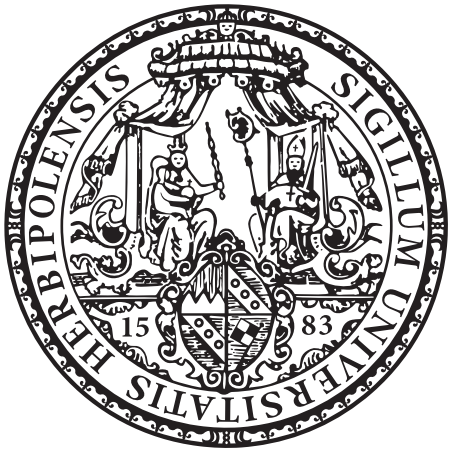


Finding a Correlation Between Sentiment and Audio of Movie Scenes

Bachelorarbeit
im Bachelorstudiengang Informatik am Institut für Informatik, vorgelegt am 29.03.2018
von

Armin Bernstetter
Matrikelnummer 1876874

Gutachter: Prof. Dr. Andreas Hotho
Universität Würzburg

Betreuer: Daniel Schlör
Universität Würzburg

Finding a Correlation Between Sentiment and Audio of Movie Scenes

Erklärung

Ich versichere hiermit, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Würzburg, den 29.03.2018

Armin Bernstetter

Abstract

Movies in their first stage of production are a text based medium. Authors write screenplays that contain information about scenes, action, characters, those characters' dialogue and emotions, and much more. Using text mining methods like sentiment analysis, information can be extracted from these movie scripts about which scenes express sentiment and certain emotions.

On the other hand, finished movies are an audiovisual medium and thus the audio track also contains information about the suspense in movie scenes. One prominent example here are so called jump scares in horror movies, often characterized in part by a large increase in volume after a period of silence.

The hypothesis is that a correlation between the sentiment and emotions of a given scene and the audio of this scene exists. Therefore, particularly emotional or thrilling scenes like action scenes, jump scares or other horror scenes can be identified in movie scripts. These scenes also reflect in the audio track because of certain characteristics like spikes or a continual rise to a crescendo in the volume.

A data set is created consisting of movie scripts, subtitles and movies. The hypothesis is examined by applying sentiment analysis methods to movie scripts and analyzing the audio track of movies. The collected data is then combined to find a correlation between the sentiment and audio of movie scenes.

Contents

List of Figures iii

List of Tables iv

Listings v

1 Introduction 1
- 1.1 Motivation and Goals . . . 1
- 1.2 Structure . . . 2

2 Related Work 4
- 2.1 Sentiment Analysis on fictional Texts . . . 4
- 2.2 Analysis of Affect in audiovisual Content through Audio Features . . . 5

3 Data Aquisition and Preprocessing 7
- 3.1 Text Data . . . 7
  - 3.1.1 Movie Scripts . . . 7
  - 3.1.2 Subtitles . . . 9
  - 3.1.3 Annotating Time Codes to Movie Scripts . . . 9
- 3.2 Audio . . . 14

4 Sentiment- and Audio Analysis 15
- 4.1 Sentiment Analysis . . . 15
  - 4.1.1 Basic Concepts and utilized Sentiment Analysis Tools . . . 15
  - 4.1.2 Implementation and Application . . . 17
- 4.2 Audio Analysis . . . 21
  - 4.2.1 Audio Features . . . 21
  - 4.2.2 Feature Extraction using a Horror Short Movie as Example . . . 22

5 Combining Sentiment and Audio Data 25
- 5.1 A preliminary Note on other Audio Features . . . 25
- 5.2 Finding a Correlation between Sentiment and Audio Energy . . . 26
  - 5.2.1 Plain Text "Fountain" Movie Scripts . . . 27
  - 5.2.2 Scenes without Time Codes . . . 28

5.2.3 Subtitles with Time Codes 28
5.2.4 Scenes with Time Codes 29

6 Discussion and Future Work **32**
6.1 Discussion 32
6.2 Future Work 33

7 Conclusion **36**

Bibliography **37**

Appendices **40**

List of Figures

1.1 The overall project structure of this thesis with references to the source code. 3

2.1 The four Audio Energy Events proposed by Moncrieff et. al [18]. 5

4.1 Word Clouds for 897 movie scripts. Split into words with high, medium and low arousal with each "level" being 1/3 of the arousal score range (1-9). 18
4.2 Word Clouds for 897 movie scripts. Split into words with high, medium and low valence with each "level" being 1/3 of the valence score range (1-9). 19
4.3 Word Clouds for 897 movie scripts. Split into words with high, medium and low dominance with each "level" being 1/3 of the dominance score range (1-9). 20
4.4 Audio energy for the horror short movie "Selfie From Hell". The graph shows spikes in energy similar to the audio energy event type 1 (surprise, alarm, here: jump scare) introduced in figure 2.1. 23
4.5 Spectral centroids and MFCC Spectrogram of "Selfie From Hell" 23

5.1 Comparison between audio energy and the first Mel-Frequency Cepstral Coefficient for Star Wars Episode IV. 26
5.2 Distribution of sentiment and audio energy of scenes from 7 movies 27

6.1 Regression Plots of a Linear Regression Model for 3 Sentiment Dimensions and Audio Energy of Movie Scenes. 35

List of Tables

3.1 Example of the iteration done by the annotation algorithm. 12
3.2 Comparison between three approaches for automatically annotating movie scripts. 14

5.1 Correlation Coefficients for movie script sentiment and audio energy for seven movies partitioned each into 200 sections of equal length) 28
5.2 Correlation Coefficients for movie script sentiment and audio energy for 1482 scenes from seven movies without annotated time codes 28
5.3 Correlation Coefficients for subtitle sentiment and audio energy for 5619 subtitles from seven movies 29
5.4 Correlation Coefficients for Warriner sentiment and audio energy for 1160 scenes with time codes from seven movies 29
5.5 Correlation coefficients for Vader sentiment and audio energy for 1160 scenes with time codes from seven movies 30
5.6 Correlation Coefficients for Warriner sentiment and audio energy for 136 scenes from the movie “Indiana Jones and the Last Crusade” 31

6.1 Example of the sentiment scores computed for a movie genre classification task. 34

Listings

3.1 Example of the Fountain movie script format from the first scene in the script of the movie Star Wars Episode IV - A New Hope 8
3.2 Example of the parsed XML-movie script format. From the first scene in the movie “Star Wars Episode IV - A New Hope” 10
3.3 Example of “srt” subtitles from the movie “Star Wars Episode IV - A New Hope” 10
3.4 XML subtitle example from the movie “Star Wars Episode IV - A New Hope” 11
3.5 Example of an annotated XML-movie script. Combines the movie script seen in listing 3.2 and the subtitles seen in listing 3.3 and 3.4. 13

1 | Introduction

Before a movie is shown on cinema screens all over the world it is passed through various production steps. It often takes years from an initial idea through a sales pitch, various drafts of the screenplay, casting, actual shooting and editing to a finished audiovisual medium. This work concentrates on screenplays or "movie scripts" on the hand and the audio track of the finished movies on the other hand.

1.1 Motivation and Goals

Bing Liu, one of the forerunners of sentiment analysis and opinion mining, defines sentiment analysis as the "field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes." [12] Sentiment analysis has often been applied to texts such as product reviews, movie reviews or social media posts and to a lesser extent to fictional texts like novels. Some of this work is introduced in chapter 2. Movie scripts, however, have not been used as much despite providing valuable information for text mining. Screenplays make it easy to extract specific information due to their clear structure. They are separated into scenes and contain information about locations, dialogue and speaking characters, as well as meta text such as descriptions of surroundings and action. Thus they provide easily accessible data for tasks such as tracking change in dialogue of the protagonists through the course of the narrative or emotions invoked by scenes. Denis et al. [6] show this in a paper on emotion detection and visualization of affect in movie scripts.
But why use movies for sentiment analysis in the first place? Finding information about emotions in movie scenes might be valuable for tasks involving viewer experience. One example here are recommendation systems for movies in online streaming platforms such as Netflix or Amazon. Often such systems are based on genre and other attributes of previously watched movies. But what if viewers could choose what to watch based on emotions and sentiment connected to the movies? For example whether a movie is generally more positive and light hearted or contains a high amount of negative, thrilling scenes? Without information about the plot such a system would have to be based on the descriptions and opinions of other users. Another application could be automatic warnings for parents whether movies are age-appropriate for their children.
Sentiment analysis on movie scripts could provide a basis for such a system.

Since finished movies are an audiovisual medium, being restricted to text based data such as movie scripts might provide information that is not very useful on its own. Complementing this with audiovisual features might improve the results. Literature has shown success in tasks such as violence detection using audio [23] as well as finding information relating to the narrative contents of movies [17].

This lead to the idea of combining sentiment analysis and audio analysis for this work. This bachelor thesis has multiple intermediate goals. Initially, a data set of easily machine-readable movie scripts has to be prepared. These scripts have to make information about scenes, meta text, dialogue, characters and genres easily extractable.
Frameworks have to be developed to access this information on multiple levels as well as extracting information from the audio track of movies.
The data is then used for sentiment- and audio analysis with the goal of finding a correlation between audio and sentiment.

1.2 Structure

The structure of this thesis closely follows the aforementioned goals. Chapter 2 introduces related work. These papers research sentiment in fictional literature as well as connections of audio features and affect. Chapter 3 describes the steps taken in preprocessing the data set used in this thesis. Chapter 4 introduces basics of sentiment- and audio analysis, and shows how these are applied to the data set. Chapter 5 describes how the results from sentiment- and audio analysis are combined to find information about correlations between sentiment and audio.
Finally chapter 6 discusses the results achieved in this thesis and introduces possible future work based on this bachelor thesis's topic. Chapter 7 concludes the thesis.

Figure 1.1 shows the processing steps followed in this work. A more detailed overview over the Python source code developed during the course of this thesis is shown in Appendix A.

