mfcc, chr} to significantly ($p < .001$) outperform both CNN_{text}{str} and $\text{CNN}_{\text{audio}}${mfcc, chr}. Further, CNN_{text}{str} significantly ($p < .001$) outperforms $\text{CNN}_{\text{audio}}${mfcc, chr} on the Q^0 and Q^- dataset, whereas this does not hold on the Q^+ dataset ($p \geq .05$).

Dataset	*Model*	*Features*	*P*	*R*	F_1
Q^+	CNN_{text}	{str}	78.7	64.2	70.8
	CNN_{audio}	{mfcc, chr}	79.2	63.8	70.4
	CNN_{mult}	{str, mfcc, chr}	82.7	70.3	75.3
Q^0	CNN_{text}	{str}	73.6	54.5	62.8
	CNN_{audio}	{mfcc, chr}	74.9	48.9	59.5
	CNN_{mult}	{str, mfcc, chr}	75.8	59.4	66.5
Q^-	CNN_{text}	{str}	67.5	30.9	41.9
	CNN_{audio}	{mfcc, chr}	66.1	24.7	36.1
	CNN_{mult}	{str, mfcc, chr}	68.0	35.8	46.7

Table 3.5: Results with multimodal lyrics lines for the alignment quality ablation test on the datasets Q^+, Q^0, Q^- in terms of Precision (P), Recall (R) and F_1 in %.

3.4 ERROR ANALYSIS

An SSM for a Rap song is depicted in Figure 3.5. As common for Rap song texts, there is no chorus (diagonal stripe parallel to main diagonal). However, there is a highly repetitive musical state from line 18 to 21 indicated by the corresponding rectangle in the SSM spanning from (18,18) to (21,21). As texts in this genre are less repetitive, the SSM-based features are usually less reliable to determine a song's structure. Moreover, when returning to the introductory example in Figure 3.2, we observe that verses (the V_i) and bridges (the B_i) are not detectable when looking at the text representation only (see Figure 3.2, middle). The reason is that these verses have different lyrics. However, as these parts share the same melody, highlighted rectangles are visible in the melody structure.

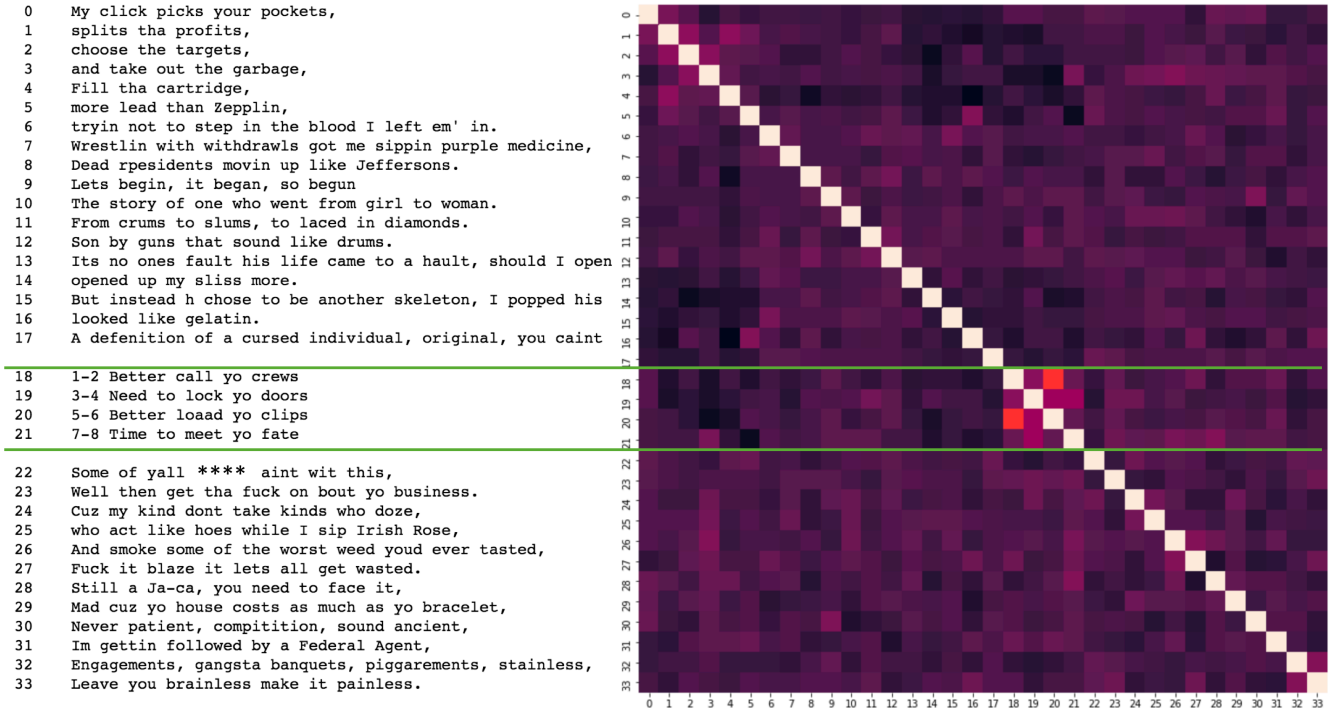


Figure 3.5: SSM computed from textual similarity sim_{str}. (“Meet Your Fate” by Southpark Mexican, MLDB-ID: 125521)

Indeed, we found our bimodal segmentation model to produce significantly ($p < .001$) better segmentations (75.3% F_1) compared to the purely text-based (70.8% F_1) and audio-based models (70.4% F_1). The increase in F_1 stems from both increased precision and recall. The model increase in precision is observed as CNN_{mult} often produces less false positive segment borders, i.e. the model delivers less noisy results. We observe an increase in recall in two ways: first, CNN_{mult} sometimes detects a combination of the borders detected by CNN_{text} and $\text{CNN}_{\text{audio}}$. Secondly, there are cases where CNN_{mult} detects borders that are not recalled in either of CNN_{text} or $\text{CNN}_{\text{audio}}$.

Segmentation algorithms that are based on exploiting patterns in an SSM, share a common limitation: non-repeated segments are hard to detect as they do not show up in the SSM. Note, that such segments are still occasionally detected indirectly when they are surrounded by repeated segments. Furthermore, a consecutively repeated pattern such as C_2-C_3-C_4 in Figure 3.2 is not easily segmentable as it could potentially also form one ($C_2C_3C_4$) or two (C_2-C_3C_4 or C_2C_3-C_4) segments. Another problem is that of inconsistent classification inside of a song: sometimes, patterns in the SSM that look the same to the human eye are classified differently. Note, however that on the pixel level there is a difference, as the inference in the used CNN is deterministic. This is a phenomenon similar to adversarial examples in image classification (same intension, but different extension).

We now analyze the predictions of our different models for the example song given in Figure 3.2. We compare the predictions of the following three different models: the text-based model $\text{CNN}_{\text{text}}\{\text{str}\}$ (see Figure 3.2, *repetitive lyrics structure*), the audio-based model $\text{CNN}_{\text{audio}}\{\text{chr}\}$ (see Figure 3.2, *repetitive melody structure*), and the bimodal model $\text{CNN}_{\text{mult}}\{\text{str, mfcc, chr}\}$. Starting with the first chorus, C_1, we find it to be segmented correctly by both $\text{CNN}_{\text{text}}\{\text{str}\}$ and $\text{CNN}_{\text{audio}}\{\text{chr}\}$. As previously discussed, consecutively repeated patterns are hard to segment and our text-based model indeed fails to correctly segment the repeated chorus (C_2-C_3-C_4). The audio-based model

$CNN_{audio}\{chr\}$ overcomes this limitation and segments the repeated chorus correctly. Finally, we find that in this example both the text-based and the audio-based models fail to segment the verses (the V_i) and bridges (the B_i) correctly. The $CNN_{mult}\{str, mfcc, chr\}$ model manages to detect the bridges and verses in our example.

Note that adding more modalities to a model does not always increase its ability to detect segment borders. While in some examples, the $CNN_{mult}\{str, mfcc, chr\}$ model detects segment borders that were not detected in any of the models $CNN_{text}\{str\}$ or $CNN_{audio}\{mfcc, chr\}$, there are also examples where the bimodal model does not detect a border that is detected by both the text-based and the audio-based models.

Finally, Figure 3.6 shows an example song where an octave shift in the chorus appears. The octave-shifted chorus appears as repetition in the lyrics structure, but is absent in the melody structure (green circles). Since octave-shifted notes are dissimilar under our employed audio similarity metrics, the diagonals parallel to the main diagonal, i.e. repetitions, are absent in the repetitive melody structure. Since in the lyrics the same text lines are used, the repetition appears in the repetitive lyrics structure. Note on the other hand how the overall acoustic pitch changes after 20 lines, this is visible in the melody structure, but not in the lyrics structure. In the melody structure, the upper left corner square that is clearly separated from the lower right corner square, demonstrates this. This further exemplifies how lyrics and audio complement each other.

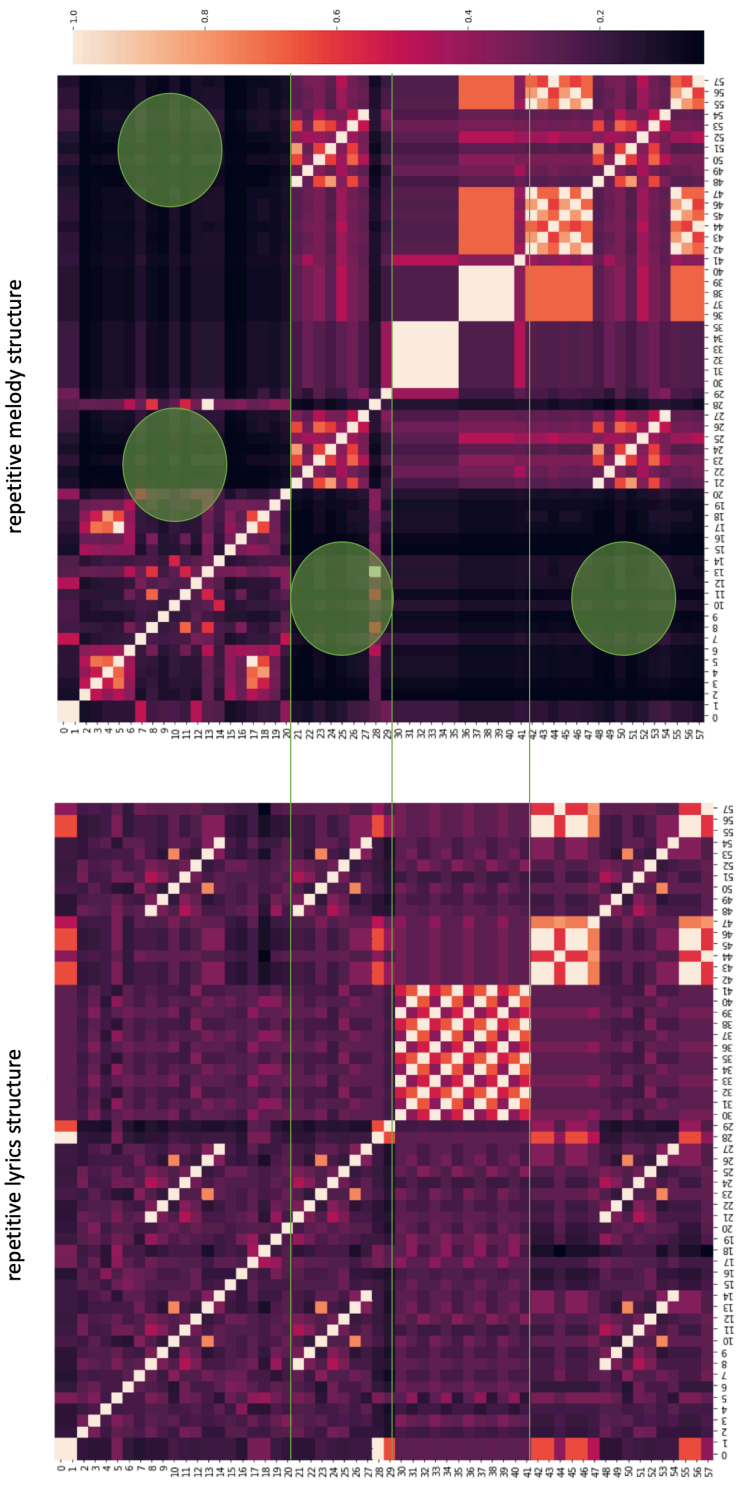


Figure 3.6: Octave-shifted chorus appears as repetition in lyrics structure, but is absent in melody structure (green circles).

3.5 RELATED WORK

Besides the work of [115] that we have discussed in detail in Section 3.2, only a few papers in the literature have focused on the automated detection of the structure of lyrics. [72] report experiments on the use of standard NLP tools for the analysis of music lyrics. Among the tasks they address, for structure extraction they focus on lyrics having a clearly recognizable structure (which is not always the case) divided into segments. Such segments are weighted following the results given by descriptors used (as full length text, relative position of a segment in the song, segment similarity), and then tagged with a label describing them (e.g. chorus, verses). They test the segmentation algorithm on a small dataset of 30 lyrics, 6 for each language (English, French, German, Spanish and Italian), which had previously been manually segmented.

More recently, [6] describe a semantics-driven approach to the automatic segmentation of song lyrics, and mainly focus on pop/rock music. Their goal is not to label a set of lines in a given way (e.g. verse, chorus), but rather identifying recurrent as well as non-recurrent groups of lines. They propose a rule-based method to estimate such structure labels of segmented lyrics, while in our approach we apply machine learning methods to unsegmented lyrics.

[28] propose a new method for enhancing the accuracy of audio segmentation. They derive the semantic structure of songs by lyrics processing to improve the structure labeling of the estimated audio segments. With the goal of identifying repeated musical parts in music audio signals to estimate music structure boundaries (lyrics are not considered), [31] propose to feed Convolutional Neural Networks with the square-sub-matrices centered on the main diagonals of several SSMs, each one representing a different audio descriptor, building their work on [49].

For a different task than ours [76], use a corpus of 100 lyrics synchronized to an audio representation with information on musical key and note progression to detect emotion. Their classification results using both modalities,

textual and audio features, are significantly improved compared to a single modality.

As an alternative to our CNN-based approach, Recurrent Neural Networks can also be applied to lyrics segmentation, for example in the form of a sequence labeller [70] or a generic text segmentation model [67].

3.6 DISCUSSION: SEGMENT LABELLING

Previously in this Chapter we have dealt with lyrics segmentation while leaving the consecutive task of **segment labelling** untouched. In this section we sketch a labelling approach based on a song text where the segmentation is already known. Figure 3.7 shows a song text and its segmentation into the segments A, B, C, D, E, as given by the annotation of the text. As a Pop song, this example has a fairly common structure which can be described as: *Verse 1-Chorus-Verse 2-Chorus-Outro*. The reasoning behind this structure analysis is that perfectly repeating parts usually correspond to the chorus. Hence, B and D should both be a chorus. A and C are verses, as they lead to the chorus and there is no visible bridge. The last segment, E, repeats the end of the chorus and is very short, so it can be classified as an outro. While the previous analysis appears plausible, it relies on world knowledge, such that the chorus is the most repeated part, a verse usually leads into a chorus (optionally via a bridge), and an outro ends a song text, but is optional[7].

One can imagine formalizing the previously used rules to build an automatic segment labeller. As a first approximation, identifying a chorus becomes tractable if we define it as the most repeated part in a song. Given the lyrics segments, we can cluster them and measure how highly they are repeated, while also allowing for partial repetitions. What this approach boils down to is finding useful

7 For more details on the set of structure types, we refer the reader to [16, 109]

A	0	I parked my car 'round back
	1	I've got the shades pulled down
	2	I told everybody including my mama
	3	I was leaving town
	4	But I've been right here
	5	Since you've been gone
	6	Belly-up at the bottom of a bottle
	7	Listening to George Jones
B	8	And just playin' possum
	9	Laying low
	10	I've got hundred watts of hurtin'
	11	Coming through the speakers of my stereo
	12	Don't want to see nobody
	13	Nowhere I want to go
	14	I'm just playin' possum
	15	And laying low
C	16	I'm gonna hide my heart
	17	And be a love recluse
	18	Oh I could cry on my best friend's shoulder
	19	But there ain't no use
	20	I need an expert on
	21	The pain I'm going through
	22	So I'll keep George on the old turntable
	23	'Til I'm over you
D	24	And just playin' possum
	25	Laying low
	26	I've got hundred watts of hurtin'
	27	Coming through the speakers of my stereo
	28	Don't want to see nobody
	29	Nowhere I want to go
	30	I'm just playin' possum
	31	And laying low
E	32	He's playin' possum
	33	And he's laying low

Figure 3.7: Segment structure of a Pop song (“Don’t Rock The Jukebox” by A. Jackson, MLDB-ID: 2954)

similarity metrics to compare the lyrics segments[8]. While chorus tend to repeat themselves close to verbatim, this does not hold for verses. While similar chorus instances are clustered well with a simple edit distance, which similarity metric will cluster the verses successfully remains an open question. Combining different similarity metrics can lead to successful clustering, for instance taking into account the audio, as we did in the lyrics segmentation approach, the verses will often cluster since they tend to share the same melody, while using different words.

3.7 CONCLUSION

In this Chapter, we have considered the problem of structure detection of song lyrics. We have broken down the task into the two subtasks lyrics segmentation and segment labelling. We then have addressed the task of lyrics segmentation on synchronized text-audio representations of songs. For the songs in the corpus DALI where the lyrics are aligned to the audio, we have derived a measure of alignment quality specific to our task of lyrics segmentation. Then, we have shown that exploiting both textual and audio-based features lead our Convolutional Neural Network-based model to significantly outperform the state-of-the-art system for lyrics segmentation that relies on purely text-based features. Moreover, we have shown that the advantage of a bimodal segment representation pertains even in the case where the alignment is noisy. This indicates that a lyrics segmentation model can be improved in most situations by enriching the segment representation by another modality (such as audio). We have briefly discussed the task of segment labelling and gave an approximation to chorus detection based on clustering the lyrics segments using different similarity metrics.

