Affect analysis for text dialogue in movies

Viktor Palerius
Abstract

Affect analysis for text dialogue in movies

Viktor Palerius

With the surge of services offering video-on-demand through streaming and the increased competition in the field, the need for the provider of the service to be able to fit its content to its users is important. Machine learning can be utilized to in an automatic fashion find user or movie patterns by looking at features from its data.

In this project I create a model with weighting schemes to find affective content from small texts and then explore the potential to also do this for movies by extracting affective features from the movies subtitles. The affective content is determined by using a dictionary with affective labeled words to in a Bag-of-Word fashion score sentences by a dimensional approach with three dimensions called Valence, Arousal and Dominance (V,A,D). The project also consist of a data gathering where two separate datasets are gathered with already V,A,D labeled data, one dataset is found online and the other is self-gathered. These datasets are then used for validation of the affective model and to find the best weighting schema. The best weighting schema is then used to determine affective content during the duration of a movie and utilized to find interesting segment in a movie but also to compare movies and find similarities.

I find that the performance of my model is somewhat decent with best scores on the dimensions Valence and Arousal and that there are small difference based on which weighting schema is used for the model. I find that the model shows potential in the movie domain, by finding interesting segments in a movie but also finding scene-similarities between movies. It does however have its limitation by not being able to distinguish genres and missing affective content expressed through visual or audio cues. Finally I argue my model could be incorporated into a larger machine learning model to determine similarities in movies or find user patterns but it also requires similar models to determine affective content from the audio and visual in the movie.

Handledare: Alden Coots
Ämnesgranskare: Kristiaan Pelckmans
Examinator: Lars-Åke Nordén
UPTEC IT 17028
Tryckt av: Reprocentralen ITC
# Contents

1 Introduction 4  
1.1 Delimitations .................................................. 4  
1.2 Problem definition ............................................. 5  

2 Previous work 6  

3 Theory 7  
3.1 Emotional models .............................................. 7  
3.1.1 Human emotions ............................................. 7  
3.1.2 Dimensional versus Categorical model .................... 7  
3.2 Natural language processing (NLP) techniques .............. 9  
3.2.1 BoW and Dictionary Lookup ................................ 9  
3.2.2 Tokenization, Stemming and Stopword removal .......... 10  
3.2.3 Weighting schemes ......................................... 11  
3.3 Error metrics .................................................. 12  

4 Data gathering 14  
4.1 Gathering VAD labeled data ................................. 15  
4.1.1 Gathering own data ....................................... 16  
4.1.2 Using preexisting data .................................. 19  
4.2 Finding and parsing movie subtitles ....................... 20  

5 Method 22  
5.1 Building a dimensional model ............................... 22  
5.1.1 Weighting schemes ....................................... 25  
5.2 Validating the model ......................................... 26  
5.2.1 Analyzing movies ......................................... 27  

6 Result 29  
6.1 Small text results ............................................. 29  
6.1.1 EmoBank text result ..................................... 29  
6.1.2 Movie text results ....................................... 31  
6.2 Movie results ................................................ 33  
6.2.1 Single Movie ............................................. 33  
6.2.2 Comparison of movie scenes ............................. 33  

7 Conclusion and Discussion 36  

8 Future Work 39
1 Introduction

Analyzing movies in an automatic fashion to be able to extract movie features has become especially important of late. With the surge of new video-on-demand services available online for consumers, which usually offers the ability to watch commercial movies or TV series for a subscription fee, the importance of the company to be able to fit its particular service to its users has increased. One way to do this is with the help of machine learning where a model is created to be able to compare and find similar movies or determine particular usage patterns. Collaborative filtering is a common approach for this which relies on feedback from users to find similar movies and make recommendations. This has the problem of being dependent on other users which usually penalizes newer movies and "unpopular" movies. Another approach is the content-based system where characteristic (features) from a movie is extracted to be used in a recommendation engine or other classifications systems to improve the service. In this project the emotional/affective impact of a movie will be extracted to be used as features for a larger machine learning model. The features will be devised by looking at the subtitles of a movie and then determine the emotional impact of the text over the duration of the movie.

Automatic analysis of affective content from text is a difficult topic due to the subjective nature of emotions. The topic is also rather unexplored and therefore hard to validate. To not only validate against own-assumed notions of emotional impact in movies, the project also consist of a data gathering. The data gathered are small texts with user labeled emotions to validate the created model against before analyzing particular movies. This will give an indication of how well the model works in terms of determining affective content from text.

1.1 Delimitations

The project is limited to only extract affective features from text in the English language. This since most natural language libraries are devised for English text and most commercial movies have a corresponding subtitles in the English language. This project will focus on extracting affective content from the text, namely the subtitles for the movie. It could potentially be extended to be used together with other affective features from a movie in future studies.
1.2 Problem definition

The project consist of an exploration into how to determine and validate the affective impact of small texts. Method for extracting the affective content as features will be explored. To do this the following steps are followed.