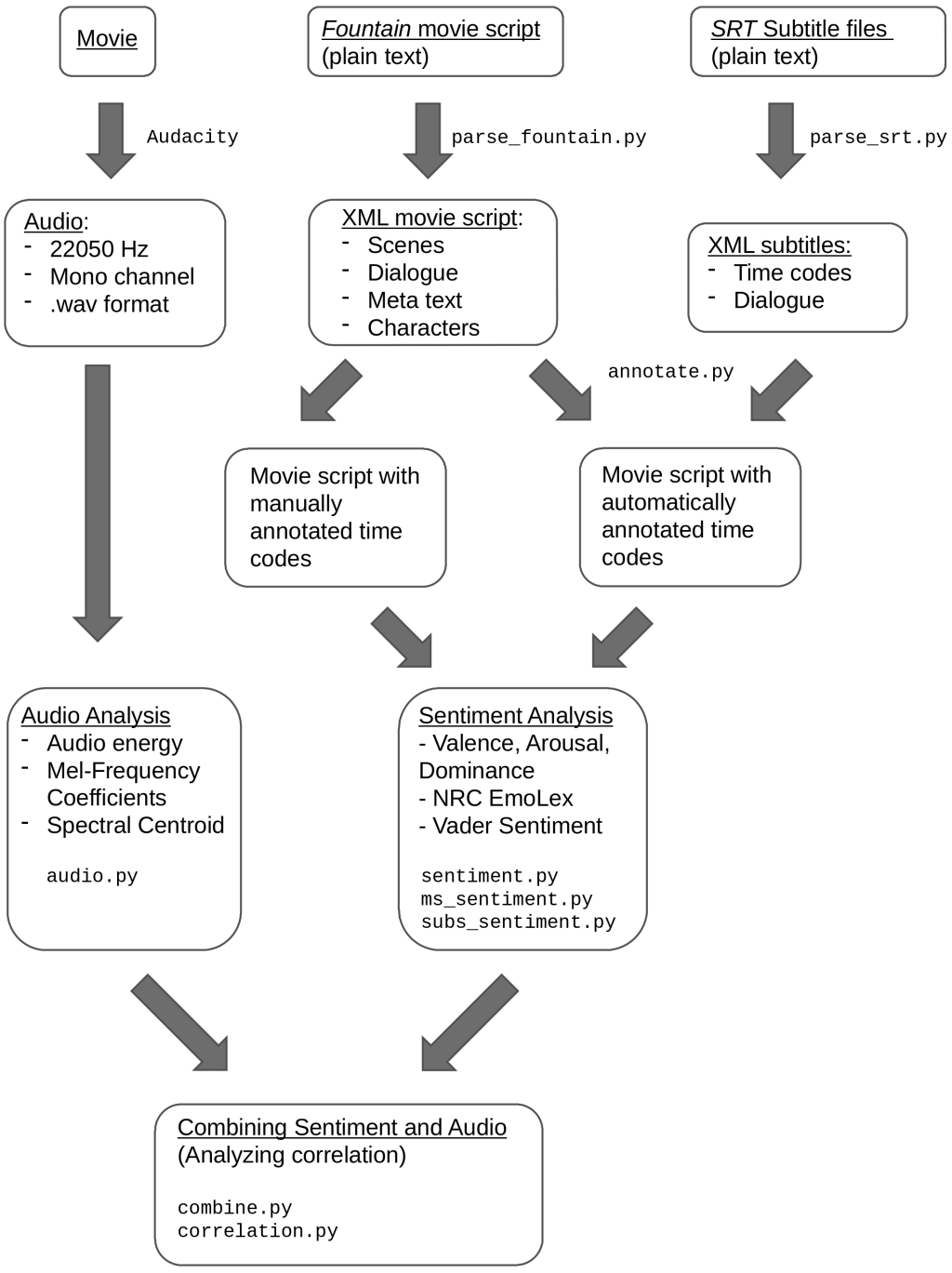


Figure 1.1: The overall project structure of this thesis with references to the source code.

2 | Related Work

On the one hand the related work introduced in this chapter uses sentiment analysis on fictional texts. That includes tasks such as happy end detection or general emotion detection in novels and fairy tales. Section 2.2 on the other hand introduces literature on analyzing audiovisual content such as movies. These papers use various audio features to identify whether movies influence affect or certain emotions of viewers.

2.1 Sentiment Analysis on fictional Texts

Saif Mohammad and Peter Turney created a large sentiment lexicon[1] by crowdsourcing, labeling words as positive or negative as well as their relation to certain emotions [15]. Apart from *positive* and *negative*, the lexicon associates words with the six basic emotions *anger*, *disgust*, *fear*, *happiness*, *sadness* and *surprise* researched by Paul Ekman [7]. Mohammad and Turney extend these emotions with *trust* and *anticipation*. Each word in the lexicon has a 0 or 1 score for each sentiment dimension. Mohammad then uses this lexicon to track emotions and affect in mail and books [16]. He analyzes different kinds of letters like love letters, hate mail and suicide notes as well as novels and fairy tales. One of the approaches in Mohammad's paper is a visualization of emotion over the timeline of a story.
The idea of showing the change of sentiment and emotions in a fictional text has been taken up by other researchers. In a series of blog posts, Matthew Jockers introduces the R package *Syuzhet* and his approach of visualizing the plot progression of novels [11]. Although the general idea was received positively, he had to face criticism for his choice of visualization and smoothing the curves [25].
Zehe et al. use the german version of the NRC lexicon to predict happy endings in german novels using machine learning for classification resulting in an F1-score of 0.73 [33].
A paper by Alexandre Denis et al. doesn't use novels but rather screenplays and approaches the visualization of affect in movie scripts [6]. Similar to the thesis at hand their dataset is taken from http://www.imsdb.com/. They use the "SATI" API[2] *(Sentiment Analysis from Textual Information)* to analyze the scripts.

[1] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
[2] http://talc2.loria.fr/empathic/

2.2 Analysis of Affect in audiovisual Content through Audio Features

Early research in this area was done by Silvia Pfeiffer et al. [23] analyzing audio features such as amplitude, frequency and pitch to implement violence detection in movies. For this they first structure the audio and distinguish music, speech, silence etc and then compare short windows to the signatures of violent sounds. This includes for example gun shots or explosions. Nam et al. [20] implement a violent scene identification system. They combine audio, i.e. sound effects of violent events, with visual features including detection of bloody pixels on screen.

An important aspect in the initial conception of this thesis were the works of Simon Moncrieff et al. and multiple papers based on their introduction of four so called *Sound- or Audio Energy Events* [18]. These events are described in figure 2.1. Moncrieff et al. state that they are used in the story telling of a movie as they correspond to specific affect and certain symbolic meanings in a movie.

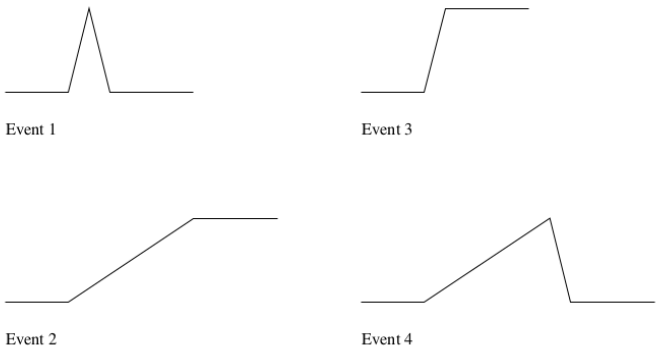


Figure 2.1: The four Audio Energy Events proposed by Moncrieff et. al [18].

Event 1: Surprise or alarm. Event 1 is characterized by a large increase in sound energy after a period of low energy with a brief attack time. A prominent example for this event are so called jump scares in horror movies.

Event 2: Apprehension. Event 2 consists of an increase in sound energy over a period of time.

Event 3: Surprise followed by sustained alarm. This event is characterized by a large increase in sound energy after a period of low energy, similar to event 1. Event 3 however keeps the high energy after the initial attack.

Event 4: Building apprehension up to a climax. Event 4 shows an increase of sound energy over a period of time and an attack followed by a decay period

Moncrieff et al. use these energy events to classify movies as belonging to the horror genre by analyzing the amount of events found in different genres [19]. In another paper

they analyze the overall narrative structure by using audio data [17]. Their hypothesis is that changes in the narrative structure are reflected by a change in the audio pace. This audio pace is introduced as a more generally applicable semantic examination of film audio [17]. The audio pace consists of multiple features, one of which is the audio energy, characterized by the average audio amplitude. They conclude that changes in the narration will coincide with changes in the audio content.

Xu et al. use audio sounds which have strong hints to affective content like laughing or horror sounds to detect affect in comedy and horror videos [31]. The audio features they use are the audio energy again as well as Mel-Frequency Cepstral Coefficients (MFCC). In another paper they research a connection between audio features, and valence and arousal [32]. They state that the audio features Short-Time Energy and MFCC are related to arousal whereas pitch is used for valence. The audio features mentioned here are introduced in greater detail in Section 4.2.1.
Overall these papers show that a connection between emotion or affect and certain audio features exists.

3 | Data Aquisition and Preprocessing

The data set used in this thesis consist of movie scripts, subtitles and the audio track of entire movies. In Appendix B, a more detailed overview of the data set is given.

3.1 Text Data

Since movie scripts are only an early step in the preproduction of a movie, they bring certain challenges with them. One of those is that there often are no official sources for the finished script. This means a restriction to online sources which sometimes have to be taken with a grain of salt as most of the content is uploaded by users. Often the available movie scripts are no 1:1 transcript of the finished movie but rather early drafts, sometimes dating multiple years before the movie even began filming. Common problems are scenes that don't make it into the final cut of a movie, dialogue or locations that get changed or scenes being in a different order than in the screenplay. Subtitles are less prone to be problematic since they are often directly taken from movie files or media like DVDs.

The general task at hand is to display sentiment and emotion on a time axis to have a way of mapping sections of movie scripts and their sentiment to the audio. Since movie scripts do not contain time codes and it can not be assumed that all scenes have equal duration, they have to be annotated with time codes. One approach for this is to use subtitles which do contain time codes to automatically annotate the screenplays. Another approach is manual annotation. This annotation process is described in section 3.1.3.

3.1.1 Movie Scripts

There are multiple online sources for movie scripts, but the "Internet Movie Script Database" (IMSDB)[1] provides easy access to over 1100 movie scripts and has been used successfully in related work (see section 2.1). 1121 movie scripts were downloaded on 18 October 2017 as plain text files from the html source of IMSDB using a python script up-

[1] www.imsdb.com

```
INT. REBEL BLOCKADE RUNNER - MAIN PASSAGEWAY

An explosion rocks the ship as two robots, Artoo-Detoo (R2-
D2) and See-Threepio (C-3PO) struggle to make their way
through the shaking, bouncing passageway. Both robots are
old and battered. Artoo is a short, claw-armed tripod. His
face is a mass of computer lights surrounding a radar eye.
Threepio, on the other hand, is a tall, slender robot of
human proportions. He has a gleaming bronze-like metallic
surface of an Art Deco design.

Another blast shakes them as they struggle along their way.

          THREEPIO
     Did you hear that? They've shut down
     the main reactor. We'll be destroyed
     for sure. This is madness!

Rebel troopers rush past the robots and take up positions in
the main passageway. They aim their weapons toward the door.

          THREEPIO
     We're doomed!
```

Listing 3.1: Example of the Fountain movie script format from the first scene in the script of the movie Star Wars Episode IV - A New Hope

loaded to github by Jeremy Kun[2]. After sighting the data and removing flawed scripts as well as cleaning some, around 900 scripts remained. The removed scripts often were not parsed correctly from html, were too uniquely formatted in comparison to the majority of the other scripts or were not easily machine-readable and parsable. Those remaining were generally similar to the format of *Fountain*[3], a markup language for screenwriting.

An example of this format can be seen in listing 3.1. Each scene is captioned by a header in upper case beginning in most cases with either `INT.` or `EXT.` declaring whether the location of the scene is inside (interior) or outside (exterior). The rest of the header describes or simply names the location. Text inside a scene is either meta text describing action of characters and other things happening, or dialogue. Dialogue is indented and prefaced by the name of the speaking character in upper case and further indented. The dialogue text follows in the lines below the character name and is ended by an empty line.