As for future work, the problem of inconsistent classification inside of a song (SSM patterns look almost identically,

8 Note that we use this approach with a simple edit distance to approximately find the chorus in our lyrics summarization approach in Chapter 4.

but classifications differ) may be tackled by clustering the SSM patterns in such a way that very similar looking SSM patterns end up in the same cluster. This can be seen as a preprocessing denoising step of the SSMs where details that are irrelevant to our task are deleted, without losing relevant information. Furthermore, the problem that the bimodal model sometimes fails to detect a segment border, even if the submodels correctly detected that border may be tackled by implementing a late fusion approach [105] where the prediction of the bimodal model is conditioned on the predictions of both the text-based and the audio-based submodels. Finally, we would like to experiment with further modalities, for instance with subtitled music videos where text, audio, and video are all synchronized to each other. For segment labelling an obvious way to go would be to manually label a dataset and then learn to predict the segment labels using supervised learning. We hypothesize that this labelling is mostly useful for lyrics with a canonical structure in genres such as Pop or Country. Instead of regressing to predefined labels, an alternative direction is to identify types of segments that are meaningful to downstream tasks such as music recommendation and then try to come up with new taxonomies to house these types of segments.

4

LYRICS CONTENT

In this Chapter we deal with the problem of representing the content of lyrics. We explain the limitations we found with representations based on topic models and information extraction. We then introduce our final content representation by means of text summarization. We propose a method to summarize the lyrics in a way that respects their intimate relation to music[1].

CONTENTS

4.1 INTRODUCTION

There are different routes to go about describing the content of a song text and which way to go depends on the goal we want to achieve with the content representation. We

1 This work has been published at RANLP 2019.

fundamentally had two applications in mind: (1) generating useful search engine snippets when someone searches for lyrics and (2) allowing a user search for general themes in lyrics.

With "theme" we mean for example (T) *Someone is swimming in the sea*. Such a theme T can be instantiated in lyrics with lines such as (t1) *I swam in the ocean* or (t2) *He saw her swimming with the dolphins*. Given a method that can identify the main themes of the lyrics and induce T from instances t1, t2, we can generate a useful content representation of the song text by means of the central themes in it. In turn, this can help solving the dual problem (2), i.e. finding examples t1, t2 of lyrics that instantiate T. To work towards achieving such an endeavor, we initially explored methods based on topic modelling and information extraction. In the following, we describe the limitations of those approaches and converge to the approach we ultimately took - based on extractive summarization.

Using standard **topic modelling** methods such as Latent Dirichlet Allocation (LDA) [14] or Non-negative Matrix Factorization (NMF) [61] we get for each song text a probability distribution over topics. In turn, topics are represented as weighted bags of words (wBOW). To associate these wBOW with semantically meaningful concepts, there are two standard approaches. First, manually labelling the wBOW and second, automatically inducing the topic labels [13]. While the manual labelling is an arguably subjective task, we found the automatic labelling approaches not suitable for our datasets. In both cases, the labels tend to be rather abstract, such as *love* or *family* or *war* - more abstract than what our requirements for the themes are. We consequently ruled out a content description of the lyrics as a mixture of topics[2].

We also experimented with an **information extraction** approach, with the goal to identify the main relations in the lyrics. For instance, *I walked up the hill* and *She climbed the mountain* were supposed to result in similar extracted

2 Note that we nevertheless implement a basic topic model in our NLP annotations for the WASABI Song Corpus, as described in Section 6.2.

relations. This then would facilitate search for lyrics where someone climbs a mountain. To achieve this technically, we extracted from each lyrics line the relations using Open Information Extraction [3]. We tried to use this approach to extract simple relations from a sentence, such as *climbs(She, the mountain)*. However, the low quality of the resulting relations prohibited the successful use of this approach. As discussed previously, standard NLP tools such as part of speech (POS) taggers are not well-suited for application to lyrics, given that lyrics lines often consist of partial sentences, also employing higher degrees of figurative language. Consequently, higher-level tasks that rely on POS taggers, such as information extraction applications, can suffer in performance.

While the topic model approach generated results that were too abstract for our application, the information extraction approach in turn gave us results that were neither abstract nor correct enough. We finally decided as an intermediate step towards finding abstract themes in lyrics to represent the main themes by generating an **extractive summary** of the lyrics. While this approach does not abstract over instances, we see this as a successful first step for extracting *correct instances* of the themes. In Section 4.6 we describe how we aim to reach more abstraction for our method.

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning of a text [1]. Numerous approaches have been developed to address this task and applied widely in various domains including news articles [29], scientific papers [73], web content as blogs [56], customer reviews [96] and social media messages [53]. But no approaches exist for summarizing song lyrics. Such summaries can, however be useful, for instance to produce adequate **snippets for a search engine** dedicated to an online song collection or for music digital libraries. From a linguistic point of view however, lyrics are a very peculiar genre of document and generic summarization methods may not be appropriate when the input for summarization

comes from a specific domain or type of genre as songs are [83]. Compared to news documents, for instance, lyrics have a very different structure. Given the repeating forms, peculiar structure (e.g. the segmentation into verse, chorus, etc.) and other unique characteristics of song lyrics, we need the summarization algorithms to take advantage of these additional elements to more accurately identify relevant information in song lyrics. But just as such characteristics enable the exploration of new approaches, other characteristics make the application of summarization algorithms very challenging, as the presence of repeated lines, the discourse structure that strongly depends on the interrelation of music and words in the melody composition, the heterogeneity of musical genres each featuring characteristic styles and wording [16], and simply the fact that not all songs tell a story.

In this direction, this Chapter focuses on the following research questions: *What is the impact of the context in summarizing song lyrics?*. This question is broken down into two sub questions: 1) *How do generic text summarization methods perform over lyrics?* and 2) *Can such peculiar context be leveraged to identify relevant sentences to improve song text summarization?*

To answer our research questions, we experiment with generic unsupervised state-of-the-art text summarization methods (i.e. TextRank, and a topic distribution based method) to perform lyrics summarization, and show that adding contextual information helps such models to produce better summaries. Specifically, we enhance text summarization approaches with a method inspired by audio thumbnailing techniques, that leverages the repetitive structure of song texts to improve summaries. We show how summaries that take into account the audio nature of the lyrics outperform the generic methods according to both an automatic evaluation over 50k lyrics, and judgments of 26 human subjects.

In the following, Section 4.2 reports on related work. Section 4.3 presents the lyrics summarization task and the proposed methods. Sections 4.4 and 4.5 report on the experiments and on the evaluation, respectively. In Section 4.6 we

discuss how to make our method more abstract. Section 4.7 concludes the Chapter.

4.2 RELATED WORK IN SUMMARIZATION

This section reports on the related work on both text and audio summarization methods.

4.2.1 *Text Summarization*

In the literature, there are two different families of approaches for automatic text summarization: extraction and abstraction [1]. **Extractive summarization** methods identify important elements of the text and generate them verbatim, i.e. they depend only on extraction of sentences or words from the original text. In contrast, **abstractive summarization** methods interpret and examine the text to generate a new shorter text that conveys the most critical information from the original text. Even though summaries created by humans are usually not extractive, most of the summarization research has focused on extractive methods. Purely extractive summaries often give better results [82], due to the fact that latter methods cope with more complex problems such as semantic representation, inference and natural language generation. Existing abstractive summarizers often rely on an extractive pre-processing component to produce the abstract of the text [10, 62]. Consequently, in this Chapter we focus on extractive summarization methods, also given the fact that lyrics *i)* strongly use figurative language which makes abstractive summarization even more challenging; and *ii)* the choice of the words by the composer may also have an importance for capturing the style of the song.

As no available gold-standard of human-produced summaries of song texts exists, we focus on **unsupervised methods** for text summarization, the ones targeted in our study. Most methods have in common the process for summary generation: given a text, the importance of each sentence of that text is determined. Then, the sentences with

highest importance are selected to form a summary. The ways different summarizers determine the importance of each sentence may differ: *Statistics-based summarizers* extract indicator features from each sentence, e.g. [42] use among others the sentence position and length and named entities as features. *Topic-based summarizers* aim to represent each sentence by its underlying topics. For instance, [55] apply Probabilistic Latent Semantic Analysis, while Latent Dirichlet Allocation is used in [4] to model each sentence's distribution over latent topics. Another type of summarization methods is *graph-based summarizers*. Three of the most popular graph-based summarization algorithms are TextRank [77], LexRank [39], and [92]. These methods work by constructing a graph whose nodes are sentences and whose graph edge weights are sentence similarities. Then, the sentences that are central to the graph are found by computing the PageRank [88]. Contrarily to all previously described methods, systems using *supervised machine learning* form another type of summarizers. For instance, [41] treats extractive summarization as a binary classification task, where they extract indicator features from sentences of gold summaries and learn to detect the sentences that should be included in a summary.

CONTEXT-SPECIFIC SUMMARIZATION. If specific knowledge about the application scenario or the domain of the summarized text is available, generic summarization methods can be adapted to take into account the prior information. In query-based summarization [87, 113], the user's query is taken into account when generating a summary. Summarization of a scientific paper can be improved by considering the citations of it, as in [34]. However, to the best of our knowledge no summarization methods have been proposed for the domain of song texts. In this paper we present a summarization method that uses prior knowledge about the text it summarizes to help generic summarizers generate better summaries.

EVALUATION CRITERIA AND METHODS. Summaries should *i)* contain the most important information from

input documents, *ii)* not contain redundant information, *iii)* be readable, hence they should be grammatical and coherent [93]. While a multitude of methods to identify important sentences has been described above, several approaches aim to make summaries less redundant and more coherent. The simplest way to evaluate summaries is to let humans assess the quality, but this is extremely expensive. The factors that humans must consider when giving scores to each candidate summary are grammaticality, non redundancy, integration of most important pieces of information, structure and coherence [102]. The more common way is to let humans generate possibly multiple summaries for a text and then automatically assess how close a machine-made summary is to the human gold summaries computing ROUGE scores [68], which boils down to measuring n-gram overlaps between gold summaries and automatic summary. More recently there have been attempts to rate summaries automatically without the need for gold summaries [83]. The key idea is that a **summary should be similar to the original text** in regard to characteristic criteria as the word distribution. [71] find that topic words are a suitable metric to automatically evaluate micro blog summaries.

4.2.2 *Audio Summarization*

Lyrics are texts that accompany music. Therefore, it is worthwhile to see if methods in audio summarization can be transferred to lyrics summarization. In audio summarization the goal is to find the most representative parts in a song, in Pop songs those are usually the chorus and the bridge, in instrumental music the main theme. The task of creating short audio summaries is also known as **audio thumbnailing** [8, 25, 66], as the goal is to produce a short representation of the music that fits onto a thumbnail, but still covers the most representative parts of it. In a recent approach of audio thumbnailing [59], the authors generate a *Double Thumbnail* from a musical piece by finding the two most representative parts in it. For this, they search for

candidate musical segments in an a priori unsegmented song. Candidate musical segments are defined as sequences of music that more or less exactly repeat themselves. The representativeness of each candidate segment to the whole piece is then estimated by their fitness metric. They define the fitness of a segment as a trade-off between how exactly a part is repeated and how much of the whole piece is covered by all repetitions of that segment. Then, the audio segments along with their fitness allow them to create an audio double thumbnail consisting of the two fittest audio segments.

4.3 OUR APPROACH TO LYRICS SUMMARIZATION

Song texts are arranged in segments and lines. For instance the song text depicted in Figure 4.1 consists of 8 segments and 38 lines. Given a song text S consisting of n lines of text, $S = (x_1, ..., x_n)$, we define the task of *extractive lyrics summarization* as the task of producing a concise summary *sum* of the song text, consisting of a subset of the original text lines: $sum(S) \subseteq S$, where usually $|sum(S)| << |S|$. We define the goal of a summary as to preserve key information and the overall meaning of a song text.

We address this task with the following unsupervised extractive summarization methods. We first apply the popular **graph-based** summarizer *TextRank*. Second, we adapt of a **topic-based** method, which we call *TopSum*. Third, we introduce a method inspired by audio thumbnailing, which we dub **Lyrics Thumbnail**. This method aims at identifying the most representative parts of the original song text and then creating a summary from them. Lastly, based on these three methods, we build model combinations. The combination process is described in Section 4.4.2. While for TextRank we rely on the off-the-shelf implementation of [7], in the following we describe the other two methods.

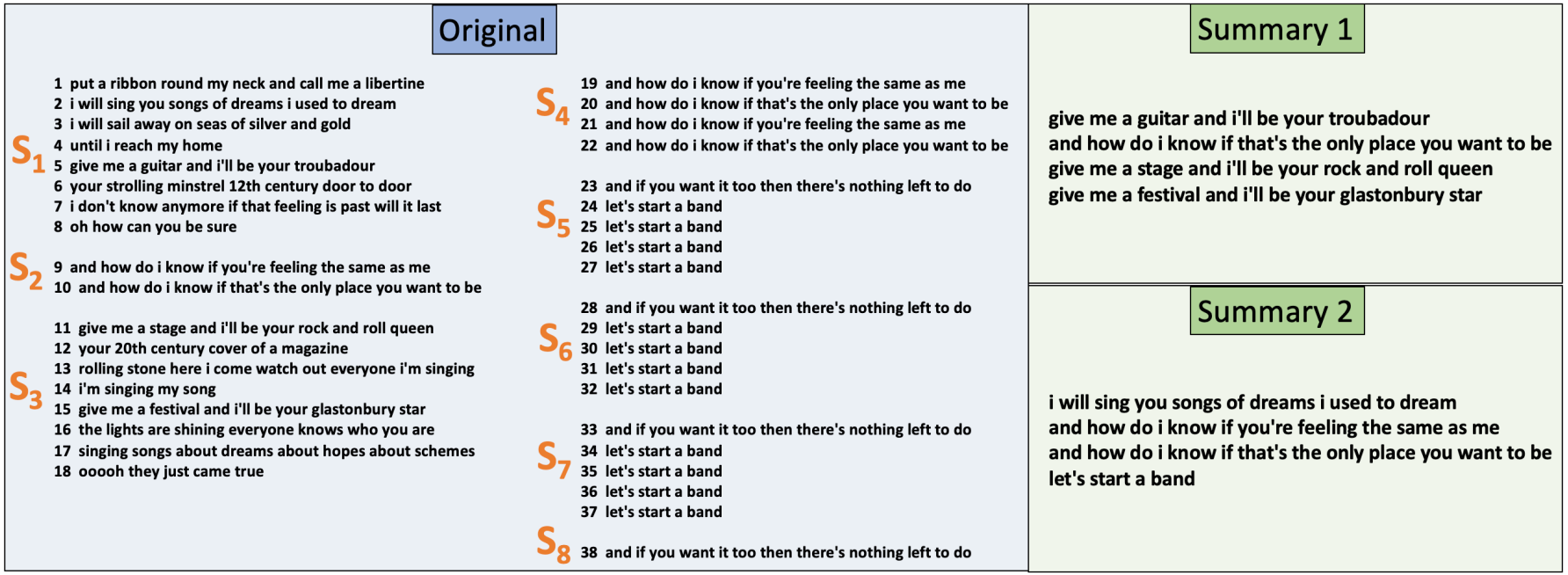


Figure 4.1: Song text of "Let's start a band" by Amy MacDonald along with two example summaries.

4.3.1 *Topic-based Summarization: TopSum*

We implement a simple topic-based summarization model, which we dub **TopSum**, that aims to construct a summary whose topic distribution is as similar as possible to that of the original text. Following [61], we train a topic model by factorizing a tf-idf-weighted term-document matrix of a song text corpus (see Section 4.4.2) using non-negative matrix factorization into a term-topic and a topic-document matrix. Given the learnt term-topic matrix, we compute a topic vector t for each new document (song text). In order to treat t as a (pseudo-) probability distribution over latent topics t_i, we normalize t by applying $\lambda t.t/\sum_{t_i \in t} t_i$ to it. Given the distributions over latent topics for each song text, we then incrementally construct a summary by greedily adding one line from the original text at a time (same mechanism as in the KLSum algorithm in [52]); that line x^* of the original text that minimizes the distance between the topic distribution t_S of the original text S and the topic distribution of the incremental summary $sum(S)$:

$$x^* = \underset{x \in (S \setminus sum(S))}{\operatorname{argmin}} \{W(t_S,\ t_{sum(S)+x})\}$$

W is the Wasserstein distance [112] and is used to measure the distance between two probability distributions (an alternative to Jensen-Shannon divergence [69]).

4.3.2 *Fitness-based Summarization: Lyrics Thumbnail*

Inspired by work in audio thumbnailing [59], we transfer their fitness measure for audio segments to compute the fitness of lyrics segments. Analog to an audio thumbnail, we define a **Lyrics Thumbnail** as the most representative and repetitive part of the song text. Consequently, it usually consists of (a part of) the chorus. In our corpus the segments are annotated (as double line breaks in the lyrics), so unlike in audio thumbnailing, we do not have to induce segments, but rather measure their fitness. In the following, we describe the fitness measure for lyrics segments and how we use this to generate a summary of the lyrics.

LYRICS FITNESS

Given a segmented song text $S = (S_1, ..., S_m)$ consisting of text segments S_i, where each S_i consists of $|S_i|$ text lines, we cluster the S_i into partitions of similar segments. For instance, the lyrics in Figure 4.1 consists of 8 segments and 38 lines, where the cluster $\{S_5, S_6, S_7\}$ are the instances of the chorus. $\{S_1, S_3\}$ are the verses and $\{S_4\}$ is the bridge leading into the chorus. The fitness Fit of the segment cluster $C \subseteq S$ is defined through the precision pr of the cluster and the coverage co of the cluster. pr describes how similar the segments in C are to each other while co is the relative amount of lyrics lines covered by C:

$$pr(C) = (\sum_{\substack{S_i, S_j \in C \\ i<j}} 1)^{-1} \cdot \sum_{\substack{S_i, S_j \in C \\ i<j}} sim(S_i, S_j)$$

$$co(C) = (\sum_{S_i \in S} |S_i|)^{-1} \cdot \sum_{S_i \in C} |S_i|$$

where sim is a normalized similarity measure between text segments. Fit is the harmonic mean between pr and co. The fitness of a segment S_i is defined as the fitness of the cluster to which S_i belongs:

$$\forall S_i \in C : Fit(S_i) = Fit(C) = 2\frac{pr(C) \cdot co(C)}{pr(C) + co(C)}$$

For lyrics segments without repetition the fitness is defined as zero. Based on the fitness Fit for segments, we define a fitness measure for a text line x. This allows us to compute the fitness of arbitrary summaries (with no or unknown segmentation). If the text line x occurs $f_i(x)$ times in text segment S_i, then its line fitness fit is defined as:

$$fit(x) = (\sum_{S_i \in S} f_i(x))^{-1} \cdot \sum_{S_i \in S} f_i(x) \cdot Fit(S_i)$$

FITNESS-BASED SUMMARY

Analog to [59]'s audio thumbnails, we create fitness-based summaries for a song text. A *Lyrics Double Thumbnail* consists of two segments: one from the fittest segment cluster (usually the chorus), and one from the second fittest segment cluster (usually the bridge)[3]. If the second fittest cluster has a zero fitness, we generate a *Lyrics Single Thumbnail* solely from the fittest cluster. If the thumbnail generated has a length of k lines and we want to produce a summary of $p < k$ lines, we select the p lines in the middle of the thumbnail following [25]'s *Section-transition Strategy* that they find to capture the *hook* of the music more likely[4].