1. Determine a representation for affective content
2. Create a model to extract affective content from text as features
3. Gather suitable data to be used for validation
4. Devise concrete error metrics to validate the model
2 Previous work

The study of defining in an automatic fashion polarities of emotion in text is a still ongoing, highly relevant field of study. One of the most explored studies of determining text polarity, is called sentiment analysis. With sentiment analysis one tries to determine polarity of the text by classifying a text into being either positive, neutral or negative. In [1] research into polarity classification of different types of reviews by using unsupervised classification is done. By looking at adjective and adverbs in the review and using semantic orientation of several of these a review is classified into being positive or negative. In [2] sentiment classification of movie reviews are made. By creating a lexicon of appraisal groups connected to a positive or negative polarity and the utilizing a Bag-of-words approach, a review can be classified into positive or negative polarity.

Expanding upon the notion of sentiment analysis is the study of emotional or affective analysis, by instead of deciding a binary polarity score incorporate and classify texts based on which emotions the text instead might invoke. In 2007 SemEval [3] launched a competition where one of the tasks included creating a model which should classify news headlines into one of six emotional categories.

In [4] they explore the dimensional domain in terms of determining affective content in texts, and which heavily inspired this project. They are able to show the potential of an automatic algorithm for deciding affective content and shows perhaps the best performance for doing this so far. By using weighting schemes and dimensional scored words they are able to defer affective ratings for small texts
3 Theory

3.1 Emotional models

This chapter briefly describes the underlying theory of human emotions, what are emotions and how are they expressed. Different emotional models are also compared which are used to be able to determine emotions based on different stimulus.

3.1.1 Human emotions

All humans have felt emotions which influences their behavior in a certain situation. For example one can be feeling fear due to being in a hostile environment or happiness by being in a safe environment. Emotions can be said to be bound to different situations and help evaluate those situations and can in that way influence our next step in our behavior [5]. But the process of emotions are complex and has many different components making it hard to pinpoint the exact definition of emotions [5]. Different definitions have been proposed by different theorist of what an emotion actually is. But researchers largely agree on the characteristics of an emotion [6]. Emotions are usually brief intense reactions occurring during a specific situation. Every emotion originates from activity in the brain [7]. However which signals the brain will send or which emotion will occur during a situation depends on the particular persons biology and experiences.

3.1.2 Dimensional versus Categorical model

Research into the process of determining affective content has been studied using a wide range of procedures. To study the way of human emotions different type of stimuli has been used to measure the reaction. Stimuli such as pictures of facial expressions [8], different audio clips [9] or written words [10]. These methodologies to determine emotions can mostly be divided into the two categories the categorical and the dimensional model.

With the categorical model, emotions are split into categories corresponding to particular emotional states. In computational studies most categorical models employ the theory of Ekmans [11] six basic emotions (anger, disgust, fear, joy, sadness) to devise emotional content into emotional groups. The theory behind this is that all humans share common emotional traits and can be classified into these corresponding emotional groups [11]. The categorical model can be used in a coarse grained approach where certain reactions are purely classified into one of the corresponding emotional groups or a more fine-grained approach can be taken where each reaction is assigned in
an interval for each emotional group, ranking the strength for each of the emotional groups for that particular reaction.

The dimensional model instead tries to label emotional granularity by using a discrete scale over several dimensions to map certain emotions. By studying human reactions when looking at pictures for emotional expressions on human faces, several studies [12][13][14] have found evidence for at least 2 or 3 dimensions to rank emotional reactions. The theory behind this being that rather than people experiencing distinct emotions we instead feel emotions on a range on these dimensions. The first two dimensions commonly called Valence (unhappy-happy) and arousal (calm-intense) ranks a person happiness and intensity felt from experiencing a emotional reaction. The third dimension introduced later and not as widely supported as the other two, is commonly called dominance which ranks a persons dominance over a situation, ranging from being under control till being in control. These dimensions together can be called the VAD emotional model and with the three dimensions being separate from each other. However using the scores for all the three dimension together one can form an emotional labeling for a particular reaction. Since this model is not as easily understood as the categorical model, Bradley et al. [15] introduced a Self-assessment manikin (SAM) (see Figure 1) to be used to be able defer the particular emotional ratings. The SAM details pictures for each of the dimension and while the ratings can be different, (commonly 1-5 or 1-9) the pictures details the same things.
3.2 Natural language processing (NLP) techniques

This section describes common techniques used to process text into more manageable chunks to be used for different types of information retrieval.

3.2.1 BoW and Dictionary Lookup

A common technique used in NLP is the Bag-of-words (BoW) [17] approach mainly used for document classification task. The BoW approach takes a text and represent it instead by the words (the bag) present in that text regardless of the grammar or word order in the text. For example the text

"Stig usually plays football in the summer. Erik also plays football in the summer."

would by the BoW approach be represented as

[Stig, usually, plays, football, in, the, summer, ., Erik, also]

which also could be extended to characterize the text by term frequency

[1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1]
The BoW approach can also be extended to represent words as _n-grams_ that is that each set in the bag not only contains the word but also the n closest neighbors to that word. This by the argument that the characteristic of a word is also described by its relationship with the words next to it. For example representing the text now instead with a bi-gram bag would give us: 

[Stig usually, usually plays, plays football, football in, in the, the summer, Erik also, also plays]

However the higher the n-gram the larger the bag will become due to the more combinations available.