To provide better machine-readable text, these movie scripts were parsed to an XML format as seen below in listing 3.2. Unfortunately there often are exceptions which make automatic parsing of these movie scripts more difficult and with a data set of nearly 900 scripts it is not possible to prevent all errors. Manual inspection confirmed, however, that in many cases the parsing works reliably and provides data which can be used for text mining tasks. A fortunate side effect of the initial download from the html-source of IMSDB is that each script contains information about title, writers and genres shown

[2]https://github.com/j2kun/imsdb_download_all_scripts
[3]https://fountain.io/

in the `info` element in listing 3.2. The text is separated into scenes with subelements for the scene header, meta text and dialogue. The latter two contain the text tokenized into sentences with the dialogue elements containing the talking character as additional information. Each element in the XML-tree has a unique ID.

3.1.2 Subtitles

A total of 1610 subtitle files were downloaded from the *Opus* corpus[4] managed by Jörg Tiedemann [27, 28] for the *Nordic Language Processing Laboratory*, a collaboration of academic research groups in northern Europe[5]. This corpus was parsed originally from subtitles in the plain text *SubRip* subtitle format *srt* provided by the web site *opensubtitles*[6] [22]. *SubRip* is a program for extracting subtitles and their time codes from movies [34].

The format of these srt-subtitles is seen in listing 3.3. Each subtitle consists of three elements. The first line shows a continuous index. The second line shows the time at which the text appears on screen on the left-hand side of the arrow and the time when it disappears. The next lines are the actual text, followed by an empty line which separates the subtitles.

The XML-format used in the NLPL corpus is shown in listing 3.4. The text is tokenized into words and each element in the XML-tree has a unique ID.

3.1.3 Annotating Time Codes to Movie Scripts

To be able to compare sentiment to audio, a mapping of sections of text to a time axis is required. Subtitles already provide this but contain only dialogue which is a small part of the text available in movie scripts. Using only subtitles would mean dismissing potentially important information and sentiment in meta text such as descriptions of characters' actions. Therefore the movie scripts have to be annotated with time codes. One approach to achieve this is to combine movie scripts and subtitles. In total, the dataset contains 220 movies for which movie script as well as subtitles are available. On average, each of those movies contains 153 scenes (median: 145). For each of these movies, both files are used to automatically find dialogue sentences that appear in the movie script as well as the subtitles. When the position of those sentences is determined, time codes from the subtitles can be added to the movie script.

The task of automatically annotating movie scripts with time codes provides interesting challenges. Some of the problems at hand are that subtitles are taken from the finished movies whereas movie scripts can be early drafts. This means dialogue might have changed or the order of scenes in the finished movie might be different from the movie script. Another problem is that sentences might occur multiple times, especially short ones such as *"Hey, what's up?"*. This leads to false positives i.e. sentences being detected by the algorithm but at the wrong positions.

[4] http://opus.nlpl.eu/OpenSubtitles.php [Accessed 22 March 2018]
[5] http://wiki.nlpl.eu/index.php/Home [Accessed 22 March 2018]
[6] https://www.opensubtitles.org/ [Accessed 22 March 2018]

```
<movie>
  <info>
    <title>Star Wars: A New Hope</title>
    <writers>George Lucas</writers>
    <genres>Action, Adventure, Fantasy, Sci-Fi</genres>
  </info>
  <scene id="sc1">
    <sceneheader>INT. REBEL BLOCKADE RUNNER - MAIN PASSAGEWAY</sceneheader>
    <meta id="sc1m1">
      <s id="sc1m1s1">An explosion rocks the ship as two robots, Artoo-Detoo (R2-
          D2) and See-Threepio (C-3PO) struggle to make their way through the
          shaking, bouncing passageway.</s>
      <s id="sc1m1s2">Both robots are old and battered.</s>
      <s id="sc1m1s3">Artoo is a short, claw-armed tripod.</s>
      <s id="sc1m1s4">His face is a mass of computer lights surrounding a radar
          eye.</s>
      <s id="sc1m1s5">Threepio, on the other hand, is a tall, slender robot of
          human proportions.</s>
      <s id="sc1m1s6">He has a gleaming bronze-like metallic surface of an Art
          Deco design.</s>
      <s id="sc1m1s7">Another blast shakes them as they struggle along their way
          .</s>
    </meta>
    <dialogue id="sc1d1" name="THREEPIO">
      <s id="sc1d1s1">Did you hear that?</s>
      <s id="sc1d1s2">They've shut down the main reactor.</s>
      <s id="sc1d1s3">We'll be destroyed for sure.</s>
      <s id="sc1d1s4">This is madness!</s>
    </dialogue>
    <meta id="sc1m2">
      <s id="sc1m2s1">Rebel troopers rush past the robots and take up positions in
          the main passageway.</s>
      <s id="sc1m2s2">They aim their weapons toward the door.</s>
    </meta>
    <dialogue id="sc1d2" name="THREEPIO">
      <s id="sc1d2s1">We're doomed!</s>
    </dialogue>
    ...
  </scene>
  ...
</movie>
```

Listing 3.2: Example of the parsed XML-movie script format. From the first scene in the movie "Star Wars Episode IV - A New Hope"

```
1
00:02:40,680 --> 00:02:42,557
Did you hear that?

2
00:02:42,680 --> 00:02:45,319
They shut down the main reactor.
We'll be destroyed for sure.
```

Listing 3.3: Example of "srt" subtitles from the movie "Star Wars Episode IV - A New Hope"

```
<s id="1">
  <time id="T1S" value="00:02:40,680"/>
  <w id="1.1">Did</w>
  <w id="1.2">you</w>
  <w id="1.3">hear</w>
  <w id="1.4">that</w>
  <w id="1.5">?</w>
  <time id="T1E" value="00:02:42,557"/>
</s>
<s id="2">
  <time id="T2S" value="00:02:42,680"/>
  <w id="2.1">They</w>
  <w id="2.2">shut</w>
  <w id="2.3">down</w>
  <w id="2.4">the</w>
  <w id="2.5">main</w>
  <w id="2.6">reactor</w>
  <w id="2.7">.</w>
  <time id="T2E" value="00:02:45,319"/>
</s>
<s id="3">
  <time id="T3S" value="00:02:42,680"/>
  <w id="3.1">We</w>
  <w id="3.2">'ll</w>
  <w id="3.3">be</w>
  <w id="3.4">destroyed</w>
  <w id="3.5">for</w>
  <w id="3.6">sure</w>
  <w id="3.7">.</w>
  <time id="T3E" value="00:02:45,319"/>
</s>
```

Listing 3.4: XML subtitle example from the movie “Star Wars Episode IV - A New Hope”

The implementation of the annotation program uses the python library *fuzzywuzzy*[7] that matches strings using the *Levenshtein* distance for string similarity. The annotation algorithm iterates through the dialogue from the movie script and the subtitles and compares sentences.

Three approaches to reduce the false positives are were tried. They have in common that they narrow down the sections from which sentences are taken and then compared.

The first approach takes the difference δ between the total number of dialogue sentences in the movie script and sentences in the subtitles.

$$\delta = |\ |\{dialogue\ sentences\ in\ movie\ script\}| - |\{sentences\ in\ subtitles\}|\ |$$

Only sentences with indices in the range of $index \pm \delta$ are compared. This δ is vastly different depending on the movie which might be a result of movie scripts containing scenes that were not included in the final movie among other reasons. For this approach a similarity threshold of 80% was chosen. This means the ratio of similarity between two strings computed by *fuzzywuzzy* has to be 80% or higher. If this condition is true, the time code from the subtitles will be added to the corresponding sentence in the movie script.

The second approach also takes a threshold of 80% but uses a relative matching range of ±5%. For each sentence from both movie script and subtitles, its relative position is computed with

$$\frac{index}{total\ number\ of\ sentences}$$

Then, only those sentences from the movie script are considered where their position is within ±5% of the subtitle's position. Simply put, if a subtitle is within the first 10th of its file it's only compared to movie script dialogue that's also in the first 10th of its file.

For the third approach a similarity threshold of 90% was used with matching range being ±10%.

The iteration and comparison between sentences from subtitles and movie scripts is illustrated in table 3.1.

$index_{sub}$	Subtitle sentence	$index_{movie}$	Movie script sentence	Similarity
0	Did you hear that?	0	**Did you hear that?**	**100%**
		1	They've shut down the main reactor.	34%
		2	We'll be destroyed for sure.	22%
		3	This is madness!	24%
		...	...	...
1	They shut down the main reactor.	0	Did you hear that?	36%
		1	**They've shut down the main reactor.**	**96%**
		2	We'll be destroyed for sure.	37%
		3	This is madness!	38%
		...	...	...
...	...	...	...	...

Table 3.1: Example of the iteration done by the annotation algorithm.

[7] https://github.com/seatgeek/fuzzywuzzy [Accessed 22 March 2018]

```
<scene id="sc1" time_avg="00:02:50">
  <sceneheader>INT. REBEL BLOCKADE RUNNER - MAIN PASSAGEWAY</sceneheader>
  <meta id="sc1m1">
    <s id="sc1m1s1">An explosion rocks the ship as two robots, Artoo-Detoo (R2- D2
        ) and See-Threepio (C-3PO) struggle to make their way through the shaking,
         bouncing passageway.</s>
    <s id="sc1m1s2">Both robots are old and battered.</s>
    <s id="sc1m1s3">Artoo is a short, claw-armed tripod.</s>
    <s id="sc1m1s4">His face is a mass of computer lights surrounding a radar eye
        .</s>
    <s id="sc1m1s5">Threepio, on the other hand, is a tall, slender robot of human
         proportions.</s>
    <s id="sc1m1s6">He has a gleaming bronze-like metallic surface of an Art Deco
        design.</s>
    <s id="sc1m1s7">Another blast shakes them as they struggle along their way.</s
        >
  </meta>
  <dialogue id="sc1d1" name="THREEPIO">
    <s id="sc1d1s1" subtitle_id="1" time="00:02:40,680">Did you hear that?</s>
    <s id="sc1d1s2" subtitle_id="2" time="00:02:42,680">They've shut down the main
         reactor.</s>
    <s id="sc1d1s3" subtitle_id="3" time="00:02:42,680">We'll be destroyed for
        sure.</s>
    <s id="sc1d1s4" subtitle_id="4" time="00:02:45,440">This is madness!</s>
  </dialogue>
  <meta id="sc1m2">
    <s id="sc1m2s1">Rebel troopers rush past the robots and take up positions in
        the main passageway.</s>
    <s id="sc1m2s2">They aim their weapons toward the door.</s>
  </meta>
  <dialogue id="sc1d2" name="THREEPIO">
    <s id="sc1d2s1" subtitle_id="5" time="00:02:56,560">We're doomed!</s>
  </dialogue>
    ...
</scene>
```

Listing 3.5: Example of an annotated XML-movie script. Combines the movie script seen in listing 3.2 and the subtitles seen in listing 3.3 and 3.4.

Listing 3.5 again shows the scene from listing 3.2, this time with annotated time codes.

Table 3.2 shows a comparison between the three approaches. For all three variants, the total number of movie scripts where the annotated time codes are not in order is very high. The ratio of annotated scenes to total scenes is $< \frac{1}{3}$ for all three approaches.