4.4 EXPERIMENTAL SETTING

In the following we describe the experimental setup we have used to evaluate the different lyrics summarization approaches. In Section 4.4.1 we describe the dataset used. Then, in Section 4.4.2 we detail the parametrizations of the summarization models and define the method of combining different summarization methods.

4.4.1 *Dataset*

From the WASABI Song Corpus (see Section 2.3) we select a subset of 190k unique song texts with available genre information, to allow for a genre-wise evaluation. We focus on the ten most frequent genres in the corpus, as the corpus has spurious genres, in total 416 different ones. We add two additional genres from the underrepresented Rap field: Southern Hip Hop and Gangsta Rap. The dataset then contains 95k song lyrics from 12 different genres.

To allow for a fair comparison between different summaries we control for the summary length. [8] recommend to create audio thumbnails of the median length of the

3 We pick the first occurring representative of the segment cluster. Which segment to pick from the cluster is a potential question for future work.

4 They also experiment with other methods to create a thumbnail, such as section initial or section ending.

chorus on the whole corpus. We follow this and estimate the chorus of each song text by computing its *Lyrics Single Thumbnail*. We find the median chorus length to be four lines, hence we decide to generate summaries of such length for all lyrics and all summarization models to exclude the length bias in the methods comparison[5]. As the length of the lyrics thumbnail is lower-bounded by the length of the chorus in the song text, we keep only those lyrics with an estimated chorus length of at least four lines. The final corpus of 12 genres consists of 50k lyrics with the following genre distribution: Rock: 8.4k, Country: 8.3k, Alternative Rock: 6.6k, Pop: 6.9k, R&B: 5.2k, Indie Rock: 4.4k, Hip Hop: 4.2k, Hard Rock: 2.4k, Punk Rock: 2k, Folk: 1.7k, Southern Hip Hop: 281, Gangsta Rap: 185.

4.4.2 *Models and Configurations*

We create summaries using the three summarization methods described in Section 4.3. The TextRank method is based on **graph centrality**; it creates summaries that contain the sentences that are most central to the text. The TopSum method creates summaries that contain sentences that capture the topics of the text; based on an analysis of the **topic distribution**. Finally, the Lyrics Thumbnail contains the lines that are most repeated and representative for the lyrics; based on the **fitness metric** of the lyrics. We hypothesize that these methods are somewhat complementary and therefore we also experiment with model combinations (described below). While the Lyrics Thumbnail is generated from the full segment structure of the lyrics including its duplicate lines, all other models are fed with unique text lines as input (i.e. rendundant lines are deleted). This is done to produce less redundant summaries, given that for instance, TextRank scores each duplicate line the same, hence it is prone to create summaries with all identical lines. TopSum can suffer from a similar shortcoming: if there is a duplicate line close to the ideal topic distribution, adding that line

5 We leave the study of other measures to estimate the summary length to future work.

again will let the incremental summary under construction stay close to the ideal topic distribution. As previously explained, all models were instructed to produce summaries of four lines. The summary lines were arranged in the same order they appear in the original text[6]. We use the TextRank implementation[7] of [7] without removing stop words. Since the lyrics lines in the input can be quite short, we avoid losing all content of the line if removing stop words. The topic model for TopSum is built using non-negative matrix factorization with scikit-learn[8] [97] for 30 topics on the full corpus of 190k lyrics[9]. For the topical distance, we only consider the distance between the three most relevant topics in the original text, following the intuition that one song text usually covers only a small amount of topics. The Lyrics Thumbnail is computed using String-based distance between text segments to facilitate clustering. This similarity has been shown in [115] to indicate segment borders successfully. In our implementation, segments are clustered using the DBSCAN [40] algorithm[10].

MODEL COMBINATION

To test the hypothesis if summaries can benefit from the complementary perspectives the three different summarization methods take, we experiment with model combinations. Table 4.1 defines the models we use in the experiments along with the model names we will use henceforth to refer to them.

In the following we detail the method we have used to combine different summarizers. For any lyrics line, we can obtain a score from each of the applied models. $\mathbb{R}ank$ provides a score for each line, $\mathbb{T}opic$ provides a distance between the topic distributions of an incremental summary and the original text, and $\mathbb{F}it$ provides the fitness of each line. We treat our summarization methods as blackboxes

6 In case of repeated parts, the first position of each line was used as original position.
7 https://github.com/summanlp/textrank
8 https://scikit-learn.org
9 loss='kullback-leibler'
10 eps=0.3, min_samples=2

Model	*Summarization Methods*
$\mathbb{R}ank$	TextRank
$\mathbb{T}opic$	TopSum
$\mathbb{F}it$	Lyrics Thumbnail (LT)
$\mathbb{R}ank\mathbb{T}opic$	TextRank, TopSum
$\mathbb{R}ank\mathbb{T}opic\mathbb{F}it$	TextRank, TopSum, LT

Table 4.1: The models used in our experiment and the summarization methods they use.

and use a simple method to combine the scores the different methods provide for each line. Given the original text separated into lines $S = (x_1, ..., x_n)$, a summary is constructed by greedily adding one line x^* at a time to the incremental summary $sum(S) \subseteq S$ such that the sum of normalized ranks of all scores is minimal:

$$x^* = \text{argmin} \bigcup_x \{\sum_A \mathrm{R_A}(x)\}$$

Here $x \in (S \setminus sum(S))$ and $A \in \{\mathbb{R}ank, \mathbb{T}opic, \mathbb{F}it\}$. The normalized rank $\mathrm{R_A}(x)$ of the score that method A assigns to line x is computed as follows: first, the highest scores[11] are assigned rank 0, the second highest scores get rank 1, and so forth. Then the ranks are linearly scaled to the [0,1] interval, so each sum of ranks $\sum_A \mathrm{R_A}(x)$ is in [0,3].

4.5 EVALUATION

We evaluate the quality of the produced lyrics summary both soliciting human judgments on the goodness and utility of a given summary (Section 4.5.1), and through an automatic evaluation of the summarization methods (Section 4.5.2) to provide a comprehensive evaluation.

11 In the case of topical distance, a "higher score" means a lower value.

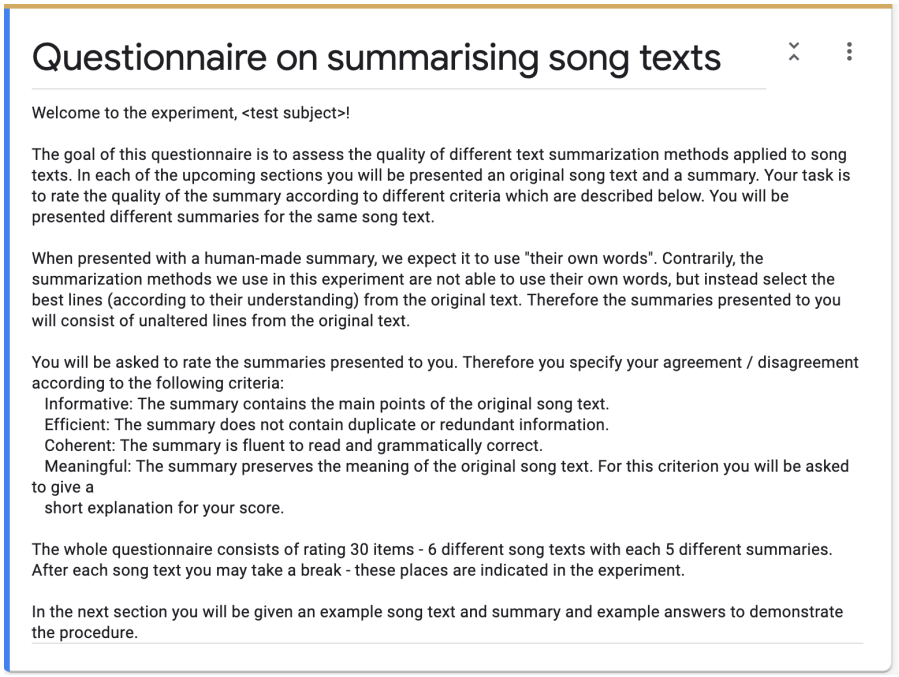
4.5.1 *Human Evaluation*

We performed human evaluation of the different summarization methods introduced before by asking participants to rate the different summaries presented to them by specifying their agreement / disagreement according to the following standard criteria [93] plus one additional criterion coming from our definition of the lyrics summarization task:

- **Informativeness**: The summary contains the main points of the original song text.
- **Non-redundancy**: The summary does not contain duplicate or redundant information.
- **Coherence**: The summary is fluent to read and grammatically correct.
- **Meaning**: The summary preserves the meaning of the original song text.

An experimental psychologist expert in Human Computer Interaction advised us in defining the questionnaire and setting up the experiment. 26 participants - 12 nationalities, 18 men, 8 women, aged from 21 to 59 - were taking a questionnaire (Google Forms), consisting of rating 30 items with respect to the criteria defined before on a Likert scale from 1 (low) to 5 (high). Each participant was presented with 5 different summaries - each produced by one of the previously described summarization models - for 6 different song texts. Participants first read the experimental instruction (see Figure 4.2) to familiarize themselves with the task at hand.

The experimental subjects were given example ratings for the different criteria in order to familiarize them with the procedure. Then, for each song text, the original song text along with its 5 summaries were presented in random order and had to be rated according to the above criteria. For the criterion of Meaning, we asked participants to give a short explanation in free text for their score. The selected 6 song


Questionnaire on summarising song texts

Welcome to the experiment, <test subject>!

The goal of this questionnaire is to assess the quality of different text summarization methods applied to song texts. In each of the upcoming sections you will be presented an original song text and a summary. Your task is to rate the quality of the summary according to different criteria which are described below. You will be presented different summaries for the same song text.

When presented with a human-made summary, we expect it to use "their own words". Contrarily, the summarization methods we use in this experiment are not able to use their own words, but instead select the best lines (according to their understanding) from the original text. Therefore the summaries presented to you will consist of unaltered lines from the original text.

You will be asked to rate the summaries presented to you. Therefore you specify your agreement / disagreement according to the following criteria:
Informative: The summary contains the main points of the original song text.
Efficient: The summary does not contain duplicate or redundant information.
Coherent: The summary is fluent to read and grammatically correct.
Meaningful: The summary preserves the meaning of the original song text. For this criterion you will be asked to give a
short explanation for your score.

The whole questionnaire consists of rating 30 items - 6 different song texts with each 5 different summaries. After each song text you may take a break - these places are indicated in the experiment.

In the next section you will be given an example song text and summary and example answers to demonstrate the procedure.

Figure 4.2: The experimental instruction we provided to the participants to explain the experimental paradigm.

texts[12] have a minimum and a median chorus length of 4 lines and are from different genres, i.e. Pop/Rock (4), Folk (1) and Rap (1), similar to our corpus genre distribution. Song texts were selected from different lengths (18-63 lines), genders of singer (3 male, 3 female), topics (family, life, drugs, relationship, depression), and mood (depressive, angry, hopeful, optimistic, energetic). The artist name and song title were not shown to the participants.

RESULTS

Figure 4.3 shows the ratings obtained for each criterion. We examine the significant differences between the models performances by performing a paired two-tailed t-test. The significance levels are: $0.05^{*}, 0.01^{**}, 0.001^{***}$, and *n.s.* First, Informativeness and Meaning are rated higher** for the combined model $\mathbb{R}ank\mathbb{T}opic$ compared to the single models $\mathbb{R}ank$ and $\mathbb{T}opic$. Combining all three models im-

12 *Pills N Potions* by Nicki Minaj, *Hurt* by Nine Inch Nails, *Real to me* by Brian McFadden, *Somebody That I Used To Know* by Gotye, *Receive* by Alanis Morissette, *Let's Start A Band* by Amy MacDonald

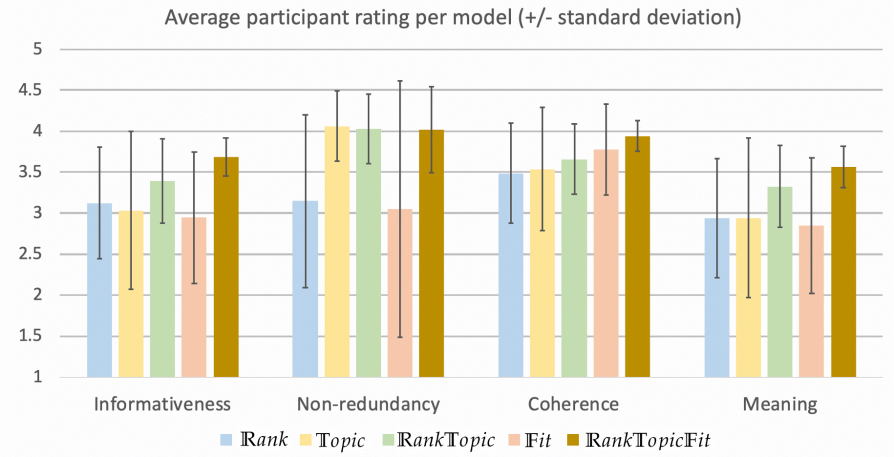


Figure 4.3: Human ratings per summarization model in terms of average and standard deviation.

proves the summaries further: both for Informativeness and Meaning the model ℝ*ank*𝕋*opic*𝔽*it* is rated higher*** than ℝ*ank*𝕋*opic*. Further, summaries created by ℝ*ank*𝕋*opic*𝔽*it* are rated higher*** in Coherence than summaries from any other model - except from 𝔽*it* (*n.s.* difference). Summaries are rated on the same level (*n.s.* differences) for Non-redundancy in all but the ℝ*ank* and 𝔽*it* summaries, which are perceived as lower*** in Non-redundancy than all others. Note, how the model ℝ*ank*𝕋*opic*𝔽*it* is more stable than all others by exhibiting lower standard deviations in all criteria except Non-redundancy. The criteria Informativeness and Meaning are highly correlated (Pearson correlation coefficient 0.84). Correlations between other criteria range between 0.29 and 0.51.

Overall, leveraging the Lyrics Fitness in a song text summary improves summary quality. Especially with respect to the criteria that, we believe, indicate the summary quality the most - Informativeness and Meaning - the ℝ*ank*𝕋*opic*𝔽*it* method is significantly better performing and more consistent.

Figure 4.1 shows an example song text and example summaries from the experiment. Summary 1 is generated by 𝔽*it* and consists of the chorus. Summary 2 is made by the method ℝ*ank*𝕋*opic*𝔽*it* and has relevant parts of the verses and the chorus, and was rated much higher in Informa-

Evaluation criterion	Genre	$\mathbb{R}ank$	$\mathbb{T}opic$	$\mathbb{R}ank\mathbb{T}opic$	$\mathbb{F}it$	$\mathbb{R}ank\mathbb{T}opic\mathbb{F}it$	original text
Distributional Semantics [%]	Rock / Pop	92	100	97	90	93	n/a
	Rap	94	100	99	86	92	
	$\sum$	92	100	98	90	93	
Topical [%]	Rock / Pop	44	100	76	41	64	n/a
	Rap	58	100	80	48	66	
	$\sum$	46	**100**	77	42	**64**	
Coherence [%]	Rock / Pop	110	95	99	99	100	100
	Rap	112	115	112	107	107	
	$\sum$	110	97	101	100	101	
Lyrics fitness [%]	Rock / Pop	71	53	63	201	183	100
	Rap	0	0	0	309	249	
	$\sum$	62	47	55	**214**	**191**	

Table 4.2: Automatic evaluation results for the 5 summarization models and 2 genre clusters. Distributional Semantics and Topical are relative to the best model (=100%), Coherence and Fitness to the original text (=100%).

tiveness and Meaning. We analyzed the free text written by the participants to comment on the Meaning criterion, but no relevant additional information was provided; the participants mainly summarized their ratings.

4.5.2 *Automatic Evaluation*

We computed four different indicators of summary quality on the dataset of 50k songs that we have described previously (see Section 4.4.1). Three of the criteria use the similarity between probability distributions P, Q, i.e. we compute the Wasserstein distance between P and Q (cf. Section 4.3.1) and apply $\lambda x.\ x^{-1}$ to it[13]. Our criteria for the automatic evaluation of summary quality are the following:

- **Distributional Semantics**: similarity between the word distributions of original and summary, cf. [69]. We give results relative to the similarity of the best performing model (=100%).
- **Topical**: similarity between the topic distributions of original and summary. Restricted to the 3 most relevant topics of the original song text. We give results

13 This works as we always deal with distances > 0.

relative to the similarity of the best performing model (=100%).

- **Coherence**: average similarity between word distributions in consecutive sentences of the summary, cf. [104]. We give results relative to the coherence of the original song text (=100%).
- **Lyrics fitness**: average line-based fitness fit (cf. Section 4.3) of the lines in the summary. We give results relative to the Lyrics fitness of the original song text (=100%).

RESULTS

When evaluating each of the 12 genres, we found two clusters of genres to behave very similarly. Therefore, we report the results for these two groups: the *Rap* genre cluster contains Hip Hop, Southern Hip Hop, and Gangsta Rap. The *Rock / Pop* cluster contains the 9 other genres. Results of the different automatic evaluation metrics are shown in Table 4.2. Distributional Semantics metrics have previously been shown [69, 104] to highly correlate with user responsiveness judgments. We would expect correlations of this metric with Informativeness or Meaning criteria therefore, as those criteria are closest to responsiveness, but we have found no large differences between the different models for this criterion. The summaries of the $\mathbb{T}opic$ model have the highest similarity to the original text and the $\mathbb{F}it$ have the lowest similarity of 90%. The difference between the highest and lowest values are low.