As a side note the movie *Forrest Gump* annotated using *Approach A* has the highest ratio of annotated scenes with 162 scenes with time code to 194 in total.

n = 220 movies, avg. 153 scenes each	**Approach A**	**Approach B**	**Approach C**
Matching Range	$\delta(movie\ script, subtitles)$	±5%	±10%
Similarity Threshold	80%	80%	90%
Avg percentage of annotated scenes	30%	27%	26%
# of movies where time codes not in order	203	180	196

Table 3.2: Comparison between three approaches for automatically annotating movie scripts.

Another problem is that an approximate duration of a scene can only be computed if multiple sentences inside a scene are found. This is often not the case and some scenes do not contain dialogue at all. To mitigate this, it was tried to interpolate the time codes of scenes where no subtitles are found. Despite the approach of matching only sentences from corresponding sections as described above, this interpolation amplifies the problem of time codes not being in order.

Automatically annotating movie scripts with time codes from subtitles is an interesting approach. Unfortunately it still requires some fine-tuning to be able to provide reliable data. Therefore it was necessary to resort to manually annotating time codes of scenes to movie scripts. This approach only allows annotating a small number of movies. However, it provides an exact mapping of scenes to time codes and enables a correct annotation of start and end time of a scene and therefore an exact duration.

3.2 Audio

In total, the data set contains audio files from 61 movies with an average duration of 119 minutes. The audio of these movies was extracted using the program *Audacity*[8] and downsampled to a mono channel WAV format with samplerate 22050 Hz. This is the default sample rate used by the python library librosa[9] which is used for audio analysis in this thesis.

[8] https://www.audacityteam.org/ [Accessed 22 March 2018]
[9] https://librosa.github.io/ [Accessed 22 March 2018]

4 | Sentiment- and Audio Analysis

This chapter describes the steps taken to analyze text and audio individually. The goal was to extract data that can then be used for finding correlations or possibly for machine learning. This was done by trying and comparing multiple approaches, some of which had to be discarded due to leading into dead ends or exceeding the scope of this work.

4.1 Sentiment Analysis

This section section introduces some basic concepts and sentiment analysis methods. The sentiment analysis done in this thesis is not an approach of developing a new comprehensive method as this was not the main topic.
This work uses either fully established, complete sentiment analysis tools or simply uses sentiment lexicons on the word level. The former mostly provide a full interface to return sentiment scores when given a text of arbitrary length as input. The latter usually come in the form of a word list with sentiment information about each word.
The choice of sentiment analysis method can be challenging due to the sheer number of approaches and depends on the field of research. Papers comparing and benchmarking various tools and lexicons can be of assistance for this task. One example is the project *SentiBench* by Filipe Nunes Ribeiro et al. [24].

4.1.1 Basic Concepts and utilized Sentiment Analysis Tools

Sentiment analysis is often applied with the goal of detecting how people think about certain products. Big brands and companies are interested in automatically analyzing reviews for products, movies etc to find out what exactly consumers like or criticize.
With the rise in importance of social media, posts by users on Twitter, Facebook, Reddit and other sites have also become important data to use with opinion mining. Some sentiment analysis libraries and lexicons even include various emoticons. This information from social media posts is potentially very useful for marketing, public relations of companies but also political campaigns.

The most basic dimension of analyzing sentiment in text is a simple positive/negative scale. The research question for such a task would be for example “Do people think positively or negatively of a topic?”. This approach to sentiment analysis has seen a lot

of research and generated multiple tools and lexicons. Some of those have initially been considered for this work but were eventually discarded due to redundancy.
SentiWordNet, a sentiment lexicon developed by Andrea Esuli, Fabrizio Sebastiani and Stefano Baccianella provides an extensive word list based on WordNet[1] [8, 3]. Apart from scores for positivity, negativity and objectivity it contains detailed information for in-depth natural language processing. Using information from WordNet, each of the over 100000 words in the SentiWordNet lexicon is assigned its part-of-speech tag, a set of synonyms and a description. This is useful for extended text mining and natural language processing but far exceeds the scope of this work.
One of the tools that has been chosen for the sentiment analysis "tool belt" for this work is VADER, the *Valence Aware Dictionary and sEntiment Reasoner* [9]. It is a lexicon and rule-based tool for sentiment analysis that has been incorporated into the Python *Natural Language Toolkit* (NLTK). VADER outputs scores for positivity, negativity, neutrality and a compound score for a given text. It was developed for analyzing sentiments expressed in social media. VADER was incorporated in this thesis to test how well sentiment analysis tools that have been developed for social media work in the context of fictional texts.

In this work, the focus is not on analyzing user generated subjective text but rather fictional text, specifically movie scripts.
This demands a more refined approach than simply determining whether a portion of text expresses positive or negative sentiment. Especially when the goal is to eventually merge the results with information from audio, a very complex medium. Simply determining whether a scene is positive or negative does not inherently provide information about whether the scene is also thrilling or exciting. A simple fictional example could be a quiet, romantic scene in contrast to a scene showing a funeral. Sentiment expressed by both scenes might be vastly different with the romantic scene containing positive words like *love* or *happiness* and the funeral containing negative words like *sadness* or *death*. Readers or viewers, however, might experience both scenes as not very thrilling in the sense of being "on the edge of their seats".

Mohammad and Turney's *NRC Emotion Lexicon*[2] extends a simple positive-negative scale with eight emotions [16]. The lexicon contains 14182 words with a score of 1 or 0 for each of the ten dimensions (*Anger*, *Anticipation*, *Disgust*, *Fear*, *Joy*, *Negative*, *Positive*, *Sadness*, *Surprise*, *Trust*). This allows a more fine-grained analysis of emotions connected to a text and improve the task of finding emotional and thrilling scenes. The NRC lexicon has been used on fictional texts in related work (see chapter 2).

Another sentiment lexicon that goes beyond basic positive/negative sentiment was created by Amy Beth Warriner et al.' [29]. The word list extends the ANEW (Affective Norms for English Words) corpus [4] from 1034 to 13915 words with most of those

[1] http://wordnet.princeton.edu/[Accessed 22 March 2018]
[2] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm [Accessed 22 March 2018]

lemmas coming from the SUBTLEX-US subtitle corpus [5]. Three types of rating for affect were chosen and crowdsourcing workers were given the task of assigning words a numeric value of these ratings. The three dimensions *Valence* (or *Pleasure*), *Arousal* and *Dominance* are a widely used model in psychology developed by Albert Mehrabian and James A. Russel [14].
The ratings are each separately expressed in [29] through numerical values between 1 (low) and 9 (high). The first, *Valence* refers to the pleasantness of emotions associated with a word (from *unhappy, annoyed or unsatisfied* on the lower spectrum to *happy, pleased or satisfied*). The second measures the *Arousal* caused by a word, ranging for example from low (*calm, relaxed or sleepy*) to high (*excited, stimulated or wide-awake*). The third rating denotes the amount of *Dominance* associated with a word which represents a feeling of being in control. This rating ranges from *controlled, influenced, submissive* but also *cared-for* or *awed* to *controlling, influential, important* or *dominant*.

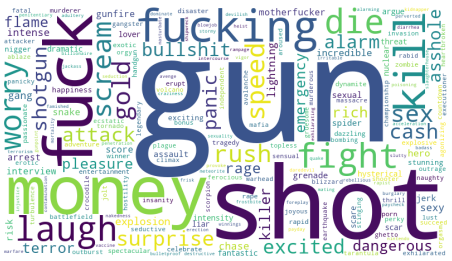
As has been stated before when discussing Mohammad's NRC lexicon, extending sentiment analysis from a simple positive negative scale (valence) to multiple dimensions can be be useful for fictional texts. For the task of this thesis, namely finding a correlation between audio and sentiment and identifying thrilling scenes in movies, an extension with the dimensions dominance and arousal is useful, especially when considering that valence and arousal have been used before in the context of audio analysis [32] as mentioned in chapter 2.

4.1.2 Implementation and Application

The implementation is straight forward. The entire text of a scene, a sentence of the subtitles or other section of the text can be given as input either to a sentiment tool like VADER or one of the self-implemented functions. These custom implementations are based on sentiment lexicons and simply look up each word from the word-tokenized input text in the lexicon. The sentiment score of an entire text is then the average of the scores of each word in the text that was found in the lexicon. If for a given text, no word could be found, the text is ignored instead of assigning a default neutral value. More detailed rules are not implemented, e.g. taking into account negation (*not good*). When working with the movie *Blade* titled after its main character, the issue arose that the large number of occurrences of the word "blade" distorted the sentiment analysis results. Therefore, the set of character names in a movie is used as stop words when working with movie scripts.

Interesting Observations

One possibly interesting observation comes from simply partitioning each of the movie scripts into five sections. This was done with the five-act structure of classical drama plays in mind where the last act is the "catastrophe" often including the death of the hero. For around two thirds of all 897 movie scripts, the last section had higher arousal scores, and lower valence and dominance scores than the first section.



(a) High Arousal (b) Medium Arousal

(c) Low Arousal

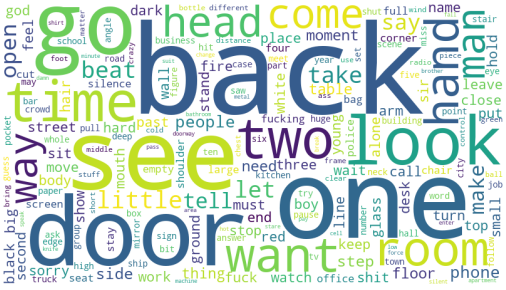
Figure 4.1: Word Clouds for 897 movie scripts. Split into words with high, medium and low arousal with each "level" being 1/3 of the arousal score range (1-9).

When looking at the genres of those movies, for *Arousal* and *Dominance* there was not much difference. For *Valence*, however, there was a noticeable difference. The five most frequent genres of movies where the last section was more negative were *Drama*, *Thriller*, *Action*, *Comedy* and *Crime*. For movies where the last section was more positive, those genres were *Drama*, *Comedy*, *Romance* and only then *Thriller* and *Action*.
In each case, Drama is in first place as it is the overall most frequent genre with 501 of 897 movies having Drama as one of their genres. The genres in position two and three, however, might slightly hint to a relation between genres and happy endings and might be interesting for possible future work.

Figures 4.1, 4.2, and 4.3 show word clouds for the 897 movie scripts in the data set. They show which words in the movie scripts represent *high*, *medium* and *low* sentiment. Words with *sentiment score* < 3.67 fall into the "low" category, words with $3.67 <= score < 6.33$ into "medium" and words with $6.33 <= score$ into "high". The font size in the word clouds visualizes how often a word occurred.

Interesting to note for this thesis are the words especially in the *High Arousal* cloud but also in *Low Valence* and *Low Dominance*. Many of those like *gun, shot, scream* or *kill* are words where in the context of movies, a person would intuitively think of action scenes or other exciting or thrilling moments.

(a) High Valence



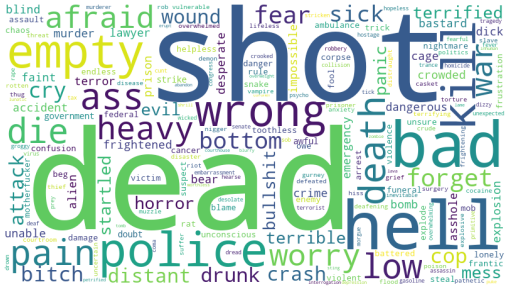
(b) Medium Valence

(c) Low Valence

Figure 4.2: Word Clouds for 897 movie scripts. Split into words with high, medium and low valence with each "level" being 1/3 of the valence score range (1-9).

(a) High Dominance

(b) Medium Dominance



(c) Low Dominance

Figure 4.3: Word Clouds for 897 movie scripts. Split into words with high, medium and low dominance with each "level" being 1/3 of the dominance score range (1-9).

4.2 Audio Analysis

Analyzing a multi-layered medium such as audio can be very complex and often demands extensive domain knowledge. The sheer number of approaches that can be taken unfortunately prevents a comprehensive analysis of the multitude of possible audio features. For the scope of this thesis, the audio analysis presented here is therefore constrained to a small number of features. The underlying framework, however, is developed with the goal in mind of being extensible to work with other audio features as well.
Most of the audio analysis in this work is done in Python using the library librosa[3] [13].

The audio files were preprocessed and downsampled to mono channel .wav files with a sample rate of 22050 Hz. Due to the size of the audio files of entire movies the audio signals were read and processed blockwise in block sizes of 2048 frames per block. The implementation provides the functionality to choose any block length for possible future use. A block size of 2048 has been chosen as it provides a reasonable compromise between short runtime of the program and being detailed enough. With 2048 frames, each block corresponds to a section of about 0.093 seconds. This work mostly focuses on analyzing audio on the level of entire scenes, which means sections with a duration of at least several seconds up to a few minutes. When combining audio with manually annotated movie scripts (as seen later in chapter 5), the granularity of scene time codes can not go below seconds. Therefore the block size does not influence the results as long as each block is distinctly shorter than 1 second.

4.2.1 Audio Features

Relevant for this thesis is a set of affect or emotion related audio features. These have been used successfully in multiple papers analyzing affective video content [10, 32].
The most intuitive of those is the audio energy, representing the loudness. It is psychologically related to the affect dimension *Arousal* [10, 32]. Broadly speaking when thinking of movies the hypothesis is that loud scenes are more arousing or emotionally intensive for the audience and thus more thrilling or exciting than calm and quiet scenes. Typically this applies to scenes like car chases or violent fights in action movies.

Another feature related to arousal are the *Mel-Frequency Cepstral Coefficients* (MFCC). The Mel scale is a subjective scale of how pitches are perceived by human listeners. It was introduced by Stevens, Volkmann, and Newman in 1937 [26]. The mel-frequency cepstrum is used in automatic speech recognition and MFCCs work well for excited and non-excited detection as shown by Xu et al. [30, 32]. They compute MFCCs in their work as follows:

> *The MFCCs are computed from the FFT power coefficients which are filtered by a triangular band pass filter bank. The filter bank consists of 19*

[3] https://librosa.github.io/

triangular filters. They have a constant mel-frequency interval, and cover the frequency range of 0Hz - 20050Hz. The first 4 coefficients are used. [30, 32]

An audio feature that has successfully been used for valence by Xu et al. [30, 10, 32] is pitch. It can be used for emotion detection and distinguishing between positive and negative affect, especially in speech and music. For example a high-pitch average relates to *happiness*, a low-pitch average, however, to *sadness* [10].

4.2.2 Feature Extraction using a Horror Short Movie as Example

This section uses a horror short movie to demonstrate the extraction of audio features introduced in the previous section.

The short movie "Selfie From Hell"[4] is a video clip with a duration of 1:41 min uploaded to YouTube on 9th August 2015. It was produced by Meelah Adams as part of her bachelor thesis on the impact of viral videos [1]. As of March 2018, the video has been watched over 20 million times. It contains little dialogue and builds its suspense and thrill using jump scares, a classic technique in the horror genre. A jump scare is generally characterized by a sudden change in a scene, often suddenly showing gruesome images like a monster. To amplify the effect of making the audience literally "jump", jump scares are mostly accompanied by a large and sudden spike in the audio volume. For this reason, "Selfie From Hell" is well suited as an example for affect-related audio features.

The python library *Librosa* offers extensive functionality for music and audio analysis. It provides a multitude of functions for extracting and visualizing various audio features. If not mentioned explicitly otherwise, all functions are given the default parameter values provided by librosa.

The audio energy for each frame can be extracted with librosa using the function `rmse` of the `librosa.feature` submodule. Figure 4.4 shows the plotted energy of "Selfie From Hell". Around seconds 13, 36, and especially 88 the graph shows spikes that match the image of audio event type 1 (surprise or alarm) as shown in figure 2.1 in section 2.2. Watching the video, especially the section at 88 seconds matches the characteristics of a jump scare, namely a sudden attack in the audio energy after a period of low energy.

[4] https://www.youtube.com/watch?v=EhAFyaObY6U [Accessed on 17 March 2018]

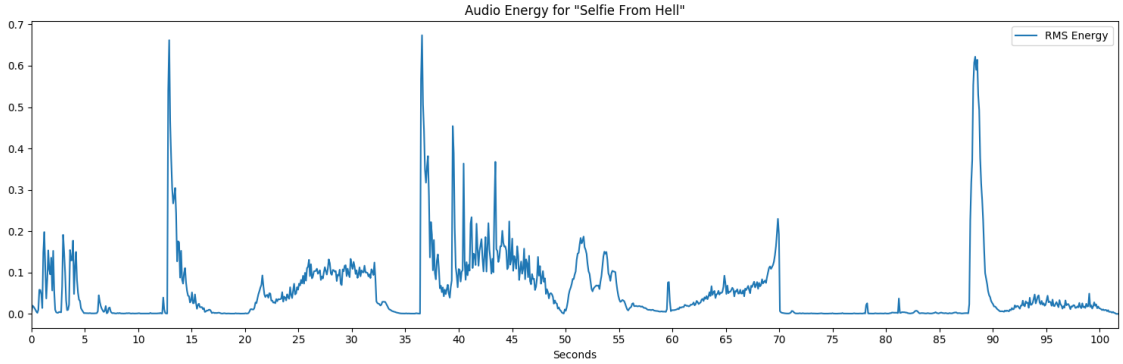


Figure 4.4: Audio energy for the horror short movie "Selfie From Hell". The graph shows spikes in energy similar to the audio energy event type 1 (surprise, alarm, here: jump scare) introduced in figure 2.1.

The Mel-Frequency Cepstral Coefficients (MFCCs) can be calculated using the function `mfcc` from the `librosa.feature` module. Figure 4.5 shows two plots. The first is a graph of the spectral centroid frequency of each frame, which is calculated using the function `librosa.feature.spectral centroid`. It indicates where the mean of the frequency spectrum of an audio frame is located. The second is a spectrogram of the first four MFCCs. The visualization of MFCCs was chosen according to an example from the librosa documentation [5].

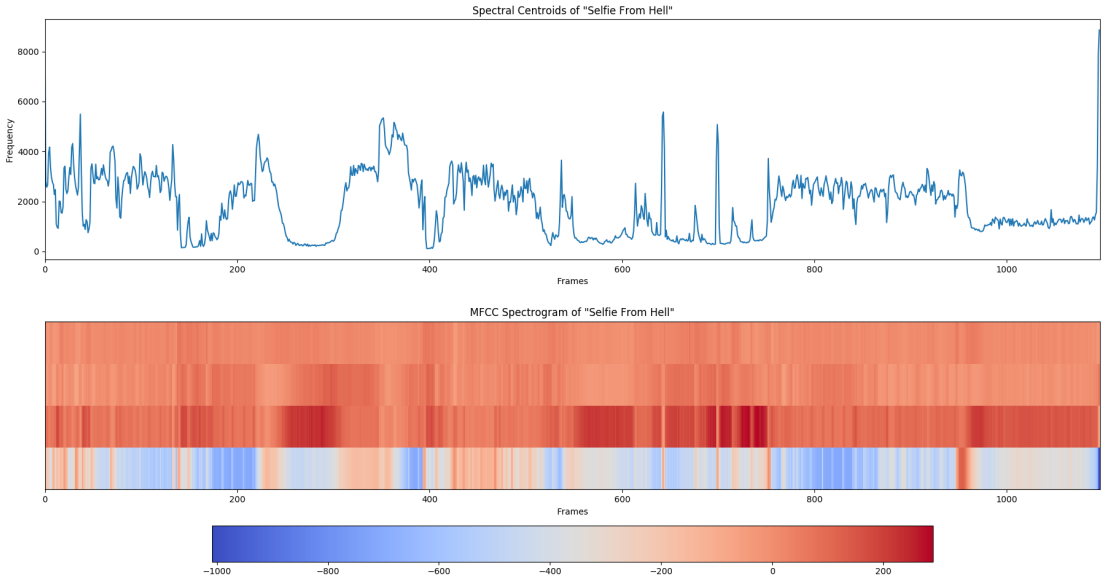


Figure 4.5: Spectral centroids and MFCC Spectrogram of "Selfie From Hell"

[5] https://librosa.github.io/librosa/generated/librosa.feature.mfcc.html

Librosa also offers functionality for analyzing pitch. During the course of this thesis, however, it became apparent that acquiring the domain knowledge needed to interpret and further utilize MFCCs, pitch or other more complicated audio features exceeded the scope of this work. The remainder of this thesis focuses mostly on the audio energy.

To further use the audio features described in this section with movie scripts, their values are split into partitions. Of each partition the average is taken, annotated with a time stamp and saved in csv files. The duration of these partitions can be user-defined although for manually annotated movie scripts, a duration of 1 second was chosen since the time codes for scenes cannot be more fine-grained.

5 | Combining Sentiment and Audio Data

Up until this chapter sentiment analysis and audio analysis have been strictly separate. To achieve the goals of this thesis i.e. mainly to find out whether a connection between sentiment in movie scripts and audio exists, results and data from both domains are combined.

For this section it was only possible to choose a small selection of manually annotated movies. This is due to the fact that movie scripts with manually annotated time codes provide the most reliable mapping of sentiment to time. However, annotating movies by hand is a rather time intensive task and it was not possible to do this for more movies.

The selected movies are mostly from the Action or Horror genre as it is assumed that those have larger swings in sentiment as well as audio and therefore yield clearer results.

These movies are *Blade*, *Hellboy*, *Indiana Jones and the Last Crusade*, *Predator*, *Scream*, *Star Wars Episode IV - A New Hope* and *The Matrix*. Whenever "seven movies" are mentioned, it refers to these movies.

This chapter focuses mostly on the *Valence*, *Arousal* and *Dominance* scores from Warriner et al. [29] and the audio energy.

5.1 A preliminary Note on other Audio Features

This chapter focuses mostly on the audio energy. For the spectral centroids as well as MFCCs, correlation was also examined. The centroid frequencies did not reach correlation scores nearly as high as the coefficients for audio energy. For the MFCCs, only the first of the four computed Mel-Frequency Cepstral Coefficients had similar values. Plotting both the audio energy as well as the first coefficient showed high similarities between both graphs when the audio energy is represented in logarithmic scale. This is shown in figure 5.1. This suggests a redundancy between audio energy and the first MFCC. From this it follows that using separate, single MFCCs likely does not offer advantages over either using other audio features or using multiple coefficients together. For the latter, however, as mentioned in section 4.2.2 the scope of this work was not enough to accumulate sufficient domain knowledge to utilize and interpret MFCCs. Therefore this chapter concentrates on the relation between audio energy and sentiment.