For the Topical similarity, the results are mostly in the same order as the Distributional Semantics ones, but with much larger differences. While the $\mathbb{T}opic$ model reaches the highest similarity, this is a self-fulfilling prophecy, as summaries of $\mathbb{T}opic$ were generated with the objective of maximizing topical similarity. The other two models that incorporate $\mathbb{T}opic$ ($\mathbb{R}ank\mathbb{T}opic$ and $\mathbb{R}ank\mathbb{T}opic\mathbb{F}it$), show a much higher topical similarity to the original text than $\mathbb{R}ank$ and $\mathbb{F}it$.

Coherence is rated best in $\mathbb{R}ank$ with 110%. All other models show a coherence close to that of the original text

- between 97% and 101%. We believe that the increased coherence of $\mathbb{R}ank$ is not linguistically founded, but merely algorithmic. $\mathbb{R}ank$ produces summaries of the most central sentences in a text. The centrality is using the concept of sentence similarity. Therefore, $\mathbb{R}ank$ implicitly optimizes for the automatic evaluation metric of coherence, based on similar consecutive sentences. Sentence similarity seems to be insufficient to predict human judgments of coherence in this case.

As might be expected, methods explicitly incorporating the Lyrics fitness produce summaries with a fitness much higher than the original text - 214% for the $\mathbb{F}it$ and 191% for the $\mathbb{R}ank\mathbb{T}opic\mathbb{F}it$ model. The methods not incorporating fitness produce summaries with much lower fitness than the original - $\mathbb{R}ank$ 62%, $\mathbb{T}opic$ 47%, and $\mathbb{R}ank\mathbb{T}opic$ 55%. In the Rap genre this fitness is even zero, i.e. summaries (in median) contain no part of the chorus.

Overall, no single automatic evaluation criterion was able to explain the judgments of our human participants. However, considering Topical similarity and Lyrics fitness together gives us a hint. The model $\mathbb{F}it$ has high fitness (214%), but low Topical similarity (42%). The $\mathbb{T}opic$ model has the highest Topical similarity (100%), but low fitness (47%). $\mathbb{R}ank\mathbb{T}opic\mathbb{F}it$ might be preferred by humans as it strikes a balance between Topical similarity (64%) and fitness (191%). Hence, $\mathbb{R}ank\mathbb{T}opic\mathbb{F}it$ succeeds in capturing lines from the most relevant parts of the lyrics, such as the chorus, while jointly representing the important topics of the song text.

Our experimental participants regularly commented that the task at hand was quite hard, leading us to believe that more context - especially the accompanying music - is required to better assess the quality of the presented lyrics summaries.

4.6 DISCUSSION: ABSTRACT THEMES

As we explained in the introduction, our ultimate goal of content description is to find the important general themes

from the lyrics. While our previously presented method achieves to derive important sentences of the song text, an interesting approach to experiment can be to first summarize extractively, then abstract over the space of extracted lyrics lines. A straightforward way to achieve abstraction is with sentence embeddings [35], i.e. embed the lyrics lines into a vector space, such that *I swam in the ocean* and *He saw her swimming with the dolphins* have similar vectors. Then, in a further step, we can measure the level of abstraction of each sentence with the help of ontologies. For example, *I* and *She* can be abstracted to *Someone*. Depending on the requirement of the specific application, we then can instantiate the theme with a useful level of abstraction.

4.7 CONCLUSION

In this Chapter we have discussed content descriptions of lyrics by different means, such as topic models, information extraction and ultimately developed an extractive summarization method tailored to song lyrics. We have defined and addressed the task of lyrics summarization. We have applied both generic unsupervised text summarization methods (TextRank and a topic-based method we called TopSum), and a method inspired by audio thumbnailing on 50k lyrics from the WASABI corpus. We have carried out an automatic evaluation on the produced summaries computing standard metrics in text summarization, and a human evaluation with 26 participants, showing that using a fitness measure transferred from the musicology literature, we can amend generic text summarization algorithms and produce better summaries.

In future work, we will model the importance of a line given the segment to avoid cutting off important parts of the chorus, as we sometimes observed. Moreover, we plan to address the challenging task of abstractive summarization over song lyrics, with the goal of creating a summary of song texts in prose-style - more similar to what humans would do, using their own words. For the more general task

of finding abstract themes in lyrics, we have sketched in the previous discussion (see Section 4.6) a route to progress.

5

LYRICS PERCEPTION

In this Chapter, we deal with the problem of how lyrics are perceived in the world. As an instantiation, we discuss the problem of detecting explicit content in a song text[1]. This task proves to be very hard and we show that the difficulty partially arises from the subjective nature of perceiving lyrics in one way or another depending on the context. Furthermore, we glance at the problem of how emotions are perceived in lyrics: we present our preliminary results on Emotion Recognition.

CONTENTS

1 This work has been published at RANLP 2019.

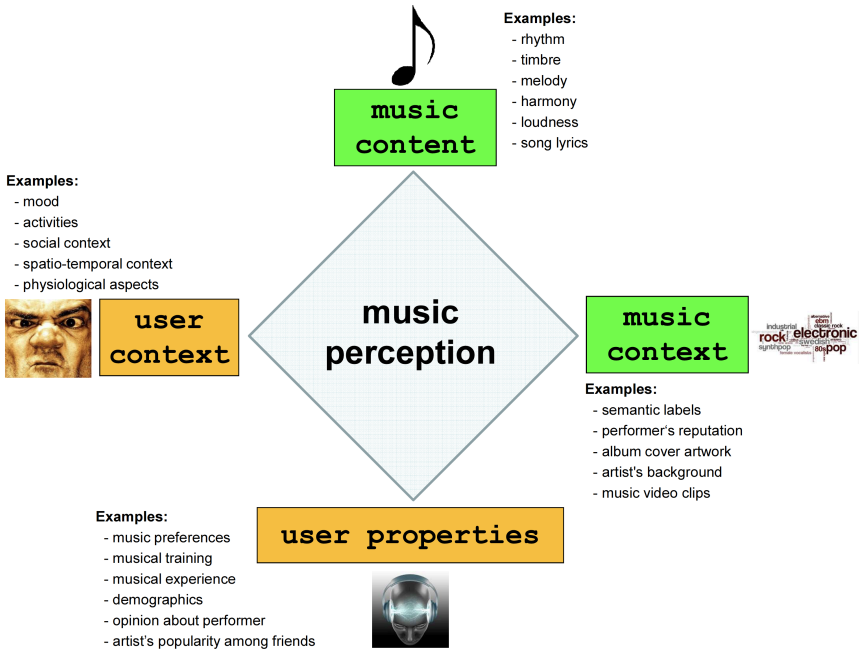


Figure 5.1: Dimensions of musical perception. Illustration taken from [103].

5.1 INTRODUCTION

As we explained at the beginning of this Thesis (Section 1.1), the different dimensions of perception of music can be described as music content, music context and listener-related factors. In this model of music perception, as illustrated in Figure 5.1, the listener-related factors are called user properties and user context. As an example, the **social context** as part of the user context, and the **musical preferences** as part of the user properties, both influence how we feel when we listen to music with a lot of swear words. While a more sensitive person who favors Country music may find this language use offensive, someone who favors Rap music and is not bothered by the use of strong language, may find it even funny.

While in the previous Chapters we have mostly focused on the music content and context dimensions, in this Chapter we face problems which to a larger degree involve the listener's perspective and thus ultimately a more subjective judgement. As an instantiation, we discuss the problem of

detecting explicit content in a song text. This task proves to be very hard and we show that the difficulty partially arises from the subjective nature of perceiving lyrics in one way or another depending on the context. Furthermore, we glance at the problem of how emotions are perceived in lyrics by presenting our preliminary results on Emotion Recognition.

Some content is inappropriate for some ages and music is no exception. Content industries have been actively searching for means to help adults determine what is and is not appropriate for children. In the USA, in 1985, the Recording Industry Association of America (RIAA) introduced the Parental Advisory Label (PAL) in order to alert parents of content unsuitable for children because of profanity or inappropriate references[2]. PAL is "a notice to consumers that recordings identified by this mark may contain strong language or depictions of violence, sex or substance abuse"[3] and that parental discretion is advised. In the UK, the British Phonographic Industry (BPI) adds to this list "racist, homophobic, misogynistic or other discriminatory language or behavior; or dangerous or criminal behavior"[4].

In the case of a song, the explicit logo is applied when the lyrics or content of a song matches one of these criteria, raising the problem of detecting and labelling explicit songs in a scalable way.

Within the Natural Language Processing community, there have been several efforts to deal with the problem of online abusive language detection, since the computational analysis of language can be used to quickly identify offenses and ease the removal of abusive messages. Several workshops [48, 90] and evaluation campaigns [15, 46, 116] have been recently organized to discuss existing approaches

2 Parental Advisory https://en.wikipedia.org/wiki/Parental_Advisory

3 RIAA PAL https://www.riaa.com/resources-learning/pal-standards/

4 BPI Parent Advisory https://www.bpi.co.uk/media/1047/parental-advisory-guidelines.pdf

to abusive language detection, propose shared tasks and foster the development of benchmarks for system evaluation. These have led to the creation of a number of datasets for abusive language detection in different languages, that have been shared within the NLP research community. The SemEval 2019 tasks HatEval [9] and OffensEval [119] have aimed at the multilingual detection of hate speech against women or immigrants and the categorization of hate speech, respectively.

In this direction, and given the similarity with the abusive language detection task, this Chapter addresses the problem of explicit content detection in song lyrics as a binary classification task: a song can be labelled either as explicit or clean (=not explicit). To this end, we deal with the following research question: *given the lyrics of a song, can we learn to detect if that text contains explicit content?* This question is broken down into the sub questions: *1) how effective are different machine learning methods in learning to detect explicit content in lyrics?* and *2) What qualitative characteristics contribute to the task's inherent difficulty and subjectivity?*

To address our research questions, we compare automated methods ranging from dictionary-based lookup to state-of-the-art deep neural networks to automatically detect explicit contents in English lyrics. We show that more complex models perform only slightly better on this task, and relying on a qualitative analysis of the data, we discuss the inherent difficulty and subjectivity of the task.

The Chapter is organized as follows: in Section 5.2 we survey the state of the art in explicit lyrics detection. In Sections 5.3 we introduce the classification methods we have applied and experiment them in Section 5.4. We discuss an alternate problem in Lyrics Perception, i.e. Lyrics-based Emotion Recognition, in Section 5.5. Finally, Conclusions end the Chapter (see Section 5.6).

5.2 RELATED WORK IN EXPLICIT LYRICS DETECTION

Only a few works on the problem of explicit lyrics detection exist. [11] consider a dataset of English lyrics (see Table 5.1,

B18) to which they apply classical machine learning algorithms such as Support Vector Machine and Random Forest. As features they extract either (i) tf-idf weighted bag-of-word (BOW) representations of each song text or (ii) represent the lyrics with paragraph vectors [63]. The explicit labels are obtained from Soundtrack Your Brand[5]. They find the Random Forest with tf-idf BOW to perform best, especially in combination with a random undersampling strategy to the highly imbalanced dataset. They also experiment with adding lyrics metadata to the feature set, such as the artist name, the release year, the music energy level, and the valence/positiveness of a song. This results in marginal improvements for some of their models.

[30] apply explicit lyrics detection to Korean song texts. They also use tf-idf weighted BOW as lyrics representation and aggregate multiple decision trees via boosting and bagging to classify the lyrics for explicit content. On their corpus (see Figure 5.1, C18) they report 78% F_1 using the bagging method. Note, that bagging with decision trees is similar to the Random Forest method used by [11]. Interestingly, they also report a baseline for dictionary lookup, i.e. given a profanity dictionary the song text is classified as explicit if and only if one of its words occurs in the profanity dictionary. With such a baseline they obtain 61% F_1.

More recently, [60] proposed a method to create explicit words dictionaries automatically by weighting a vocabulary according to the word frequencies in the explicit class vs. the clean class, accordingly. For instance the word "fuck" is typical for explicit lyrics and atypical for clean lyrics. They compare different methods to generate such a lexicon. The achieved performances using solely dictionary lookup range from 49% F_1 for a man-made dictionary to 75.6% F_1 when using relative class frequencies. Note, that the latter performance is achieved with a dictionary of only 25 words. They work with a corpus of Korean lyrics (see Figure 5.1, K19). Unlike previous work, they apply a recurrent neural network (RNN) to the task, resulting in 76.6% F_1, slightly

5 https://www.soundtrackyourbrand.com

higher than the simple dictionary lookup. They find performance to increase to 78.1% when combining the vector representation of the RNN with a one-hot vector indicating for each profane word from the dictionary if the lyric contains it. They argue to use the RNN to find such cases where the expliciteness arises from the context and not from a dictionary check. However, no examples of finding this phenomenon are presented.

5.3 DETECTION METHODS

We compare a range of classification methods for the task of explicit lyrics detection. Common to all methods is that they classify a full song into one of two mutually exclusive classes - explicit or clean (=not explicit). This means, the decision if a song text is explicit is taken globally, rendering our task as text classification. We assess the performance of different classification methods ranging from simple dictionary lookup / lexicon checking to general purpose deep learning language understanding models. We try to identify contextual effects by applying a method that outputs the *importance* for each word (see Section 5.3.4).

5.3.1 *Dictionary-Based Methods*

The most straightforward way to implement an automated explicit content detection method, is checking against a dictionary of explicit words. The dictionary can be man-made or automatically created from example explicit and clean lyrics. Then, a classifier uses this dictionary to predict the class of an unseen song text.

DICTIONARY CREATION

It is possible to use handcrafted dictionaries such as No-Swearing[6]. However, performance using an automatically created lexicon has previously been shown [60] to improve over the manually created dictionary. We therefore consider only the case of the machine-made dictionary in this work.

6 https://www.noswearing.com/

We generate a dictionary of words that are indicative of explicit lyrics. We define the importance I of a word w for explicit lyrics by the frequency $f(w, ex)$ of w in explicit lyrics compared to its frequency $f(w, cl)$ in clean lyrics:

$$I(w) = f(w, ex) / f(w, cl)$$

We filter out unique and too common words and restrict the number of terms to 1,000 to avoid overreliance on terms that are very corpus specific. The dictionary D_n of the n words most important for explicit lyrics, is now straightforwardly defined as containing the n words with the highest I score.

DICTIONARY LOOKUP

Given a dictionary D_n, this method simply checks if a song text S contains any of the explicit terms defined in D_n. Then, S is classified as explicit iff it contains at least one explicit term from D_n.

DICTIONARY REGRESSION

This method uses BOW made from D_n as the feature set of a classifier. We used a logistic regression as classifier, but Random Forest and Support Vector Machine have been used alike in [11].

5.3.2 *Tf-idf BOW Regression*

Similar to the Dictionary Regression, but the BOW contains the whole vocabulary of a training sample instead of only the explicit terms. The word features are weighted with the well-known tf-idf weighting scheme.

5.3.3 *Transformer Language Model*

Recently, approaches based on self-attention [111] have been proposed and have proven effective for natural language understanding tasks. These models are structured

as an encoder-decoder, and they are trained on unsupervised tasks (such as masked language modelling) in order to learn dense representations of sentences or documents. These models differ from more traditional recurrent neural networks in different aspects. In particular, while recurrent models can process sequences (in NLP, typically word embeddings) in order, transformers use a joint model of the right and left context of each word in order to encode an entire sequence or document. Additionally, transformers are typically less computationally expensive than recurrent models, especially when trained on a GPU accelerator.

One of the most successful transformer-based models proposed in the last few years is BERT [36]. This model is composed of multiple transformers connected by residual connections. Pre-trained models are provided by the authors, and they are used in our work to perform explicit language detection in lyrics, without re-training the full model.

5.3.4 *Textual Deconvolution Saliency*

We use the Textual Deconvolution Saliency (TDS) model of [110], which is a Convolutional Neural Network (CNN) for text classification. It is a simple model containing an embedding layer for word representations, a convolutional layer with max pooling and two fully connected layers. The interesting part about this model is that they manage to reverse the convolution. Given the learned feature map (the output of the convolution before max pooling) of the CNN, they upsample it to obtain a 3-dimensional sample with dimensions (#words, embedding size, #filters). The TDS for each word is now defined as the sum along the embedding axes of the output of the deconvolution. The TDS represents the importance of each word of the input with respect to the learned feature maps. We use this model with the goal to find local explanations for the global decision of the classification as explicit or clean. Such explanations can arise from contexts or phrases that the model assigns a high importance.

5.4 EXPERIMENTAL SETTING AND EVALUATION

We compare the different methods as introduced in the previous section in the task of explicit lyrics detection. We attempt a comparison to the related work as well, although due to different datasets comparing the reported scores directly is problematic. We finally analyze the classification qualitatively with examples, and demonstrate the intrinsic difficulty and subjectivity of the explicit lyrics detection task.
Abbreviations used: to refer to related works in Table 5.1 and 5.3, we use the following abbreviations. B18 stands for [11], C18 is [30], K19 means [60], while Ours is this work.

5.4.1 *Dataset*

The WASABI Song Corpus (see Section 2.3) contains song-wise labels for explicit lyrics, such as *explicit*, *unknown*, *no advice available*, or *clean* (=not explicit). These labels are provided by the music streaming service Deezer[7]. We selected a subset of English song texts from the corpus which are tagged as either explicit or clean. We filtered out duplicate lyrics and such that contain less than 10 tokens. Finally, our experimental dataset (henceforth called WAS) comprises of 179k lyrics, with a ratio of explicit lyrics of 9.9%. The details and comparison with related work datasets are depicted in Table 5.1.

For training any of the models described in the previous Section, we once randomly split the data into training-development-test sets with the common 60%-20%-20% ratio. We tuned the hyperparameters of the different classification algorithms on the development set to then test with the best performing parameters on the test set. As evaluation metrics we use precision (P), recall (R), and f-score (F_1). Unless stated otherwise, the scores are macro-averaged over the two possible classes.

7 https://www.deezer.com

Work	*total*	*explicit*	*ratio*	*language*
B18	25,441	3,310	13.0%	English
C18	27,695	1,024	3.7%	Korean
K19	70,077	7,468	10.7%	Korean
Ours	179,391	17,808	9.9%	English

Table 5.1: Comparison of our dataset (# songs) to the related works datasets.