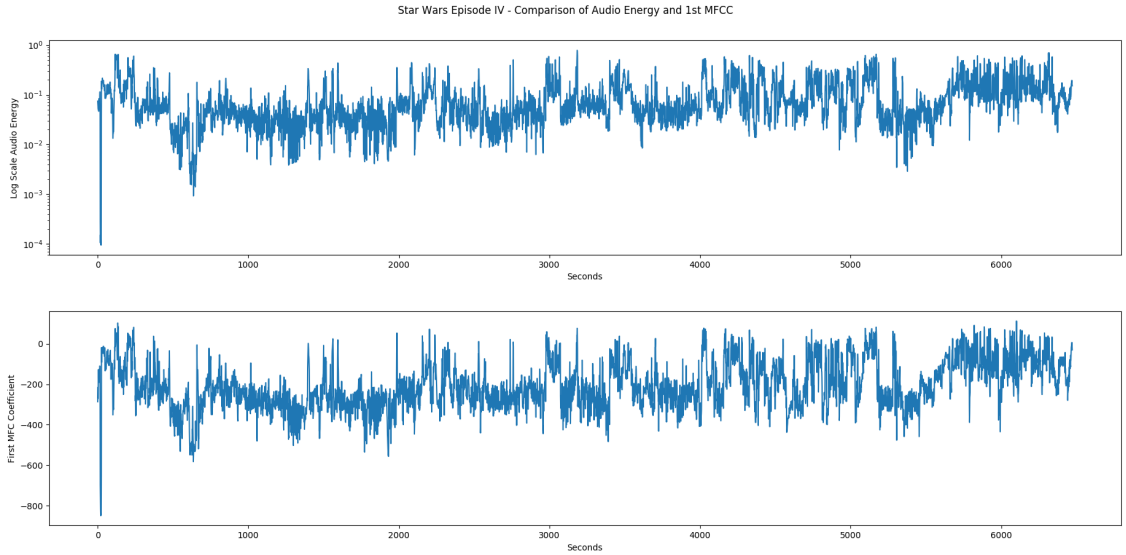


Figure 5.1: Comparison between audio energy and the first Mel-Frequency Cepstral Coefficient for Star Wars Episode IV.

Detailed results for multiple audio features including those that are not described in more detail in this chapter, other sentiment dimensions and broken down for each movie can be found in Appendix C.

5.2 Finding a Correlation between Sentiment and Audio Energy

Initially, the results of sentiment analysis and audio feature extraction have to be brought together in some way. For this task there are multiple approaches more or less suited for finding actual correlation. Once the data is combined it can then be examined for a statistical relationship by using correlation coefficients.

The general hypothesis is that

a) Higher *Arousal* scores of a scene correlate to higher audio energy during this scene and vice versa

b) Higher *Valence* and *Dominance* scores of a scene correlate to lower audio energy during this scene and vice versa

Prominent measures of correlation between two variables are the *Pearson* correlation coefficient, *Spearman's* rank correlation coefficent and the *Kendall* rank correlation coefficient also called *Kendall's tau* coefficient.

The Pearson coefficient assumes an approximately normal distribution for both variables. Figure 5.2 shows that this holds true for valence, arousal and dominance but not

for the audio energy. Therefore the Pearson coefficient had to be discarded. Spearman and Kendall's tau both fit the task at hand which is why in the remainder of this work both are used and compared to each other. In general a correlation coefficient results in a value between -1 and +1 where 0 implies no correlation whereas -1/+1 imply exact monotonic relationship. Positive values imply that as one variable increases so does the other one. Negative values imply that as one variable decreases, the other one increases.

The correlation coefficients and their p-values are computed using the python library SciPy[1].

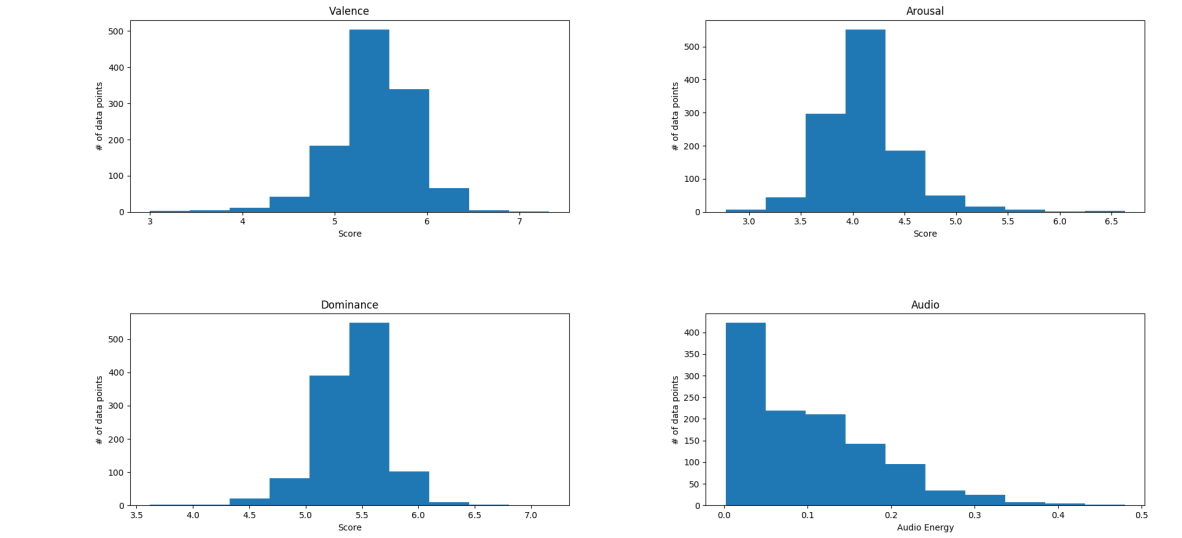


Figure 5.2: Distribution of sentiment and audio energy of scenes from 7 movies

There are multiple approaches described below that were used to examine the existence of a correlation. All have in common that text and audio were split into sections and for each section the average sentiment scores as well as average audio energy were computed.

5.2.1 Plain Text "Fountain" Movie Scripts

The simplest approach is taking the plain text *Fountain* movie script and splitting it into n parts of the same size. The audio data can then also be split into the same number of parts. For each of the n sections the *audio energy* and *sentiment* scores can then be computed.

The resulting correlation coefficients and p-values for seven movies, each partitioned into 200 as of the same length can be seen in table 5.1.

While the absolute correlation coefficients are clearly smaller than 1 the general tendency is that the correlations of valence respectively dominance and audio energy are

[1] https://www.scipy.org/

negative and the correlation of arousal and audio energy is positive.

n = 1400	**Valence**	**Arousal**	**Dominance**
Spearman	-0.143	0.153	-0.109
Spearman p-value	7.726e-08	8.252e-09	4.660e-05
Kendall's tau	-0.096	0.103	-0.073
Kendall's tau p-value	6.605e-08	8.650e-09	4.285e-05

Table 5.1: Correlation Coefficients for movie script sentiment and audio energy for seven movies partitioned each into 200 sections of equal length)

5.2.2 Scenes without Time Codes

A different approach is to take the movie scripts already split into scenes and simply assume that every scene is of the same length. This has been tried and shows similar tendencies as the plain text fountain approach as can be seen in table 5.2. In total, the seven movies contain $n = 1482$ scenes.

Realistically, however, the assumption that all scenes have the same duration can not be held. Therefore this approach was discarded.

n = 1482	**Valence**	**Arousal**	**Dominance**
Spearman	-0.115	0.138	-0.123
Spearman p-value	8.556e-06	9.772e-08	2.062e-06
Kendall's tau	-0.078	0.091	-0.083
Kendall's tau p-value	6.746e-06	1.614e-07	1.887e-06

Table 5.2: Correlation Coefficients for movie script sentiment and audio energy for 1482 scenes from seven movies without annotated time codes

5.2.3 Subtitles with Time Codes

The first approach that takes into account actual time codes of text is to use subtitles. Each subtitle has time codes giving a duration that relates to a specific section in the audio track. This allows a direct mapping of text to audio. Sentences where the sentiment score would have been artificially added due to no word being found in the sentiment lexicon, were ignored. In total, this results in $n = 5369$ dialogue lines. The correlation coefficients and p-values can be seen in table 5.3. Although the p-values are all ≤ 0.011, the actual correlation coefficients are extremely small as well. It is highly likely that this is a result of the general nature of dialogue in movies. The duration of single dialogue lines is only a few seconds which means it is also likely that the alignment of text to audio is not completely correct. Additionally it can be assumed that for the speech being audible by viewers, the audio energy in dialogue scenes is generally lower. This possibly results into very similar audio energy levels in dialogue scenes during the entire

movie. Therefore the reason for the correlation coefficients being small might be that despite differences in sentiment, the audio energy is similar for each subtitle.

n = 5619	**Valence**	**Arousal**	**Dominance**
Spearman	-0.066	0.034	-0.030
Spearman p-value	8.130e-07	0.011	0.024
Kendall's tau	-0.044	0.023	-0.029
Kendall's tau p-value	8.242e-07	0.010	0.025

Table 5.3: Correlation Coefficients for subtitle sentiment and audio energy for 5619 subtitles from seven movies

5.2.4 Scenes with Time Codes

The last approach is to use scenes with their time codes that have been added to the movie scripts as described in section 3.1.3. This results in a total sample size of $n = 1160$ scenes with a direct mapping from text to audio.

Correlation of Audio Energy and Warriner Sentiment

As can be seen in table 5.4 the correlation coefficients are greater than for the previous approaches with the p-values being far smaller at the same time.

n = 1160	**Valence**	**Arousal**	**Dominance**
Spearman	-0.288	0.343	-0.243
Spearman p-value	1.357e-23	2.447e-33	4.478e-17
Kendall's tau	-0.198	0.233	-0.168
Kendall's tau p-value	5.701e-24	1.171e-32	1.112e-17

Table 5.4: Correlation Coefficients for Warriner sentiment and audio energy for 1160 scenes with time codes from seven movies

Permutation tests for the Spearman correlation coefficients of sentiment and audio confirmed the general magnitude of the p-values. A test for the correlation coefficients of audio energy and arousal in table 5.4 with 10^6 random permutations resulted in none having a correlation ≥ 0.343. For valence and dominance, the permutation test resulted in zero permutations having a correlation coefficient ≥ -0.288 respectively ≥ -0.243.