5.4.2 *Hyperparameters*

For the dictionary-based methods, we found the ideal dictionary size to be 32 words for the lookup and 128 words for the regression. The Tf-idf BOW regression performed best when the full vocabulary of unigrams and bigrams was used. We used the sklearn implementation of logistic regression with the class weighting scheme *balanced* to account for the class imbalance in the dataset. We used TDS with max sequence length 512 and dropout probability 50%. As is the default with TDS, corpus-specific word vectors were trained using Word2Vec [78] with dimensionality 128. The BERT model comes pre-trained and no further pre-training was performed. We used the smaller of the two published models (bert-base-uncased). BERT then was fine-tuned to our task using max sequence length 256 and batch size 16, otherwise default parameters for text classification task learning. We used the PyTorch implementation[8] of HuggingFace [117].

5.4.3 *Results*

Overall, the results of the different classification methods we tried are all close to each other. The simple dictionary lookup with 32 words performs comparably to the deep neural network with 110M parameters (bert-base-uncased). As baseline, we include the majority class classifier that always predicts the clean class. Furthermore, all related

8 https://github.com/huggingface/transformers

works show similar tendencies of performance on their respective datasets. The results of all the different methods we applied are depicted in Table 5.2 and described in the following.

The majority class classifier delivers a performance of 47.4% F_1, which is the only outlier in the sense that this is far below any other model. The dictionary lookup with a vocabulary of the 32 most indicative explicit words obtains a balanced performance as precision and recall are close to each other, the overall performance is 77.3% F_1. The dictionary regression performs somewhat better in terms of f-score (78.5% F_1), achieving this with the highest overall recall of 81.5%, but it has lower precision. The tf-idf BOW regression performs very similarly to the dictionary regression. This proves that a limited number of words influences the overall performance of the models, and that they do not need to consider the whole vocabulary, just the most offensive words. The increased vocabulary of 929k unigrams and bigrams is gigantic compared to the explicit words dictionary (32 words). As most of these n-grams may be noise to the classifier, this could explain the slight decrease in performance over the dictionary regression. Finally, the neural-network-based methods behave a bit differently: the BERT language model is clearly better in precision (84.4%) over all other models - the second best is TDS with 81.2%. However, BERT performs the worst in recall with only 73.7%. The overall performance of BERT is average with 77.7% F_1. Finally, TDS performs best in terms of 79.6% F_1. We tested if TDS outperforming BERT was due to TDS using domain-specific word vectors trained on our corpus (BERT is trained on books and Wikipedia). This was not the case as TDS performed almost identically, when using generic word vectors (GloVe [98], 200d): 80.4% P, 78.7% R, 79.5% F_1.

A closer look at the classification performance shows that the F_1 scores for the minority class (explicit lyrics) is highest with TDS (63%) and lowest with the dictionary lookup (58.9%). The majority class (clean lyrics) on the other hand is best detected by BERT (96.3% F_1) and worst with the tf-idf BOW (95.1% F_1).

Model	*P*	*R*	F_1
Majority Class	45.0	50.0	47.4
Dictionary Lookup	78.3	76.4	77.3
Dictionary Regression	76.2	81.5	78.5
Tf-idf BOW Regression	75.6	81.2	78.0
TDS Deconvolution	81.2	78.2	79.6
BERT Language Model	84.4	73.7	77.7

Table 5.2: Performance comparison of our different models. Precision (*P*), Recall (*R*) and f-score (F_1) in %.

Work	*Model*	F_1
Ours	Dictionary Lookup	77.3
Ours	Dictionary Regression	78.5
C18	Man-made Dictionary	61.0
K19	Man-made Dictionary	49.0
K19	Dictionary Lookup	75.6
Ours	Tf-idf BOW Regression	78.0
C18	Tf-idf BOW	78.0
C18	Tf-idf BOW+	80.0
B18	Tf-idf BOW	67.5
B18	Tf-idf BOW+	82.6
Ours	TDS Deconvolution	79.6
Ours	BERT Language Model	77.7
K19	HAN	76.7
K19	HAN + Dictionary	78.1

Table 5.3: Performances of dictionary-based methods (top), tf-idf BOW models (middle) and deep models (below). Note that different works use different datasets. f-score (F_1) in %.

We attempt a comparison of the different approaches used in the different related works as well as ours. While the scores achieved (see Table 5.3) are not strictly comparable, we can see clear tendencies. According to K19, a man-made dictionary is inferior to an automatically generated one. This is supported by the man-made lexicon in C18 performing subpar to their tf-idf BOW. An appropriate lexicon of explicit terms, on the other hand, can compete with a tf-idf BOW model, as we showed with both the dictionary lookup and the regression performance. This is further supported by the generated dictionary of K19 which competes with the deep Hierarchical Attention Network (HAN) [118]. Optimizations to the standard tf-idf BOW models are marked with the + sign. Restricting the POS tags to more likely ones found in explicit terms (C18) improves performance slighly. Using random undersampling to fight the imbalanced class problem (B18) increases performance drastically, however makes the problem somewhat different from the imbalanced problem. The final takeaway is that **deep models do not necessarily outperform shallow models**. Neither HAN, TDS, nor BERT deliver much higher scores than the dictionary-based or the BOW method.

5.4.4 *Qualitative Analysis*

In this section we analyze examples of explicit content lyrics and point to the inherent difficulty and subjectivity in classifying and even labelling such data.

EXPLICITNESS IN CONTEXT?

The highest difference in model performance we measured was between the deep TDS model (79.6% F_1) and the dictionary lookup (77.3% F_1). We analyzed why the TDS method performed better than the dictionary lookup by inspecting those examples that (i) were explicit, (ii) were classified as

clean by the dictionary lookup, and (iii) were detected as explicit by TDS with high confidence[9].

From the 13 examples analyzed, we found three main phenomena: (1) Four texts contained explicit terms that were not contained in the dictionary of explicit terms. Words such as *f**kin'*, *motherf**kers* were too rare to be included in the generated lexicon and other words like *fucking*, *cunt*, *cum*, *shit* were not uniquely contained in explicit lyrics. The reason why this is the case can be traced back to problems in the annotations or the fact that these words are relatively frequently used in lyrics. (2) Five texts whose explicitness arises in context rather than on a word level. Examples with violent context found were "organization with horns of satan performs the ancient rituals" or "bombin on mc's, crushin crews with ease". There were also instances of sexual content such as "give it to him down in the parking lot in the backseat, in the backseat of the car". Note that the words {give, it, to, him} in isolation do not belong to an explicit terms list and the sexuality arises from the context. Similarly in "(turn the lights on) so i can see that ass work". Also here, putting "ass" in an explicit terms dictionary is tempting but may not be ideal, as its meaning is not necessarily explicit. (3) Four texts appeared to have been mislabelled since no explicitness could be found. We found for three of them that the album the song is contained in is tagged as explicit. In cases as these, inheriting the label from the album is wrong, but it seems this is exactly what had happened here. In one Raggae lyric, in particular, we found no explicit content, so we suspect the song was mislabelled.

Since we found some annotation to be problematic, we will discuss difficulties that arise from annotating explicitness in lyrics.

HOW HARD IS THIS TASK?

As stated in the introduction, the explicit label is voluntary and we will argue that it is also somewhat subjective in its

9 The last layer of TDS outputs probabilities for the input text being explicit or clean. We looked at examples where the explicit class was predicted with at least 80% probability.

nature. There are lyrics which are not tagged as explicit although they have profanity in them. Consider for example the song *Bitch* by Meredith Brooks. While it already contains profanity in the title, it does not carry the explicit label and one can argue that in the context of the song, the term "bitch" is used as a contrastive term and to raise attention to the struggle the songwriter sees in her life, torn between potentially conflicting expectations of society ("I'm a little bit of everything - All rolled into one - I'm a bitch, I'm a lover - I'm a child, I'm a mother - I'm a sinner, I'm a saint - I do not feel ashamed").

Another example is *Check Your Head* by Buckcherry where it says "Ooh and you still bitch about your payments" where "bitch" is used as a verb and one can argue that the acceptance in this verb form is higher than in the noun form. A similar case where the part of speech influences the perceived level of profanity is *Hail Hail Rock 'n' Roll* by Discipline. It contains the line "the band starts to play loud as fuck".

We encounter a different kind of problem when dealing with substance abuse or other drug-related content. It is evident that the legal status of the substances mentioned plays a major role in how such content is labelled. This is further complicated by the fact that legislation about substances can vary wildly between different countries. The labels applied to this content are not culture-invariant, and furthermore changes in the societal view can lead to labels that are not relevant anymore. This, like other examples, shows why the labels applied to lyrics are subject to change in different cultures and time periods.

Another aspect that is very sensitive to time periods and cultures comes from words themselves: an inoffensive word can become offensive in slang or common language. One such example can be found in Johnny Cash's *The Christmas Guest*: "When the cock was crowing the night away - The Lord appeared in a dream to me". Here, cock means male chicken, as opposed to the offensive meaning that is now arguably more common.

We finally want to raise attention to the problem of genre confounding. We found that the genre *Hip Hop* contributed

by far the most to all explicit lyrics - 33% of all *Hip Hop* lyrics. Since only about 5% of the whole corpus are tagged as *Hip Hop*, this genre is highly overrepresented. This raises the question in how far our task is confounded with genre classification. When inspecting the explicit terms dictionaries we have created, we clearly see that genre bias reflected. The dictionary of 32 terms that we used for the dictionary lookup method consists approximately half of terms that are quite specific to the Rap genre, such as glock, gat, clip (gun-related), thug, beef, gangsta, pimp, blunt (crime and drugs). Finally, the terms holla, homie, and rapper are arguably no causes for explicit lyrics, but highly correlated with explicit content lyrics. Biasing an explicit lyrics detection model away from genres is an interesting future direction of work.

5.5 TOWARDS MUSIC EMOTION RECOGNITION

Just like in the explicit lyrics detection problem, we have treated before, listeners, to some extent, also disagree on which emotions are conveyed in a song. Ultimately, their judgement relies not only on factors inherent to the songs, but also to their socialization and other personal factors.

On the one hand, the task of **Music Emotion Recognition** (MER) has a long tradition in the MIR community. Its goal is to automatically identify which emotion / mood is conveyed in a song. Consequently, identifying songs as e.g. *happy* or *sad*, allows to recommend other tracks with similar emotion to the listener or generate playlists full of happy songs.

On the other hand, emotion recognition from text is of interest in NLP. The task originates from the more basic task of sentiment analysis. The goal of the latter is to predict if a text has a positive or a negative emotional valence. In the recent years, a transition from sentiment analysis to more complex formulations of emotion detection (e.g. joy, fear, surprise) [80] has become more visible; even tackling the problem of emotion in context [26].

While the music plays a central role in how the emotion in a song is perceived, it has also been shown that the mood can be inferred solely from the song text. We conducted preliminary experiments in MER based on song lyrics which we will describe in the following, proceeding as follows. We introduce two popular models of representing emotions in Section 5.5.1 and describe approaches for the conversion between different emotion representations. Then in Section 5.5.2, we review a state-of-the-art approach to MER and conduct our own preliminary experiment. In Section 5.5.3 we give an overview over the available datasets for MER, and recommend which kinds of datasets to use and why.

5.5.1 *Emotion Representations*

Two common models to represent emotion are Plutchik's wheel of emotion [100] and Russell's valence-arousal plane [101]. In the wheel of emotions, a fixed number of basic emotions is laid out as depicted in Figure 5.2. Opposing emotions are located in opposite positions, e.g. *joy* is above and *sadness* is below the center of the wheel - *anger* is left of the center and *fear* is right to it. The model also specifies how combinations of emotions can form more complex emotions, for example *serenity* and *acceptance* form *love*. Since this model puts all emotions into categories, it is also called a **categorical model** of emotion. As an alternative description, the valence-arousal model of emotion [101], locates every emotion in a continuous two-dimensional plane based on its valence (positive vs. negative) and arousal (excited vs. calm)[10]. Figure 5.3 illustrates the placement of emotions in the valence-arousal model. We find the prototypical emotions in the corners of the four quadrants of plane to be *joyful* (high valence, high arousal), *angry* (low valence, high arousal), *content* (high valence, low arousal), and *depressing* (low valence, low arousal). Between these extreme emotions, all other emotions can be placed. For example, *annoyed* and *bored* are both emotions of comparably

10 Sometimes, a third dimension of dominance is part of the model.

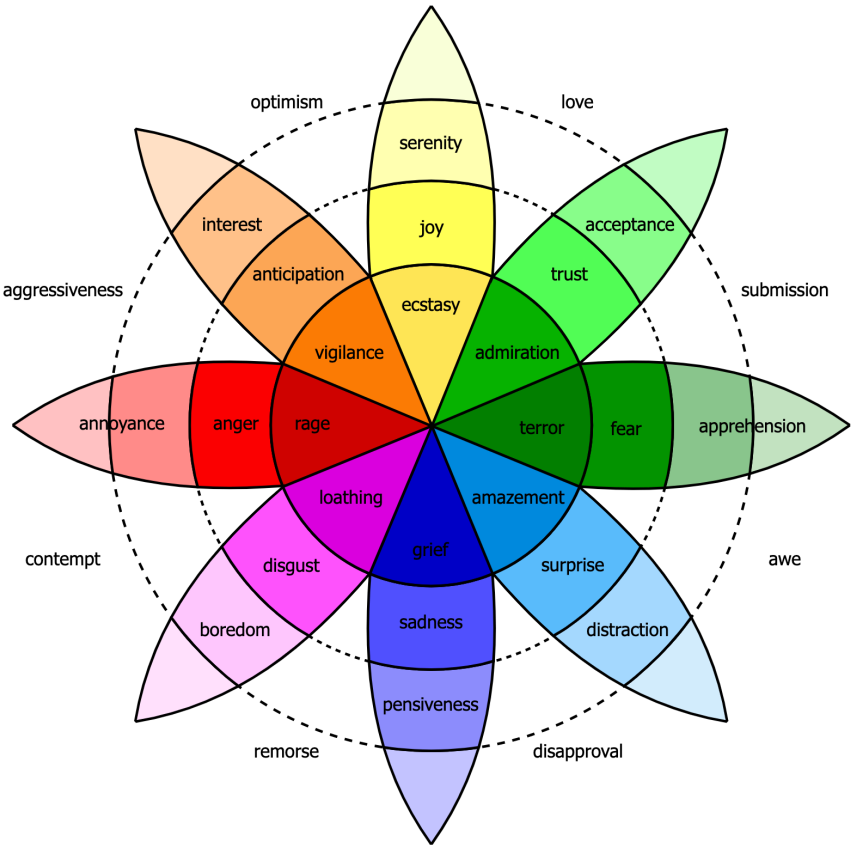


Figure 5.2: Emotion model of Plutchik. Illustration taken from Wikimedia Commons.

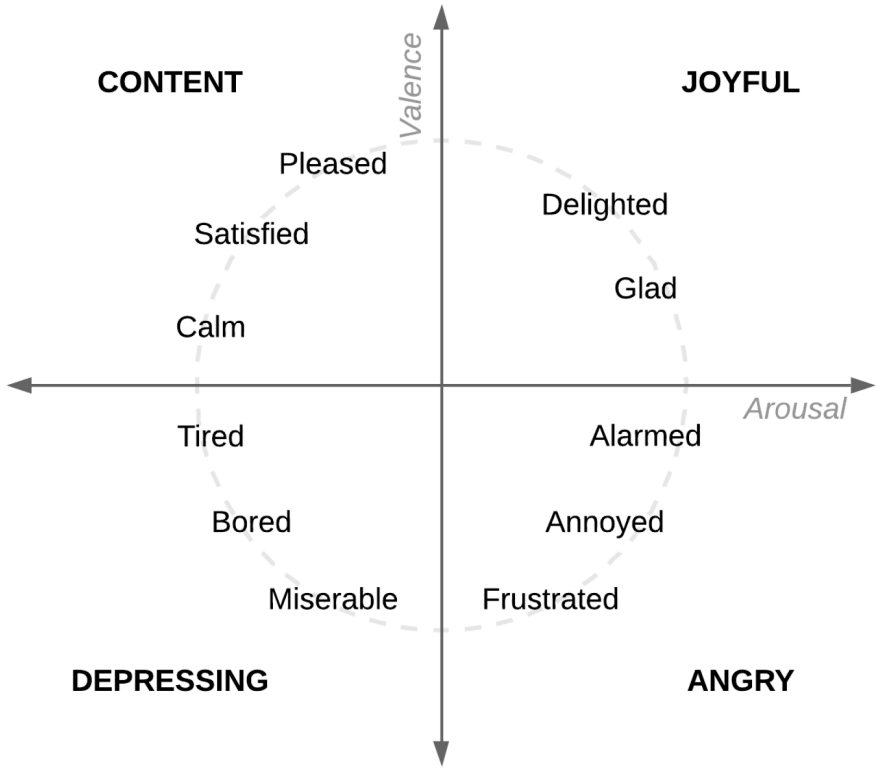


Figure 5.3: Placement of emotions in the valence-arousal model of Russell. Illustration taken from [89].

low valence, but *bored* has low arousal while *annoyed* has a high arousal. Since all emotions are represented by their intensites in two different dimensions, this model is often called a **dimensional model** of emotion.

EMOTION ANNOTATIONS

Researchers have created emotion lexicons, i.e. dictionaries where each word is manually annotated with either a basic emotion from the wheel of emotions [79] or a coordinate in the valence-arousal plane [114]. Annotating sentences or larger units of texts is very time-consuming, as new annotations have to be created for each new context. Therefore, to obtain gold labels for song lyrics, researchers have resorted to methods of distant supervision by leveraging **social tags from LastFM**. While such tags in principle can be any word (e.g. Rock, favorite songs, happy, best song ever, yeeeeeeeahhhh, 1975), these approaches [24, 58] define a list of social tags that are related to emotion (e.g. happy, anger, mellow, celebrate). Then lyrics datasets are filtered such that only lyrics associated with emotion tags pertain. The placement of the emotion associated (in the wheel or the plane) in turn is defined by the definition of the social tag in one of the emotion lexicons that we previously discussed.

CONVERTING REPRESENTATIONS

As both emotion representations have their advantages and disadvantages, some applications lean themselves more to one of them. For instance, you may want to build an emotion detection system based on the dimensional model, but only have categorical annotations at hand. To that end, there have been several attempts at converting between the different emotion representations. For example, in [107] the authors convert from categorical to dimensional model by using a dictionary lookup and a BOW model of composition. Specifically, in their annotation a text T is labelled with probabilities of categorical emotions e_i. Then, given a lexicon of emotion annotations according to the dimensional model [114], they look up the valence and arousal $V(e_i), A(e_i)$ and weight them according to their probabili-

ties $P(e_i)$ in T. While this model simplifies the conversion problem in several ways, e.g. by modelling T as BOW and by equating the name of the emotion with the actual emotion, they demonstrate the approach to yield promising results. While this approach can be seen as assuming the composition function from word level to text level as a weighted average $(V, A) = (\sum_i P(e_i) \cdot V(e_i), \sum_i P(e_i) \cdot A(e_i))$, recently supervised learning has been employed to learn a more complex and precise mapping from the categorical to the valence-arousal-dominance representation [91].

5.5.2 *Lyrics-based Music Emotion Recognition*

In the following, we describe our preliminary experiments with music emotion regression, which are restricted to the static and lyrics-based case[11]. For this, we closely follow the problem formulation of [33]. The goal of this task is to predict as closely as possible the valence and arousal (VA) of a song, which has previously been annotated with VA. We only consider the lyrics-based regression problem in the following, leaving the multimodal case for future work.

DATASET

Deezer has created and made available VA annotations for 18k English songs[12]. These annotations have been constructed using both the social tags filtering and the categorical-to-dimensional conversion methods described above. Since the dataset does not come with lyrics (for obvious copyright reasons), we aligned it to our WASABI Song Corpus. We successfully aligned 16k of the original 18k of the lyrics, which makes our dataset somewhat different from the one Deezer used in their experiments.

11 The end goal of our experiments is dynamic emotion modelling [22], i.e. assuming that emotion can change over the course of the song. Since we so far have no positive results on that, we present our findings on static emotion modelling, which will be the baseline to compare our future dynamic emotion approaches to.

12 https://github.com/deezer/deezer_mood_detection_dataset

Model	*valence*	*arousal*	*average*
Best feature engineering [33]	14.0	3.2	8.6
Best neural approach [33]	13.4	2.6	8.0
RNN with attention	10.9	4.2	7.6
Finetuned BERT	**17.2**	**8.6**	**12.9**

Table 5.4: R^2 scores in % of the different models on the Deezer lyrics dataset for the different dimensions valence and arousal as well as their average.

MODELS

We compare four models which all work in a comparable way and are described in the following. First, they extract features from the lyrics, then they predict the valence and arousal of that song text. Since the problem is formulated as regression against the VA gold labels, the supervised learning works by minimizing the mean squared error between predicted and gold VA values. The models differ in the feature extraction step, which can be based on convolutional filters or recurrent layers (in the case of neural approaches) or even on hand-crafted feature computation in the case of the feature-engineering approach. As baselines, we report two results of [33]: the best-performing feature engineering approach [57] and the best neural approach of Deezer (a combination of a CNN and an RNN). Then, we implement a similar neural architecture in the form of an RNN with attention mechanism [37]. And lastly, we finetune the transformer-based pretrained language model BERT [35], which has shown state-of-the-art performance in numerous text classification tasks before.

PRELIMINARY RESULTS

Figure 5.4 shows both the results reported by Deezer on their dataset of 18k lyrics and our results on our aligned 16k lyrics dataset. One important finding of Deezer was that, unlike in most other NLP tasks, the feature engineering approach is on par with the neural approach (R^2 average 8.6%) vs. (R^2 average 8.0%). Then, our RNN with attention achieves similar results (R^2 average 7.6%) as the best

approaches reported by Deezer. Furthermore, it has been observed in previous work that the arousal is much harder to predict from the lyrics than the valence - and our models confirm this. Note that our RNN exhibits a different trade-off between valence and arousal performance, trading in lower valence for higher arousal. Finally, BERT performs far superior (R^2 average 12.9%) in this task as it can leverage its pretraining to create more useful document vectors for each song text. While this may not be too surprising in light of BERT's previous successes in text classification, it is noteworthy that this is the first neural approach we know of that performs clearly better than the best feature engineering approach for music emotion regression.

5.5.3 *Which Dataset to use?*

EXISTING SONG-EMOTION DATASETS

Only a few datasets exist in which songs are associated with emotions. We describe two of them and argue based on a comparative experiment how to select a high quality dataset for lyrics-based MER. The first dataset is the previously described **Deezer corpus** of 18k song-emotion associations. Then, the second dataset is called **MoodyLyrics4Q**, in which 2,000 songs are associated with each one emotion from the corners of the valence-arousal plane: joy, sadness, anger or fear (cf. Section 5.5.1). The authors [23, 24] "polarize" the emotions, which means that only songs which are highly associated with an emotion e from the corners of the plane and lowly associated with emotions different from e, are finally tagged with e. This aims at reducing noise in the dataset, since unclear cases are removed.

CONSEQUENCES OF CONVERSION AND POLARIZATION

Note that besides the dataset size, the Deezer corpus and MoodyLyrics4Q differ in the following two regards. First, in the Deezer corpus the emotions were converted from categorical into dimensional representation; the labels were

Dataset	*Converted*	*Polarized*	*Domain*
Deezer corpus	yes	no	Lyrics
Rappler converted	yes	no	News
MoodyLyrics4Q	no	yes	Lyrics
Rappler polarized	no	yes	News

Table 5.5: Different datasets to disentangle the factors conversion, polarization and domain.

not polarized. In MoodyLyrics4Q on the other hand, the labels were not converted, but polarized. To test the impact of representation conversion and polarization on emotion recognition performance, we introduce a third corpus called the **Rappler corpus**, previously described in [107]. This corpus does not contain lyrics, but 14k news articles, each tagged with a probability distribution over the categorical emotions of the Plutchik model. The labels are manually annotated via crowdsourcing. We have selected it to allow for comparing our emotion recognition models on different text domains.

We created two versions of the Rappler corpus: **Rappler polarized** is a subset which only contains texts which are associated with one emotion with more than 50% probability mass and tagged the text with this emotion; we discarded less certain cases. Then, we created **Rappler converted** by applying the categorical-to-dimensional conversion procedure (described in Section 5.5.1) to the Rappler corpus. We consequently obtained three datasets from two domains and different conversion and polarization status; as summarized in Table 5.5.

We measure the effects of the domain, the conversion and the polarization on emotion recognition performance as follows. Based on BERT, we experiment the different datasets as follows: we perform emotion regression on the converted (dimensional) corpora and emotion classification on the non-converted (categorical) corpora. First, we compare the datasets Deezer and Rappler converted. The emotion regression on the Deezer dataset yields 12.9% average R^2, as previously reported in Section 5.5.2 while on the

Rappler converted dataset we achieve an average 31.6% R^2. Second, we compare MoodyLyrics4Q with Rappler polarized in emotion classification with the three most common emotions (to ensure we have enough data). The results on both corpora are similar, slightly above 70% F_1 for 3-class classification.

RECOMMENDATION

The conclusions from these experiments are as follows. From our experiments, we cannot clearly conclude which effect the **conversion** from categorical into dimensional representation has. However, the promising result of Rappler converted hints that conversion does not drastically diminish performance. **Polarization** on the other hand appears to improve performance drastically, as shown by emotion classification results on MoodyLyrics4Q being similar to those on Rappler polarized. The latter result also hints to the **lyrics domain** being just as viable for emotion recognition as the news domain. Consequently, our recommendation for selecting or creating a lyrics emotion dataset is to **definitely use polarization** and to **be optimistic about conversion**.

5.6 CONCLUSION

Classifying song lyrics as explicit or clean is an inherently hard task to accomplish since what is considered offensive strongly depends on cultural aspects that can change over time. We showed that shallow models solely based on a dictionary of profane words achieve a performance comparable to deep neural networks. We argued that even the hand-labelling is highly subjective, making it problematic to automatically detect if a song text should be tagged as explicit or clean. Furthermore, we have presented our preliminary work on lyrics-based emotion recognition (ER). We have described the different competing emotion representations that emphasize different aspects of emotions and pointed out attempts to harmonize them. We have conducted a comparative experiment and shown that pre-

trained language models excel in this text classification task, with BERT clearly outperforming the previous state of the art. We found that datasets of sufficient quality and size are hard to get by, but we have given recommendations on how to construct such datasets based on our understanding of existing work and our experiments. We finally want to point to the assumption that musical emotion and lyrical emotion are congruent. This may not always be the case and such a detail causes ER to become more subjective.

For ER, we are currently experimenting with dynamic emotion recognition, i.e. under the assumption that emotion changes during a song. We believe that understanding the changes in emotion can improve overall detection rates. For explicit lyrics detection, we propose as a possible simplification and objectification to study the local detection of explicit content. If we present an authority a report on found trigger words, found contextual sexual content, and alike, they can come to their own subjective conclusion about the final label of the text. For both tasks, explicit lyrics detection and ER, we think that the intended emotion and the intended explicitness can be inferred with higher accuracy when extra-linguistic context is available, such as the music clip, the music, or information about the band from interviews etc.

6

THE ANNOTATED WASABI SONG CORPUS

In this Chapter, we describe the annotated WASABI Song Corpus, as resulting from enriching the initial dataset described in Section 2.3 with NLP annotations of different levels. The annotations result from the application of the methods we proposed in this work to extract relevant information from the lyrics.[1].

CONTENTS

6.1 INTRODUCTION

We have previously introduced the WASABI Song Corpus, a large corpus of 2.10M songs (1.73M with lyrics) enriched with various kinds of metadata extracted from music databases on the Web and resulting from the processing of audio analysis (see Section 2.3). Alongside, we have given an overview over its key statistics, such as the language and genre distributions and the years of publication of its songs.

Based on the results of the NLP methods for lyrics analysis which we have proposed in the previous chapters, we have annotated the lyrics in the WASABI Song Corpus on the following levels: their structure segmentation, the explicitness of the lyrics content, the salient passages of a song, the addressed topics and the emotions conveyed. We detail these annotations in Section 6.2. An analysis of the correlations among the above mentioned annotation layers reveals interesting insights about the song corpus.

1 This work will be published at LREC 2020.

For instance, we demonstrate the change in corpus annotations diachronically: we show that certain topics become more important over time and others are diminished. We also analyze such changes in explicit lyrics content and expressed emotion (see Section 6.3). Finally, in Section 6.4 we give an overview over the most relevant annotations in the WASABI Song Corpus and conclude.

6.2 CORPUS ANNOTATIONS

LYRICS STRUCTURE

Previously, in Chapter 3 we have proposed a method to segment lyrics based on their repetitive structure in the form of a self-similarity matrix (SSM). Figure 6.1 shows a line-based SSM for the song text written on top of it[2]. The song text consists of seven segments and shows the typical repetitive structure of a Pop song. The main diagonal is trivial, since each line is maximally similar to itself. Notice further the additional diagonal stripes in segments S_2, S_4 and S_7; this indicates a repeated part, typically the chorus. Based on the simple idea that eyeballing an SSM will reveal (parts of) a song's structure, we proposed a Convolutional Neural Network architecture that successfully learned to predict segment borders in the lyrics when "looking at" their SSM.

In the WASABI Interactive Navigator, the line-based SSM of a song text can be visualized. It is toggled by clicking on the violet-blue square on top of the song text. For a subset of songs the color opacity indicates how repetitive and representative a segment is, based on the fitness metric that we defined in Section 4.3.2. For illustration, note how in Figure 6.1 the segments S_2, S_4 and S_7 are shaded more darkly than the other ones. As highly fit (opaque) segments often coincide with a chorus, this is a first approximation of chorus detection. A more complete labelling of the segments as Intro, Verse, Bridge, Chorus etc seems still out of reach, given the variability in the set of structure types

2 https://wasabi.i3s.unice.fr/#/search/artist/Britney%20Spears/album/In%20The%20Zone/song/Everytime

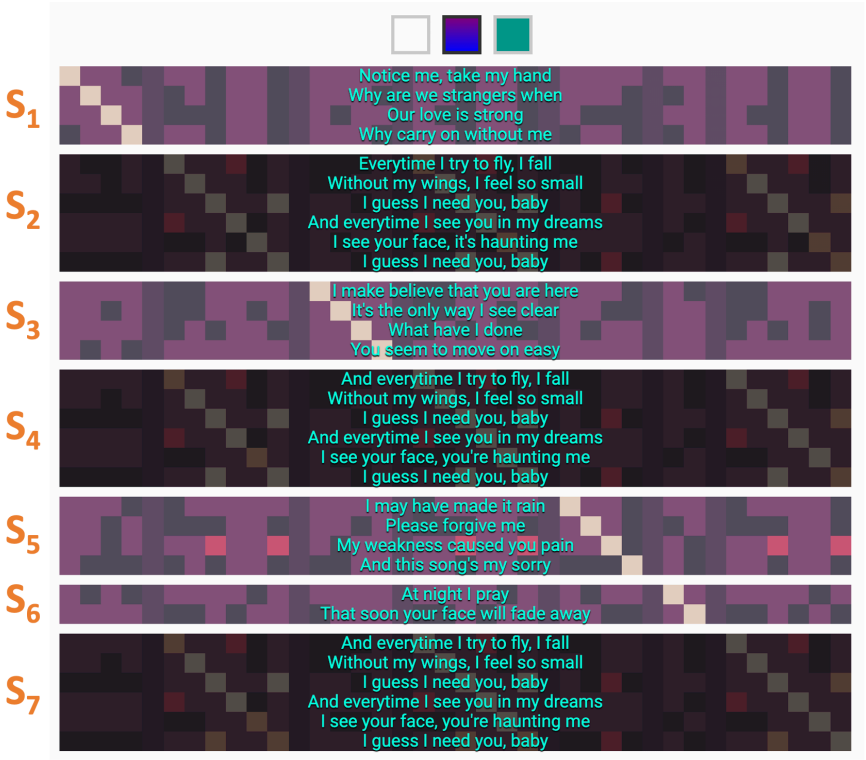


Figure 6.1: Structure of the lyrics of *Everytime* by Britney Spears as displayed in the WASABI Interactive Navigator.

provided in the literature according to different genres [16, 109]. For each song text we provide an SSM based on a normalized character-based edit distance[3] on two levels of granularity to enable other researchers to work with these structural representations: line-wise similarity and segment-wise similarity.

LYRICS SUMMARY

In Chapter 4 we have introduced a method for extractive summarization of song lyrics which is reminiscent of audio thumbnailing approaches that summarize audio. Figure 6.2 shows an example summary of four lines length obtained with our proposed method. It is toggled in the WASABI Interactive Navigator by clicking on the green square on top of the song text. The four-line summaries of 50k En-

3 In our segmentation experiments we found this simple metric to outperform more complex metrics that take into account the phonetics or the syntax.

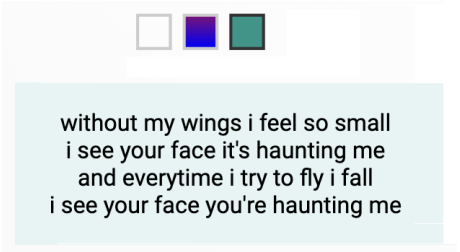


Figure 6.2: Summary of the lyrics of *Everytime* by Britney Spears as displayed in the WASABI Interactive Navigator.

glish lyrics (cf. Section 4.4.1) is freely available within the WASABI Song Corpus; the Python code of the applied summarization methods is also available[4].

EXPLICIT LANGUAGE IN LYRICS

In Chapter 5 we have compared different approaches for automated explicit lyrics detection. We found a very simple method of checking against an automatically generated swear word lexicon to perform on par with much more complex models such as BERT [35] as a text classifier. Our corpus contains 52k tracks labelled as explicit and 663k clean (not explicit) tracks[5]. We have trained a classifier (77.3% f-score on test set) on the 438k English lyrics which are labelled and classified the remaining 455k previously untagged English tracks. We provide both the predicted labels in the WASABI Song Corpus and the trained classifier to apply it to unseen text.

EMOTIONAL DESCRIPTION

As previously described, the Deezer corpus consists of valence-arousal annotations for 18k songs (cf. Section 5.5.3). We aligned the Deezer corpus to our WASABI Song Corpus since the Deezer corpus lacks the song lyrics (due to obvious copyright reasons). In Figure 6.3 the green dots visualize the emotion distribution of these songs. Based on their annotations, we trained an emotion regression model on the aligned portion of the WASABI Song Corpus using

4 https://github.com/TuringTrain/lyrics_thumbnailing

5 Labels provided by Deezer. Furthermore, 625k songs have a different status such as unknown or censored version.

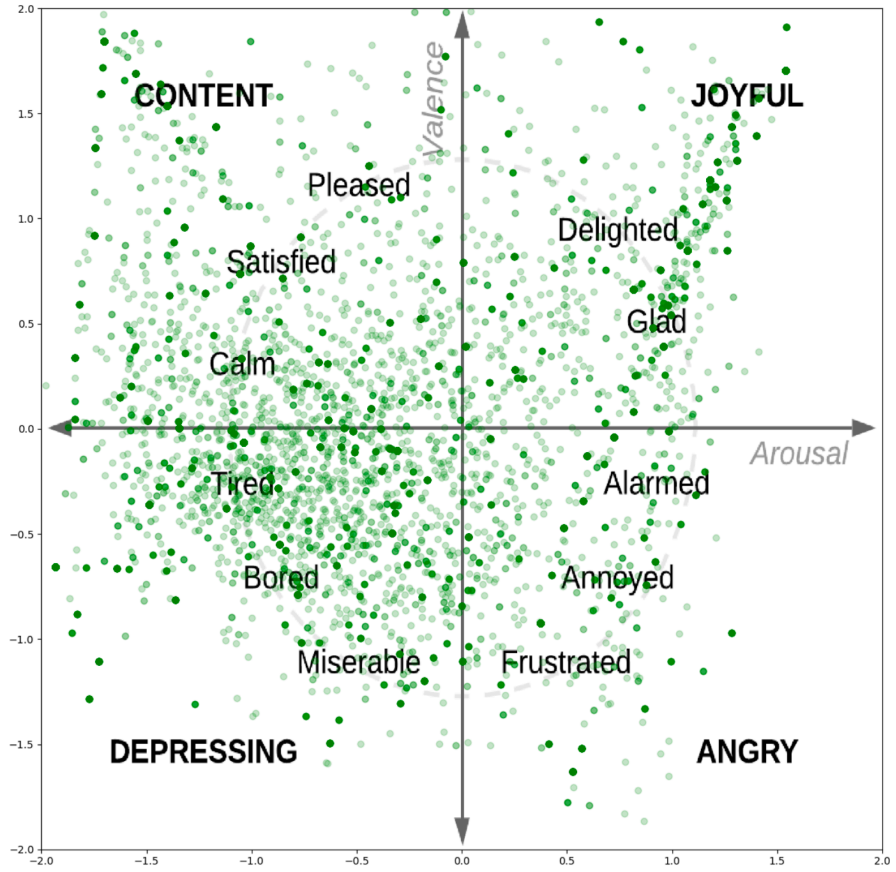


Figure 6.3: Emotion distribution in the corpus in the valence-arousal plane. Illustration without scatterplot taken from [89].