Correlation of Audio Energy and Vader Sentiment

Table 5.5 shows correlation coefficients and p-values for the correlation of audio energy and Vader sentiment.
Interesting to note here is that the *Vader negative*, *Vader positive* and *Vader compound* scores show correlation tendencies as expected. This means *neg* has positive coefficients and *pos* as well as *compound* have negative coefficients, i.e. the higher the audio energy

the more negative and vice versa. The *Vader neutral* score is not useful in this context since a low *neu* score could indicate both negative or positive scenes.

n = 1160	**Vader neg**	**Vader neu**	**Vader pos**	**Vader compound**
Spearman	0.143	-0.066	-0.227	-0.115
Spearman p-value	8.973e-07	0.024	5.184e-15	7.997e-05
Kendall's tau	0.100	-0.044	-0.161	-0.077
Kendall's tau p-value	5.215e-07	0.027	3.981e-15	9.891e-05

Table 5.5: Correlation coefficients for Vader sentiment and audio energy for 1160 scenes with time codes from seven movies

Permutation tests for the Vader sentiment dimensions also confirmed the general magnitude of the computed p-values. Calculating the Spearman correlation coefficients for 10^6 random permutations had the following results:

Vader neg: 0% of the 10^6 permutations had a Spearman correlation coefficient ≥ 0.143

Vader neu: 1.2% of the 10^6 permutations had a Spearman correlation coefficient ≤ -0.066

Vader pos: 0% of the 10^6 permutations had a Spearman correlation coefficient ≤ -0.227

Vader compound: 0.004% of the 10^6 permutations had a Spearman correlation coefficient ≤ -0.115

Correlation broken down for individual movies

More detailed correlation results with an overview over each individual movie can be seen in Appendix C. Notable negative outliers are the correlation between *Dominance* and *Audio Energy* for the movie "Scream"(96 scenes, spearman: -0.038, kendall: -0.028) and the *Valence* correlation for "Predator" (113 scenes, spearman -0.057, kendall : -0.031).

On the other hand, "Indiana Jones and the Last Crusade" with 136 scenes has the highest correlation coefficients observed in this thesis (see table 5.6).

n = 136	**Valence**	**Arousal**	**Dominance**
Spearman	-0.452	0.501	-0.393
Spearman p-value	3.398e-08	5.161e-10	2.175e-06
Kendall's tau	-0.321	0.346	-0.280
Kendall's tau p-value	2.952e-08	2.409e-09	1.344e-06

Table 5.6: Correlation Coefficients for Warriner sentiment and audio energy for 136 scenes from the movie "Indiana Jones and the Last Crusade"

This section strongly implies that a correlation between sentiment and audio energy exists, especially when using entire scenes with time codes and a correct mapping to sections of the audio. The low p-values indicate a significant correlation despite the correlation coefficients being far from an exact monotonic relationship. Movie scenes with higher values of *Arousal* in the movie script tend to show higher audio energy in their respective section of the audio track. Scenes with higher *Valence* or *Dominance* scores tend to correlate with audio sections with lower audio energy.

6 | Discussion and Future Work

6.1 Discussion

Several steps have been taken in this thesis to analyze the audio of movies and the sentiment of the corresponding movie scripts. A reusable data set of 897 movie scripts in a machine-readable XML format has been created. The information contained in these movie scripts can be easily extracted and used for text mining. The approach of automatically annotating movie scripts with time codes from movie subtitles did not result in data which can be reliably used for comparing sections of text to sections of an audio file. To achieve a reliable mapping of text to time movie scripts had to be manually annotated. The scope of this thesis only allowed a very limited amount of movie scripts to be annotated. However, if this were to be extended with multiple additional movies, it would result in a very useful data set.

Thematically this thesis was broad in scope and spans over very distinct research areas. Each of those would profit from more detailed attention. This especially shows when discussing audio analysis which is a very deep research topic. To understand, interpret, and use audio features more detailed domain knowledge is needed.

Related work in the field of sentiment analysis has extracted emotions and sentiment from fictional texts. In the area of audio analysis, related literature has shown a connection between audio features and affect.
This thesis has shown that a certain correlation exists between sentiment expressed by movie scenes in a movie script and the audio track of these scenes in the movie.

Due to the small number of movies the results presented in this thesis can only show tendencies that have to be verified by more detailed analysis in a work of larger scale.

The initial goal of this thesis included using machine learning to automatically detect thrilling or exciting scenes in movies and predict sentiment from audio and vice versa. After some initial attempts (briefly discussed in section 6.2) this topic had to be postponed to future work.

6.2 Future Work

This thesis combined two large research areas, sentiment analysis and audio analysis. The initial approaches presented in this work could provide the basis for interesting future work.

Further utilizing the Data Set/Corpus

The XML-movie scripts could provide the basis for character based sentiment analysis. The sentiment of a character's dialogue could be tracked throughout the course of a movie. How do emotions in the text change? Is this a sign of character development? Can relationships between characters be identified, for example the protagonist and their love interest?
Subtitles and audio files could be the basis of a task related to speech recognition. With the subtitles and their time codes given, the corresponding portion of the movie could be extracted and analyzed. However, it is highly likely that the data is not precise/detailed enough for this task.
Since each XML-movie script contains information about authors and genres, connections between sentiment or audio and genres could be further researched as well as examining possible similarities between movies by the same author. Does the same author use some words time and time again in different movies? Do they structure their movie scripts similarly for every movie?

Machine Learning Tasks

The constructed data set could provide the basis for several machine learning tasks. Some simple approaches were rudimentarily explored in this thesis. With a more refined approach and data set that has been preprocessed to be more suitable, the tasks introduced here could provide interesting future work.

Genre classification of Movies using Sentiment Information

Each movie script contains information about the movie's genres. Therefore a task like classifying movie scripts to genres using sentiment analysis might be possible. One initial approach to this task was computing a single overall sentiment score for each of the 897 movie scripts. Another approach was to compute the relative frequency of words with high, medium or low sentiment scores using Warriner's sentiment lexicon. Each of these three categories was assumed as a third of the entire range a sentiment score could take, i.e. 1 to 9. In section 4.1 these categories are used for the creation of word clouds.

The table below shows the percentages of high and low sentiment words in the two movies "Cars 2", a family friendly Comedy/Animation movie and "Insidious", a Horror movie. The values most likely would not lead to an intuitive decision about which of those movies is a Horror movie if title and genre wasn't known. This example demonstrates, that such an approach is too simple to detect differences between genres.

Movie	Genre	High Valence	Low Valence	High Arousal	Low Arousal	High Dominance	Low Dominance
Insidious	Horror	27.8%	7.7%	3.0%	1.9%	9.0%	4.8%
Cars 2	Comedy	25%	6.5%	4.7%	0.9%	9.3%	3.8%

Table 6.1: Example of the sentiment scores computed for a movie genre classification task.

The resulting data was then fed into classifier algorithms provided by scikit-learn[1] but did not result in reliable predictions of genre labels.

One problem with this approach are the genre-labels. Many movies have multiple genres associated which makes the differentiation between genres challenging. For example 501 of 897 movies are labeled with the genre *Drama*. Using other sources for information about movies' genres could help solving this issue. Another challenge is to examine whether it is possible to compute sentiment scores which are significantly different between for example *Comedy* and *Horror* movies.

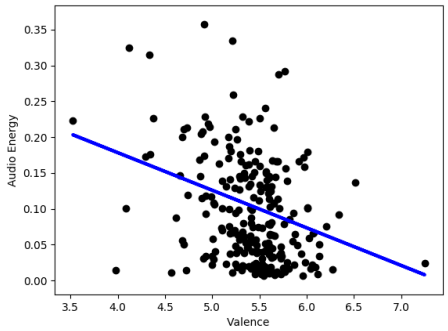
Machine Learning to predict Audio and Sentiment

As shown in chapter 5, a correlation for certain audio features and sentiment dimensions does exist. Based on this an interesting task could be to find out whether it is possible to predict the audio energy from sentiment data or the other way around. This could lead to a system automatically detecting emotional scenes from movies and their screenplays.

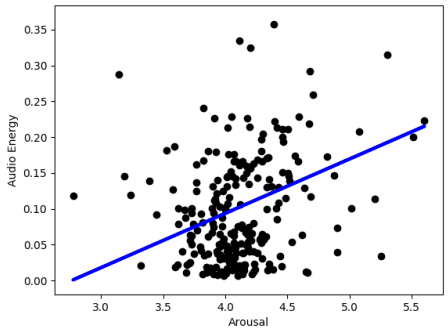
Initial attempts at regression analysis were explored in this thesis using a linear regression model provided by scikit-learn. The data set consisted of 1160 scenes from seven movies. For each scene, the average *Valence*, *Arousal*, *Dominance* and *Audio Energy* was calculated. The linear regression model was trained for predicting the audio energy from each sentiment dimension individually. Each time the data was split into a training set (80%) and a test set (20%).

The resulting predictions on the test set can be seen in figures 6.1 (a)- (d). The graphs show a scatter plot of the test set and the predicted regression line. I interpreted the regression lines as showing similar tendencies to the correlation coefficients in chapter 5. When compared to the scatter plot, however, it is clear that the data can not simply be predicted this way by linear regression. Future work could examine the reasons for this issue and possibly make another attempt at this prediction task with a more refined data set and improved implementation.

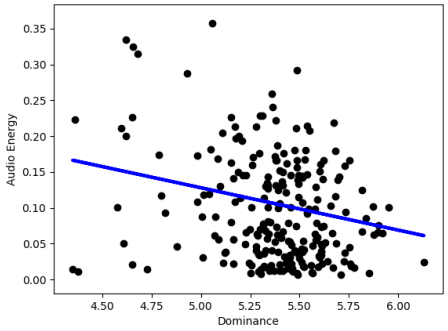
[1] http://scikit-learn.org/stable/



(a) Valence and Audio Energy



(b) Arousal and Audio Energy



(c) Dominance and Audio Energy

Figure 6.1: Regression Plots of a Linear Regression Model for 3 Sentiment Dimensions and Audio Energy of Movie Scenes.

Extending the Analysis Framework

Both the frameworks for sentiment and audio analysis constructed in this work can be extended and optimized. For example a more detailed sentiment analysis tool based on the Warriner sentiment lexicon can be implemented. This application could include functionality to work on a sentence level and take into account things like negation and the general context of a sentence or the length of the input text. The audio analysis tools can be improved as well, incorporating for example a multitude of other audio features.

Experiment on viewer experience

Based on this thesis, a possible extension could be an experiment on viewer experience. This could include test persons watching movies while wearing fitness trackers/bracelets or being connected to a heart rate monitor. This could possibly confirm whether scenes that showed as having high arousal sentiment scores and/or high audio energy are actually perceived as arousing or thrilling by viewers.

7 | Conclusion

This thesis examined affect in movies expressed by the audio on one hand and the movie script on the other. It thus combines approaches from related work that has worked on affect and sentiment in text data and audio separately. A data set was created consisting of a text corpus with movie scripts and subtitles, and the audio tracks of several movies. To analyze audio and text, frameworks for sentiment analysis and audio feature extraction were developed. Using these frameworks, information on sentiment and emotions was extracted from movie scripts. Particular emphasis was placed on the affect dimensions valence, arousal and dominance. Numeric data of audio features was computed from the audio files with the main focus lying on the audio energy. This data was then used to find a correlation between the sentiment dimensions *Valence*, *Arousal* and *Dominance*, and the *Audio Energy*. This correlation especially shows when working on the level of entire movie scenes.

Bibliography

[1] Meelah Adams. Die macht von virals. konzipierung, realisierung und auswertung von viralen spots für den roman 'fuck you zombie'. 2015.

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[4] Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Citeseer, 1999.

[5] Marc Brysbaert and Boris New. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*, 41(4):977–990, 2009.