BERT, with an evaluated 0.44/0.43 Pearson correlation/Spearman correlation for valence and 0.33/0.31 for arousal on the test set. We integrated Deezer's labels into the WASABI Song Corpus and also provide the valence-arousal predictions for the 1.73M tracks with lyrics. We also provide the LastFM social tags (276k) and emotion tags (87k entries) to facilitate researchers to build variants of emotion recognition models.

TOPIC MODELLING

We built a topic model on the lyrics of our corpus using Latent Dirichlet Allocation (LDA) [14]. We determined the hyperparameters α, η and the topic count such that the coherence was maximized on a subset of 200k lyrics. We

Figure 6.4: Topic War

Figure 6.5: Topic Death

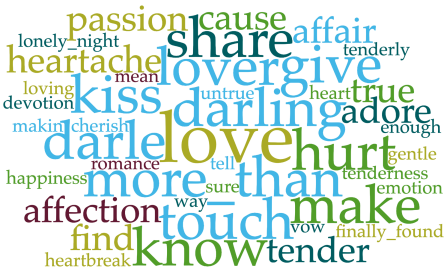


Figure 6.6: Topic Love

Figure 6.7: Topic Family

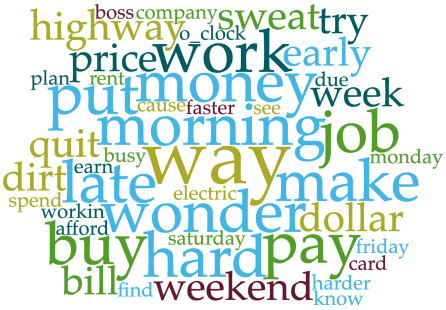


Figure 6.8: Topic Money

Figure 6.9: Topic Religion

then trained a topic model of 60 topics on the unique English lyrics (1.05M). We have manually labelled a number of more recognizable topics. Figures 6.4-6.9 illustrate these topics with word clouds[6] of the most characteristic words per topic. For instance, the topic Money contains words of both the field of earning money (job, work, boss, sweat) as well as spending it (pay, buy). The topic Family is both about the people of the family (mother, daughter, wife) and the land (sea, valley, tree). We provide the topic distribution of our LDA topic model for each song and make available

6 made with https://www.wortwolken.com/

the trained topic model to enable its application to unseen lyrics.

6.3 DIACHRONIC ANALYSIS

We examine the changes in the annotations over the course of time by grouping the corpus into decades of songs according to the distribution shown in Figure 2.4c.

CHANGES IN TOPICS

The importance of certain topics has changed over the decades, as depicted in Figure 6.10a. Some topics have become more important, others have declined, or stayed relatively the same. We define the importance of a topic for a decade of songs as follows: first, the LDA topic model trained on the full corpus gives the probability of the topic for each song separately. We then average these song-wise probabilities over all songs of the decade. For each of the cases of growing, diminishing and constant importance, we display two topics. The topics War and Death have appreciated in importance over time. This is partially caused by the rise of Heavy Metal in the beginning of the 1970s, as the vocabulary of the Death topic is very typical for the genre (see for instance the "Metal top 100 words" in [45]). We measure a decline in the importance of the topics Love and Family. The topics Money and Religion seem to be evergreens as their importance stayed rather constant over time.

CHANGES IN EXPLICITNESS

We find that newer songs are more likely being tagged as having explicit content lyrics. Figure 6.10b shows our estimates of explicitness per decade, the ratio of songs in the decade tagged as explicit to all songs of the decade. Note that the Parental Advisory Label was first distributed in 1985 and many older songs may not have been labelled retroactively. The depicted evolution of explicitness may therefore overestimate the "true explicitness" of newer music and underestimate it for music before 1985.

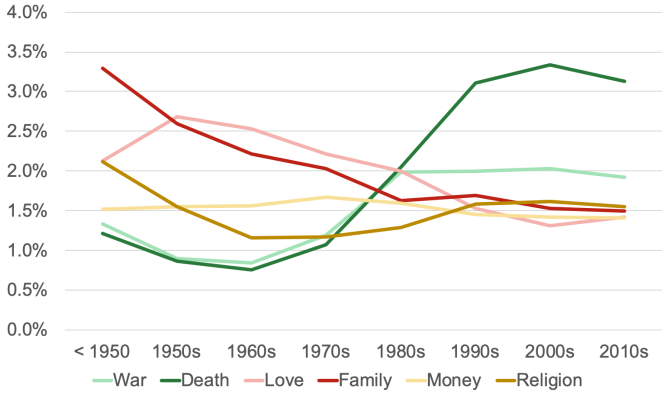
CHANGES IN EMOTION

We estimate the emotion of songs in a decade as the average valence and arousal of songs of that decade. We find songs to decrease both in valence and arousal over time. This decrease in positivity (valence) is in line with the diminishment of positively connotated topics such as Love and Family and the appreciation of topics with a more negative connotation such as War and Death.

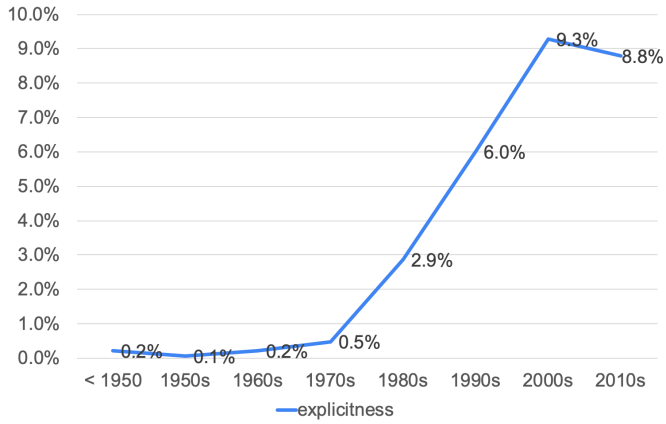
6.4 CONCLUSION

In this chapter we have described the WASABI dataset of songs, in particular the lyrics annotations resulting from the applications of the methods we proposed to extract relevant information from the lyrics. So far, lyrics annotations concern their structure segmentation, their topic, the explicitness of the lyrics content, the summary of a song and the emotions conveyed. Some of those annotation layers are provided for all the 1.73M songs included in the WASABI corpus, while some others apply to subsets of the corpus, due to various constraints described in this chapter. Table 6.1 summarizes the most relevant annotations in our corpus.

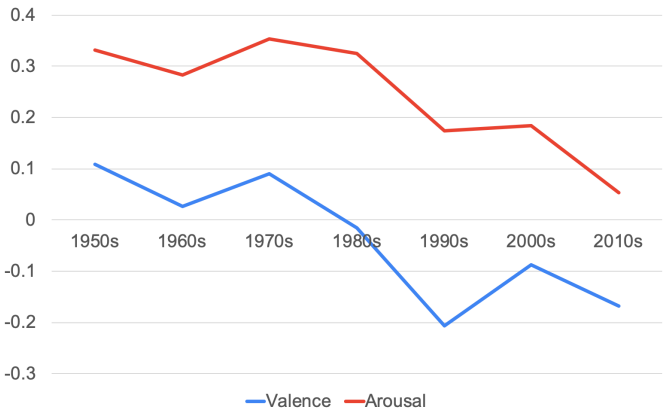
As the creation of the resource is still ongoing, we plan to integrate an improved emotional description in future work. In [5] the authors have studied how song writers influence each other. We aim to learn a model that detects the border between heavy influence and plagiarism.



(a) Evolution of topic importance



(b) Evolution of explicit content lyrics



(c) Evolution of emotion

Figure 6.10: Evolution of different annotations during the decades.

Annotation	*Labels*	*Description*
Lyrics	1.73M	segments of lines of text
Languages	1.73M	36 different ones
Genre	1.06M	528 different ones
Last FM id	326k	UID
Structure	1.73M	SSM $\in \mathbb{R}^{n \times n}$ (n: length)
Social tags	276k	$\mathbb{S} = \{\text{rock, joyful, 90s, ...}\}$
Emotion tags	87k	$\mathbb{E} \subset \mathbb{S} = \{\text{joyful, tragic, ...}\}$
Explicitness	715k	True (52k), False (663k)
Explicitness ♣	455k	True (85k), False (370k)
Summary♣	50k	four lines of song text
Emotion	16k	(valence, arousal) $\in \mathbb{R}^2$
Emotion♣	1.73M	(valence, arousal) $\in \mathbb{R}^2$
Topics♣	1.05M	Prob. distrib. $\in \mathbb{R}^{60}$
Total tracks	2.10M	diverse metadata

Table 6.1: Most relevant song-wise annotations in the WASABI Song Corpus. Annotations with ♣ are predictions of our models.

7

CONCLUSION

This Thesis presents and discusses the relevant results on our research on Natural Language Processing for Music Information Retrieval. We have performed a deep analysis of song lyrics, focusing on their structure, content and perception. This work has been done as fundamental research towards establishing the inclusion of song lyrics in MIR applications with the end goal of improving the music listening experience for everyone.

In Chapter 2 we have given a brief overview over the WASABI Project in the context of which this Thesis has been written. We have clarified the differences to similar projects and introduced the reader to the WASABI Song Corpus, the central dataset we have used for experimentation throughout this work.

In Chapter 3 we have dealt with the problem of detecting the structure in lyrics. We have reduced the problem to the subtasks lyrics segmentation and segment labelling. We have introduced a model that efficiently segments the lyrics. More specifically, we have addressed the task of lyrics segmentation on synchronized text-audio representations of songs. For the songs in the corpus DALI where the lyrics are aligned to the audio, we have derived a measure of alignment quality specific to our task of lyrics segmentation. Then, we have shown that exploiting both textual and audio-based features lead our Convolutional Neural Network-based model to significantly outperform the state-of-the-art system for lyrics segmentation that relies on purely text-based features. Moreover, we have shown that the advantage of a bimodal segment representation pertains even in the case where the alignment is noisy. This indicates that a lyrics segmentation model can be improved in most situations by enriching the segment representation by another modality (such as audio). Finally, we have briefly discussed the task of segment labelling and gave an

approximation to chorus detection based on clustering the lyrics segments using different similarity metrics.

In Chapter 4 we have dealt with the problem of representing the content of lyrics. We have explained the limitations we have found with representations based on topic models and information extraction. We then have introduced our final content representation by means of text summarization. We have proposed a model to summarize the lyrics in a way that respects their intimate relation to music. More specifically, we have defined and addressed the task of lyrics summarization. We have applied both generic unsupervised text summarization methods (TextRank and a topic-based method we called TopSum), and a method inspired by audio thumbnailing on 50k lyrics from the WASABI corpus. We have carried out an automatic evaluation on the produced summaries computing standard metrics in text summarization, and a human evaluation with 26 participants, showing that using a fitness measure transferred from the musicology literature, we can amend generic text summarization algorithms and produce better summaries.

In Chapter 5 we have dealt with the problem how lyrics are perceived in the world. As an instantiation we have discussed the problem of detecting explicit content in a song text. This task has proven to be very hard and we have shown that the difficulty partially arises from the subjective nature of perceiving lyrics in one way or another depending on the context. Classifying song lyrics as explicit or clean is an inherently hard task to accomplish since what is considered offensive strongly depends on cultural aspects that can change over time. We have shown that shallow models solely based on a dictionary of profane words achieve a performance comparable to deep neural networks. We have argued that even the hand-labelling is highly subjective, making it problematic to automatically detect if a song text should be tagged as explicit or clean. Finally, we have glanced at the problem of how emotions are perceived in lyrics: we have presented our preliminary results on Emotion Recognition.

In Chapter 6 we have described the annotated WASABI Song Corpus, as resulting from enriching the initial dataset described in Section 2.3 with NLP annotations of different levels. The annotations have resulted from the application of the methods we proposed in this work to extract relevant information from the lyrics. So far, lyrics annotations concern their structure segmentation, their topic, the explicitness of the lyrics content, the summary of a song and the emotions conveyed. Some of those annotation layers are provided for all the 1.73M songs included in the WASABI corpus, while some others apply to subsets of the corpus, due to various constraints described.

RECSYS CHALLENGE 2018

We participated in the RecSys Challenge 2018[1] as members of of the D2KLab team. The Challenge focused on music recommendation, specifically the task of automatic playlist continuation. The idea is to recommend additional songs for a playlist to make playlist creation easier, as well as to extend listening beyond the end of existing playlists. The ground truth for training a machine learning model for playlist continuation, was the Million Playlist Dataset from Spotify, a public dataset of playlists, consisting of a large number of playlist titles and associated track listings. The evaluation contained a set of playlists from which a number of tracks had been withheld. The task was then to predict the missing tracks in those playlists. Our team proposed an ensemble strategy of different RNNs leveraging pre-trained embeddings representing tracks, artists, albums, and titles as inputs. Our specific contribution to the team effort was to align the Challenge dataset to the lyrics in our WASABI Song Corpus, and then extract such features from the lyrics that model different dimensions of the lyrics, such as vocabulary, style, semantics, orientation towards the world, and song emotion. Our lyrics features were used along the other features in the RNN where they contributed to improve the performance of our playlist completion approach [81]. This finding supports the hypothesis that

1 http://www.recsyschallenge.com/2018/

lyrics are a valuable addition for numerous applications in Music Information Retrieval.

7.1 PERSPECTIVES

The research we have conducted in this Thesis leaves space for future improvements and opens up possibilities for different applications in MIR. In the following, we enumerate some ideas we have for interesting future work to broaden and deepen the path of our research.

For **dataset creation** we think that a broad range of the applications we have discussed in this Thesis can profit from a resource such as Genius[2], a crowdsourcing platform for lyrics annotations. Here, parts of the lyrics are annotated and explained or given background information by the platform users. Such structured contextual data could help improve extracting the most important content in the lyrics.

For **multimodality**, we have shown improvements of lyrics segmentation performance when using both the text and the audio modality. We envision both using more modalities (such as an additional aligned video clip) and experimenting such an approach on more tasks, such as the tasks we worked on - summarization, explicit lyrics detection, emotion recognition - and beyond. We are hopeful, that the prerequisite for this, multimodal datasets, are becoming more and more available [74, 89].

For **music search engines** we are excited to see our newly released WASABI Song Corpus put to use in the MIR landscape. While we have provided NLP annotations for our two million song dataset, we will also continue working in the WASABI Project towards more complex and useful search interfaces to facilitate searches such as *find songs where the chorus talks about hope, but the verse talks about struggle*.

For **music recommendation** we envision a deeper integration of lyrics-based knowledge. While we obtained promising results in the RecSys Challenge 2018, as described above, that approach was based on manual feature

2 https://genius.com/

engineering. The advent of pretrained language models such as BERT [35] paves the way to the extraction of even more useful features from the lyrics.

In the spirit of [64] who learn **style transfer** for lyrics between the musical genres Hip Hop and Pop, we imagine to create lyrics from prose text or spoken language. To produce a text of similar length, in a first step, we may need to summarize the input text to then add the characteristic traits of lyrics such as rhyme and repetition. This might be a valuable help to composers, assisting them in rapdily writing lyrics to trending topics or news.

Finally, there is evidence that musical emotion arises strongly from the dynamics [22] and is formed by expectations [38]. This line of **dynamic emotion modelling** has seen little attention, also because of the lack of larger datasets [27, 106]. We believe that modelling emotion dynamically, e.g. one emotion per sentence instead of per document, will ultimately improve emotion recognition performance.

BIBLIOGRAPHY

[1] Mehdi Allahyari, Seyed Amin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. "Text Summarization Techniques: A Brief Survey." In: *CoRR* abs/1707.02268 (2017).

[2] A. Allik, F. Thalmann, and M. Sandler. "MusicLynx: Exploring music through artist similarity graphs." In: *Companion Proc. (Dev. Track) The Web Conf. (WWW 2018)*. Lyon, France, 2018.

[3] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. "Leveraging Linguistic Structure For Open Domain Information Extraction." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, 2015, pp. 344–354.

[4] Rachit Arora and Balaraman Ravindran. "Latent dirichlet allocation based multi-document summarization." In: *Proceedings of the second workshop on Analytics for noisy unstructured text data*. ACM. 2008, pp. 91–97.

[5] Jack Atherton and Blair Kaneshiro. "I Said it First: Topological Analysis of Lyrical Influence Networks." In: *ISMIR*. 2016, pp. 654–660.

[6] A. Baratè, L. A. Ludovico, and E. Santucci. "A Semantics-Driven Approach to Lyrics Segmentation." In: *2013 8th International Workshop on Semantic and Social Media Adaptation and Personalization*. 2013, pp. 73–79.

[7] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. "Variations of the Similarity

Function of TextRank for Automated Summarization." In: *CoRR* abs/1602.03606 (2016).

[8] Mark A. Bartsch and Gregory H. Wakefield. "Audio Thumbnailing of Popular Music Using Chroma-based Representations." In: *Trans. Multi.* 7.1 (2005), pp. 96–104. ISSN: 1520-9210.

[9] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. "Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter." In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019, pp. 54–63.

[10] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. "Jointly Learning to Extract and Compress." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Portland, Oregon: Association for Computational Linguistics, 2011, pp. 481–490. ISBN: 978-1-932432-87-9.

[11] Linn Bergelid. *Classification of explicit music content using lyrics and music metadata*. 2018.

[12] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. "The Million Song Dataset." In: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*. 2011.

[13] Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. "Automatic labelling of topics with neural embeddings." In: *arXiv preprint arXiv:1612.05340* (2016).

[14] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation." In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[15] Cristina Bosco, Felice Dell'Orletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. "Overview of the EVALITA 2018 Hate Speech Detection Task." In: *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the*

Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy. 2018.

[16] David Brackett. *Interpreting Popular Music*. Cambridge University Press, 1995. ISBN: 9780521473378.