[6] Alexandre Denis, Samuel Cruz-Lara, Nadia Bellalem, and Lotfi Bellalem. Visualization of affect in movie scripts. In *Empatex, 1st International Workshop on Empathic Television Experiences at TVX 2014*, 2014.

[7] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

[8] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: a high-coverage lexical resource for opinion mining. 2007.

[9] CJ Hutto Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 2014.

[10] Alan Hanjalic and Li-Qun Xu. Affective video content representation and modeling. *IEEE transactions on multimedia*, 7(1):143–154, 2005.

[11] Matthew L. Jockers. Revealing sentiment and plot arcs with the syuzhet package, February 2015. URL http://www.matthewjockers.net/2015/02/02/syuzhet/.

[12] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[13] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[14] Albert Mehrabian and James A Russell. *An approach to environmental psychology.* the MIT Press, 1974.

[15] Saif M Mohammad and Peter D Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics, 2010.

[16] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word-emotion association lexicon. 29(3):436–465, 2013.

[17] Simon Moncrieff and Svetha Venkatesh. Narrative structure detection through audio pace. In *Multi-Media Modelling Conference Proceedings, 2006 12th International*, pages 8–pp. IEEE, 2006.

[18] Simon Moncrieff, Chitra Dorai, and Svetha Venkatesh. Affect computing in film through sound energy dynamics. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 525–527. ACM, 2001.

[19] Simon Moncrieff, Svetha Venkatesh, and Chitra Dorai. Horror film genre typing and scene labeling via audio analysis. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II–193. IEEE, 2003.

[20] Jeho Nam, Masoud Alghoniemy, and Ahmed H Tewfik. Audio-visual content-based violent scene characterization. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 353–357. IEEE, 1998.

[22] opensubtitles contributors. opensubtitles.org - online subtitle database, 2018. URL https://www.opensubtitles.org/. [Online; accessed 24-February-2018].

[23] Silvia Pfeiffer, Stephan Fischer, and Wolfgang Effelsberg. Automatic audio content analysis. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 21–30. ACM, 1997.

[24] Filipe N Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29, 2016.

[25] Benjamin Schmidt. Commodius vici of recirculation: The real problem with syuzhet, 2015. URL http://benschmidt.org/2015/04/03/commodius-vici-of-recirculation-the-real-problem-with-syuzhet/.

[26] Stanley Smith Stevens, John Volkmann, and Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[27] Jörg Tiedemann. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, 2009. ISBN 978 90 272 4825 1.

[28] Jörg Tiedemann. Opensubtitles corpus, 2009. URL http://opus.nlpl.eu/OpenSubtitles.php. [Online; accessed 24-February-2018].

[29] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45 (4):1191–1207, 2013.

[30] Min Xu, Namunu C Maddage, Changsheng Xu, Mohan Kankanhalli, and Qi Tian. Creating audio keywords for event detection in soccer video. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II–281. IEEE, 2003.

[31] Min Xu, L-T Chia, and Jesse Jin. Affective content analysis in comedy and horror videos by audio emotional event detection. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[32] Min Xu, Jesse S. Jin, Suhuai Luo, and Lingyu Duan. Hierarchical movie affective content analysis based on arousal and valence features. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 677–680. ACM, 2008.

[33] Albin Zehe, Martin Becker, Lena Hettinger, Andreas Hotho, Isabella Reger, and Fotis Jannidis. » prediction of happy endings in german novels based on sentiment information «. In *3rd Workshop on Interactions between Data Mining and Natural Language Processing, Riva del Garda, Italy*, 2016.

[34] Zuggy. Subrip - subtitle utilities, 2015. URL http://zuggy.wz.cz/. [Online; accessed 24-February-2018].

Appendices

Appendix A: Source Code

This section presents an overview over the code developed during this thesis.

combine.py Combines sentiment and audio information in csv files.

correlation.py Computes correlation coefficients for sentiment and audio features.

stats_and_visualization.py, utility.py and helper_script.py Scripts containing various functions that are used to perform utilities, visualization of data (i.e. graphs, word clouds etc) and simple tasks such as automated renaming or moving of files etc.

src_audio Contains python scripts working with audio files.

audio.py Extracts audio features from audio files and writing them to csv files

src_text Contains python scripts working with text data such as movie scripts and subtitles.

preprocessing

parse_fountain.py: Parses plain text fountain movie scripts to custom XML format

parse_srt.py: Parses plain text .srt subtitle files to XML

moviescript.py: Extracts scenes, dialogue, characters etc from XML movie scripts.

subtitles.py: Extracts information from XML subtitle files

annotate.py: Annotates XML movie scripts with time codes found in subtitle files

sentiment

lexicons the used sentiment lexicons (NRC EmoLex, Warriner ratings, SentiWordNet)

sentiment.py: Contains the main sentiment class. Reading information from sentiment lexicons and returning sentiment scores for given texts.

ms_sentiment.py: Extracts sentiment information for movie scripts (fountain and xml).

subs_sentiment.py: Extracts sentiment information for XML subtitle files

src_ml Various experimental python scripts implementing approaches to machine learning with data from audio and sentiment analysis

Appendix B: The Data Set

Note: Whenever "seven movies" are mentioned, this refers to the movies *Blade*, *Hellboy*, *Indiana Jones and the Last Crusade*, *Predator*, *Scream*, *Star Wars Episode IV - A New Hope* and *The Matrix*

data_audio The audio files of the seven movies used in this thesis in .wav format.

audio_csvfiles The extracted audio features saved in csvfiles for each of the seven movies. The audio files were partitioned into sections of 1 second duration.

subtitles_xml Subtitles of 1610 movies as XML-files.

moviescripts_fountain 897 movie scripts as plain text files in *fountain* format.

moviescripts_xml 897 movie scripts parsed to XML-files.

moviescripts_xml_time XML-movie scripts of 220 movies with automatically annotated time codes in three versions (see section 3.1.3 and table 3.2). The files in subfolder *diff_80ratio* also contain interpolated time codes.

moviescripts_xml_time_manually XML-movie scripts of seven movies with manually annotated time codes.

genre_csvfiles Several experimental csv files containing sentiment scores and genres for all 897 movies.

audiosentiment_csvfiles csv files containing sentiment information and audio features for seven movies.

Appendix C: Detailed Correlation Results

This appendix contains detailed correlation coefficients for various combinations of sentiment and audio features.

Warriner Sentiment and MFCC for 1160 scenes from seven movies

First MFCC:
Valence
spearman: -0.267
p-value: 2.688e-18
kendall's tau: -0.182
p-value: 1.786e-18
Arousal
spearman: 0.303
p-value: 1.990e-23
kendall's tau: 0.205
p-value: 4.588e-23
Dominance
spearman: -0.202
p-value: 4.995e-11
kendall's tau: -0.138
p-value: 2.703e-11

Second MFCC:
Valence
spearman: -0.028
p-value: 0.361
kendall's tau: -0.018
p-value: 0.363
Arousal
spearman: 0.0479
p-value: 0.124
kendall's tau: 0.031
p-value: 0.126
Dominance
spearman: -0.037
p-value: 0.225
kendall's tau: -0.023
p-value: 0.249

Third MFCC:
Valence
spearman: 0.126
p-value: 4.374e-05
kendall's tau: 0.083
p-value: 5.701e-05
Arousal
spearman: -0.218
p-value: 1.336e-12
kendall's tau: -0.146
p-value: 2.052e-12
Dominance
spearman: 0.070
p-value: 0.024
kendall's tau: 0.046
p-value: 0.026

Fourth MFCC:
Valence
spearman: -0.063
p-value: 0.042
kendall's tau: -0.041
p-value: 0.044
Arousal
spearman: 0.118
p-value: 0.0001
kendall's tau: 0.079
p-value: 0.0001
Dominance
spearman: -0.0523
p-value: 0.092
kendall's tau: -0.034
p-value: 0.099

Audio Energy and Vader Sentiment for seven movies

neg
spearman: 0.135
p-value: 3.303e-06
kendall's tau: 0.094
p-value: 1.896e-06

neu
spearman: -0.058
p-value: 0.047
kendall's tau: -0.038
p-value: 0.052

pos
spearman: -0.231
p-value: 1.107e-15
kendall's tau: -0.164
p-value: 9.016e-16

compound
spearman: -0.113
p-value: 0.0001
kendall's tau: -0.075
p-value: 0.0001

Spectral Centroid and Warriner Sentiment for seven movies

Valence
spearman: -0.013
p-value: 0.653
kendall's tau: -0.008
p-value: 0.681

Arousal
spearman: 0.010
p-value: 0.737
kendall's tau: 0.006
p-value: 0.775

Dominance
spearman: 0.030
p-value: 0.314
kendall's tau: 0.020
p-value: 0.311

Audio Energy and Warriner Sentiment for each of the seven Movies

blade (73 scenes)
Valence
spearman: -0.330
p-value: 0.004
kendall's tau: -0.231
p-value: 0.004

Arousal
spearman: 0.132
p-value: 0.267
kendall's tau: 0.091
p-value: 0.253

Dominance
spearman: -0.283
p-value: 0.015
kendall's tau: -0.194
p-value: 0.015

scream (96 scenes)
Valence
spearman: -0.261
p-value: 0.011
kendall's tau: -0.183
p-value: 0.009

Arousal
spearman: 0.132
p-value: 0.202
kendall's tau: 0.095
p-value: 0.172

Dominance
spearman: -0.038
p-value: 0.718
kendall's tau: -0.028
p-value: 0.690

indiana-jones-3 (136 scenes)
Valence
spearman: -0.452
p-value: 3.398e-08
kendall's tau: -0.321
p-value: 2.952e-08

Arousal
spearman: 0.501
p-value: 5.161e-10
kendall's tau: 0.346
p-value: 2.409e-09

Dominance
spearman: -0.393
p-value: 2.175e-06
kendall's tau: -0.280
p-value: 1.344e-06

the-matrix (142 scenes)
Valence
spearman: -0.277
p-value: 0.001
kendall's tau: -0.189
p-value: 0.001

Arousal
spearman: 0.288
p-value: 0.001
kendall's tau: 0.197
p-value: 0.0005

Dominance
spearman: -0.214
p-value: 0.011
kendall's tau: -0.151
p-value: 0.008

star-wars-4 (424 scenes)
Valence
spearman: -0.262
p-value: 4.596e-08
kendall's tau: -0.176
p-value: 5.715e-08

Arousal
spearman: 0.271
p-value: 1.398e-08
kendall's tau: 0.182
p-value: 2.292e-08

Dominance
spearman: -0.221
p-value: 4.523e-06
kendall's tau: -0.148
p-value: 4.999e-06

predator (113 scenes)
Valence
spearman: -0.057
p-value: 0.548
kendall's tau: -0.031
p-value: 0.623

Arousal
spearman: 0.193
p-value: 0.040
kendall's tau: 0.124
p-value: 0.052

Dominance
spearman: -0.118
p-value: 0.212
kendall's tau: -0.079
p-value: 0.215

hellboy (176 scenes)
Valence
spearman: -0.295
p-value: 6.933e-05
kendall's tau: -0.196
p-value: 0.0001

Arousal
spearman: 0.216
p-value: 0.004
kendall's tau: 0.140
p-value: 0.006

Dominance
spearman: -0.258
p-value: 0.001
kendall's tau: -0.174
p-value: 0.001