[17] M. Buffa and J. Lebrun. "Real time tube guitar amplifier simulation using WebAudio." In: *Proc. 3rd Web Audio Conference (WAC 2017)*. London, UK, 2017.

[18] M. Buffa and J. Lebrun. "Web Audio Guitar Tube Amplifier vs Native Simulations." In: *Proc. 3rd Web Audio Conf. (WAC 2017)*. London, UK, 2017.

[19] Michel Buffa, Jerome Lebrun, Jari Kleimola, Stéphane Letz, et al. "Towards an open Web Audio plugin standard." In: *Companion Proceedings of the The Web Conference 2018*. International World Wide Web Conferences Steering Committee. 2018, pp. 759–766.

[20] Michel Buffa, Jerome Lebrun, Johan Pauwels, and Guillaume Pellerin. "A 2 Million Commercial Song Interactive Navigator." In: *WAC 2019 - 5th WebAudio Conference 2019*. Trondheim, Norway, 2019.

[21] Michel Buffa, Jerome Lebrun, Guillaume Pellerin, and Stéphane Letz. "WebAudio Plugins in DAWs and for Live Performance." In: *14th International Symposium on Computer Music Multidisciplinary Research (CMMR'19)*. 2019.

[22] Marcelo Caetano, Athanasios Mouchtaris, and Frans Wiering. "The role of time in music emotion recognition: Modeling musical emotions from time-varying music features." In: *International Symposium on Computer Music Modeling and Retrieval*. Springer. 2012, pp. 171–196.

[23] Erion Çano. "Text-based Sentiment Analysis and Music Emotion Recognition." PhD thesis. Turin, Italy: Computer Engineering, Politecnico di Torino, 2018.

[24] Erion Çano and Maurizio Morisio. "Music Mood Dataset Creation Based on Last.fm Tags." In: *2017 International Conference on Artificial Intelligence and Applications, Vienna Austria*. 2017.

[25] Wei Chai and Barry Vercoe. "Music thumbnailing via structural analysis." In: *Proceedings of the eleventh ACM international conference on Multimedia*. 2003, pp. 223–226.

[26] Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. "SemEval-2019 task 3: EmoContext contextual emotion detection in text." In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019, pp. 39–48.

[27] Yu-An Chen, Yi-Hsuan Yang, Ju-Chiang Wang, and Homer Chen. "The AMG1608 dataset for music emotion recognition." In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 693–697.

[28] H. T. Cheng, Y. H. Yang, Y. C. Lin, and H. H. Chen. "Multimodal structure segmentation and analysis of music using audio and textual information." In: *2009 IEEE International Symposium on Circuits and Systems*. 2009, pp. 1677–1680.

[29] Jianpeng Cheng and Mirella Lapata. "Neural Summarization by Extracting Sentences and Words." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 484–494.

[30] Hyojin Chin, Jayong Kim, Yoonjong Kim, Jinseop Shin, and Mun Y Yi. "Explicit Content Detection in Music Lyrics Using Machine Learning." In: *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE. 2018, pp. 517–521.

[31] Alice Cohen-Hadria and Geoffroy Peeters. "Music Structure Boundaries Estimation Using Multiple Self-Similarity Matrices as Input Depth of Convolutional Neural Networks." In: *AES International Conference Semantic Audio 2017*. Erlangen, Germany, 2017.

[32] Steven B. Davis and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." In: *ACOUSTICS, SPEECH AND SIGNAL PROCESSING, IEEE TRANSACTIONS ON* (1980), pp. 357–366.

[33] Rémi Delbouys, Romain Hennequin, Francesco Piccoli, Jimena Royo-Letelier, and Manuel Moussallam. "Music mood detection based on audio and lyrics with deep neural net." In: *arXiv preprint arXiv:1809.07276* (2018).

[34] Jean Yves Delort, Bernadette Bouchon-Meunier, and Maria Rifqi. "Enhanced Web Document Summarization Using Hyperlinks." In: *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*. HYPERTEXT '03. Nottingham, UK: ACM, 2003, pp. 208–215. ISBN: 1-58113-704-4.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805* (2018).

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805* (2018).

[37] Changshun Du and Lei Huang. "Text classification research with attention-based recurrent neural networks." In: *International Journal of Computers Communications & Control* 13.1 (2018), pp. 50–61.

[38] Hauke Egermann, Marcus T Pearce, Geraint A Wiggins, and Stephen McAdams. "Probabilistic models of expectation violation predict psychophysiological emotional responses to live concert music." In: *Cognitive, Affective, & Behavioral Neuroscience* 13.3 (2013), pp. 533–553.

[39] Günes Erkan and Dragomir R Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." In: *Journal of artificial intelligence research* 22 (2004), pp. 457–479.

[40] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 1996, pp. 226–231.

[41] Mohamed Abdel Fattah. "A hybrid machine learning model for multi-document summarization." In: *Applied intelligence* 40.4 (2014), pp. 592–600.

[42] Mohamed Abdel Fattah and Fuji Ren. "GA, MR, FFNN, PNN and GMM Based Models for Automatic Text Summarization." In: *Comput. Speech Lang.* 23.1 (2009), pp. 126–144. ISSN: 0885-2308.

[43] Michael Fell. "Lyrics classification." MA thesis. Germany: Saarland University, 2014.

[44] Michael Fell, Yaroslav Nechaev, Elena Cabrio, and Fabien Gandon. "Lyrics Segmentation: Textual Macrostructure Detection using Convolutions." In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 2044–2054.

[45] Michael Fell and Caroline Sporleder. "Lyrics-based analysis and classification of music." In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 2014, pp. 620–631.

[46] Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. "Overview of the Task on Automatic Misogyny Identification at IberEval 2018." In: *IberEval@SEPLN*. Vol. 2150. CEUR Workshop Proceedings. CEUR-WS.org, 2018, pp. 214–228.

[47] Thomas Fillon, Joséphine Simonnot, Marie-France Mifune, Stéphanie Khoury, Guillaume Pellerin, and Maxime Le Coz. "Telemeta: An open-source web framework for ethnomusicological audio archives management and automatic analysis." In: *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*. ACM. 2014, pp. 1–8.

[48] Darja Fišer, Ruihong Huang, Vinodkumar Prabhakaran, Rob Voigt, Zeerak Waseem, and Jacqueline Wernimont. "Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)." In: *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Brussels, Belgium: Association for Computational Linguistics, 2018.

[49] Jonathan Foote. "Automatic audio segmentation using a measure of audio novelty." In: *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*. Vol. 1. IEEE. 2000, pp. 452–455.

[50] Takuya Fujishima. "Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music." In: *ICMC*. Michigan Publishing, 1999.

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[52] Aria Haghighi and Lucy Vanderwende. "Exploring content models for multi-document summarization." In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2009, pp. 362–370.

[53] Ruifang He and Xingyi Duan. "Twitter Summarization Based on Social Network and Sparse Reconstruction." In: *AAAI*. 2018.

[54] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. *Spleeter: A Fast And State-of-the Art Music Source Separation Tool With Pre-trained Models*. Late-Breaking/Demo ISMIR 2019. Deezer Research. 2019.

[55] Leonhard Hennig. "Topic-based multi-document summarization with probabilistic latent semantic analysis." In: *Proceedings of the International Conference RANLP-2009*. 2009, pp. 144–149.

[56] Meishan Hu, Aixin Sun, Ee-Peng Lim, and Ee-Peng Lim. "Comments-oriented Blog Summarization by Sentence Extraction." In: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. CIKM '07. Lisbon, Portugal: ACM, 2007, pp. 901–904. ISBN: 978-1-59593-803-9.

[57] Xiao Hu and J Stephen Downie. "Improving mood classification in music digital libraries by combining lyrics and audio." In: *Proceedings of the 10th annual joint conference on Digital libraries*. 2010, pp. 159–168.

[58] Xiao Hu, J Stephen Downie, and Andreas F Ehmann. "Lyric text mining in music mood classification." In: *American music* 183.5,049 (2009), pp. 2–209.

[59] Nanzhu Jiang and Meinard Müller. "Estimating double thumbnails for music recordings." In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 146–150.

[60] Jayong Kim and Y Yi Mun. "A Hybrid Modeling Approach for an Automated Lyrics-Rating System for Adolescents." In: *European Conference on Information Retrieval*. Springer. 2019, pp. 779–786.

[61] Florian Kleedorfer, Peter Knees, and Tim Pohle. "Oh Oh Oh Whoah! Towards Automatic Topic Detection In Song Lyrics." In: *Ismir*. 2008, pp. 287–292.

[62] Kevin Knight and Daniel Marcu. "Statistics-Based Summarization - Step One: Sentence Compression." In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press, 2000, pp. 703–710. ISBN: 0-262-51112-6.

[63] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents." In: *International conference on machine learning*. 2014, pp. 1188–1196.

[64] Joseph Lee, Ziang Xie, Cindy Wang, Max Drach, Dan Jurafsky, and Andrew Y Ng. "Neural Text Style Transfer via Denoising and Reranking." In: *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. 2019, pp. 74–81.

[65] Vladimir I Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals." In: *Soviet physics doklady*. Vol. 10. 1966, pp. 707–710.

[66] Mark Levy, Mark Sandler, and Michael Casey. "Extraction of high-level musical structure from audio data and its application to thumbnail generation." In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 5. IEEE. 2006, pp. V–V.

[67] Jing Li, Aixin Sun, and Shafiq Joty. "SegBot: A Generic Neural Text Segmentation Model with Pointer Network." In: *IJCAI*. 2018, pp. 4166–4172.

[68] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries." In: *Text Summarization Branches Out*. 2004.

[69] Annie Louis and Ani Nenkova. "Automatically Assessing Machine Summary Content Without a Gold Standard." In: *Computational Linguistics* 39.2 (2013).

[70] Xuezhe Ma and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf." In: *arXiv preprint arXiv:1603.01354* (2016).

[71] Stuart Mackie, Richard McCreadie, Craig Macdonald, and Iadh Ounis. "On choosing an effective automatic evaluation metric for microblog summarisation." In: *Proceedings of the 5th Information Interaction in Context Symposium*. ACM. 2014, pp. 115–124.

[72] Jose P. G. Mahedero, Álvaro Martínez, Pedro Cano, Markus Koppenberger, and Fabien Gouyon. "Natural Language Processing of Lyrics." In: *Proceedings of the 13th Annual ACM International Conference on Multimedia*. MULTIMEDIA '05. Hilton, Singapore: ACM, 2005, pp. 475–478. ISBN: 1-59593-044-2.

[73] Qiaozhu Mei and ChengXiang Zhai. "Generating Impact-Based Summaries for Scientific Literature." In: *ACL*. 2008.

[74] Gabriel Meseguer-Brocal, Alice Cohen-Hadria, and Geoffroy Peeters. "DALI: a large Dataset of synchronized Audio, Lyrics and notes, automatically created using teacher-student machine learning paradigm." In: *ISMIR Paris, France*. 2018.

[75] Gabriel Meseguer-Brocal et al. "WASABI: a Two Million Song Database Project with Audio and Cultural Metadata plus WebAudio enhanced Client Applications." In: *Web Audio Conference 2017 – Collaborative Audio #WAC2017*. Queen Mary University of London. London, United Kingdom, 2017.

[76] Rada Mihalcea and Carlo Strapparava. "Lyrics, music, and emotions." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics. 2012, pp. 590–599.

[77] Rada Mihalcea and Paul Tarau. "TextRank: Bringing Order into Text." In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 2004.

[78] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781* (2013).

[79] Saif M. Mohammad and Peter D. Turney. "Crowdsourcing a Word-Emotion Association Lexicon." In: 29.3 (2013), pp. 436–465.

[80] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. "Semeval-2018 task 1: Affect in tweets." In: *Proceedings of the 12th international workshop on semantic evaluation*. 2018, pp. 1–17.

[81] Diego Monti, Enrico Palumbo, Giuseppe Rizzo, Pasquale Lisena, Raphaël Troncy, Michael Fell, Elena Cabrio, and Maurizio Morisio. "An Ensemble Approach of Recurrent Neural Networks using Pre-Trained Embeddings for Playlist Completion." In: *Proceedings of the ACM Recommender Systems Challenge, RecSys Challenge 2018, Vancouver, BC, Canada, October 2, 2018*. 2018, 13:1–13:6.

[82] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. "SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents." In: *AAAI*. 2016.

[83] Ani Nenkova, Kathleen McKeown, et al. "Automatic summarization." In: *Foundations and Trends® in Information Retrieval* 5.2–3 (2011), pp. 103–233.

[84] Markus Ojala and Gemma C. Garriga. "Permutation Tests for Studying Classifier Performance." In: *J. Mach. Learn. Res.* 11 (2010), pp. 1833–1863. ISSN: 1532-4435.

[85] Sergio Oramas, Luis Espinosa Anke, Mohamed Sordo, Horacio Saggion, and Xavier Serra. "ELMD: An automatically generated entity linking gold standard dataset in the music domain." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016, pp. 3312–3317.

[86] Sergio Oramas, Mohamed Sordo, Luis Espinosa-Anke, and Xavier Serra. "A semantic-based approach for artist similarity." In: *Müller M, Wiering F, editors. Proceedings of the 16th International Society for Music Information Retrieval (ISMIR) Conference; 2015 Oct 26-Oct 30; Malaga, Spain.[Sl]: International Society for Music Information Retrieval; 2015. p. 100-6.* International Society for Music Information Retrieval (ISMIR). 2015.

[87] Jahna Otterbacher, Güneş Erkan, and Dragomir R Radev. "Using random walks for question-focused sentence retrieval." In: *Proceedings of the conference on Human Language Technology and Empirical Meth-*

ods in Natural Language Processing. Association for Computational Linguistics. 2005, pp. 915–922.

[88] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[89] Loreto Parisi, Simone Francia, Silvio Olivastri, and Maria Stella Tavella. "Exploiting Synchronized Lyrics And Vocal Features For Music Emotion Detection." In: *CoRR* abs/1901.04831 (2019).

[90] Ji Ho Park and Pascale Fung. "One-step and Two-step Classification for Abusive Language Detection on Twitter." In: *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, 2017, pp. 41–45.

[91] Sungjoon Park, Jiseon Kim, Jaeyeol Jeon, Heeyoung Park, and Alice Oh. "Toward Dimensional Emotion Detection from Categorical Emotion Annotations." In: *arXiv preprint arXiv:1911.02499* (2019).

[92] Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. "Topical coherence for graph-based extractive summarization." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1949–1954.

[93] Daraksha Parveen and Michael Strube. "Integrating Importance, Non-redundancy and Coherence in Graph-based Extractive Summarization." In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 1298–1304. ISBN: 978-1-57735-738-4.

[94] J. Pauwels and M. Sandler. "A Web-Based System For Suggesting New Practice Material To Music Learners Based On Chord Content." In: *Joint Proc. 24th ACM IUI Workshops (IUI2019)*. Los Angeles, CA, USA, 2019.

[95] J. Pauwels, A. Xambó, G. Roma, M. Barthet, and G. Fazekas. "Exploring Real-time Visualisations to Support Chord Learning with a Large Music Collection." In: *Proc. 4th Web Audio Conf. (WAC 2018)*. Berlin, Germany, 2018.

[96] Samuel Pecar. "Towards Opinion Summarization of Customer Reviews." In: *Proceedings of ACL 2018, Student Research Workshop*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 1–8.

[97] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[98] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 295–313.

[99] Lawrence Philips. "The Double Metaphone Search Algorithm." In: *C/C++ Users Journal* 18 (2000), pp. 38–43.

[100] Robert Plutchik and Henry Kellerman. *Emotion, theory, research, and experience*. Academic press, 1980.

[101] James A Russell. "A circumplex model of affect." In: *Journal of personality and social psychology* 39.6 (1980), p. 1161.

[102] Horacio Saggion and Thierry Poibeau. "Automatic Text Summarization: Past, Present and Future." In: Springer, Berlin, Heidelberg, 2013, pp. 3–21.

[103] Markus Schedl, Arthur Flexer, and Julián Urbano. "The neglected user in music information retrieval research." In: *Journal of Intelligent Information Systems* 41.3 (2013), pp. 523–539. ISSN: 1573-7675.

[104] Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. "Summarization Evaluation in the Absence of Human Model Summaries Using the Compositionality of Word Embeddings." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 905–914.

[105] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. "Early Versus Late Fusion in Semantic Video Analysis." In: *Proceedings of the 13th Annual ACM International Conference on Multimedia*. MULTIMEDIA '05. Hilton, Singapore: ACM, 2005, pp. 399–402. ISBN: 1-59593-044-2.

[106] Jacquelin A Speck, Erik M Schmidt, Brandon G Morton, and Youngmoo E Kim. "A Comparative Study of Collaborative vs. Traditional Musical Mood Annotation." In: *ISMIR*. Vol. 104. Citeseer. 2011, pp. 549–554.

[107] Jacopo Staiano and Marco Guerini. "Depeche Mood: a Lexicon for Emotion Analysis from Crowd Annotated News." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 427–433.

[108] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. "Open-unmix-a reference implementation for music source separation." In: *Journal of Open Source Software* (2019).

[109] Philip Tagg. "Analysing popular music: theory, method and practice." In: *Popular Music* 2 (1982), pp. 37–67.

[110] Laurent Vanni, Mélanie Ducoffe, Carlos Aguilar, Frederic Precioso, and Damon Mayaffre. "Textual Deconvolution Saliency (TDS): a deep tool box for linguistic analysis." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 548–557.

[111] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need." In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008.

[112] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

[113] Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. "A sentence compression based framework to query-focused multi-document summarization." In: *arXiv preprint arXiv:1606.07548* (2016).

[114] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. "Norms of valence, arousal, and dominance for 13,915 English lemmas." In: *Behavior research methods* 45.4 (2013), pp. 1191–1207.

[115] Kento Watanabe, Yuichiroh Matsubayashi, Naho Orita, Naoaki Okazaki, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, Jordan Smith, and Masataka Goto. "Modeling Discourse Segments in Lyrics Using Repeated Patterns." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, pp. 1959–1969.

[116] Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. "Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language." In: *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*. 2018.

[117] Thomas Wolf et al. "HuggingFace's Transformers: State-of-the-art Natural Language Processing." In: *ArXiv* abs/1910.03771 (2019).

[118] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. "Hierarchical Attention Networks for Document Classification." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, 2016, pp. 1480–1489.

[119] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. "SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval)." In: *CoRR* abs/1903.08983 (2019).