A common approach is to combine the BoW with a dictionary to be able to classify a text based on its words and a dictionary defining word characteristics. This is common to use in a sentiment classification task where the dictionary contains words and the polarity scores, in a simple example the words could be defined as either; negative, neutral or positive. Then by representing the text by its BoW and term frequency perform a dictionary lookup to devise their polarity scores and combine them to get a final polarity score for the text. For example defining our simple unigram represented text with the example dictionary:

<table>
<thead>
<tr>
<th>Word</th>
<th>Polarity score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likes</td>
<td>Positive</td>
</tr>
<tr>
<td>Summer</td>
<td>Positive</td>
</tr>
<tr>
<td>In</td>
<td>Neutral</td>
</tr>
</tbody>
</table>

We would get: 3 Positive, 1 Neutral, 0 Negative and classifying the text as being _positive_.

### 3.2.2 Tokenization, Stemming and Stopword removal

When utilizing NLP techniques it is very common to parse the text in specific ways to create a better underlying structure. Three common approaches to this are _Tokenization_, _Stemming_ and _Stopword removal_. Tokenization is the process of dividing a text into smaller subset called tokens. Usually done at the word level indicating that each token is a word. Tokenization can be used as a underlying step for the BoW approach to be able to divide a text into its words. Several word-level tokenization tools might use different heuristic to split and differ in the tokens created due to the somewhat diffuse definition of what a word is. However the most common heuristics are to split words based on whitespace or punctuation.
Stemming is the process of reducing a word to its word stem, root form. This to avoid having word in different forms which could for example decrease the performance of a dictionary lookup or word comparison algorithms. A stemming algorithm tries to create linguistic rules to either conclude a word is in its root form or transform it. Most algorithms are created for the English language and one of the most common one is called Porter stemmer [18] which also have been used to create additional extension stemmers.

Finally stopword removal is done to remove words that lack any characteristic for the task at hand. Usually a list is created, domain specific for the particular task, with words that should be removed and not processed in later steps. Most NLP libraries already include a stopword list with most of the uncharacteristic common words which can be utilized, words in those list usually include words such as and, if, or and so on.

### 3.2.3 Weighting schemes

In text information retrieval (IR) a common approach is the create weighting schemes to set different weights for documents, sentences or words to determine their impact on the text to work on. This with the underlying presumption that some entities define the text more than other and thus giving them a higher weight, emphasizes their overall contribution to the text over other lower scored entities. Several different weighting schemes and extensions on the most common ones have been developed to be used for different approaches and to be compared. Two of the most common ones will be explained, here defined as working on word level.

**Term frequency:** Term frequency, TF is as the name suggest a weighting schema that weights in our case words based on their frequency present in the document. Then weights could be set for each word for example based on raw frequency counts as in our example sentence; "Stig usually plays football in the summer. Erik also plays football in the summer." which as mentioned could be represented as the TF weight vector \( V = [1, 1, 2, 2, 2, 2, 2, 2, 1, 1] \). Term frequency weight vectors can either be created by one weight vector for all documents or a weight vector for each distinct document.

**Term frequency - Inverse Document Frequency:** Tfidf is an extension on the regular TF frequency weighting scheme, with the logic that words that appear frequently both in several documents as in one document should be weighted lower than words that only occur frequently in one document but few in others. That is that the word that are rare in terms of appearance in several documents will be given a higher weight than the more common
ones. It can be described with the mathematical notation:

$$TFIDF_i = TF_i \times \log \frac{|DOC|}{DF_i}$$

Where $DOC$ is the number of documents and $DF$ the amount of documents containing the word. In a simple example, Tfidf weights for particular words can be calculated using the following two example documents.

<table>
<thead>
<tr>
<th>Word</th>
<th>Word Count</th>
<th>Word</th>
<th>Word Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likes</td>
<td>1</td>
<td>Hate</td>
<td>1</td>
</tr>
<tr>
<td>Summer</td>
<td>2</td>
<td>Winter</td>
<td>1</td>
</tr>
<tr>
<td>In</td>
<td>3</td>
<td>In</td>
<td>2</td>
</tr>
</tbody>
</table>

Where the weight for the word In can be calculated as followed for the two different documents

$$tfidf_{d=1}(in) = 3 \times \log \frac{2}{2} = 0$$

$$tfidf_{d=2}(in) = 2 \times \log \frac{2}{2} = 0$$

(1)

and the word summer and winter for respective document can be calculated as followed:

$$tfidf_{d=1}(summer) = 2 \times \log \frac{2}{1} \approx 0.602$$

$$tfidf_{d=2}(winter) = 1 \times \log \frac{2}{1} \approx 0.301$$

(2)

### 3.3 Error metrics

When creating a prediction model, several error metrics need to be used to be able to measure the accuracy of the model against test data. Two commonly used error estimators are the mean squared error (MSE) and the root mean squared error (RMSE). They measure how close a fitted line is to data points, with the squaring done so negative values do not cancel out the positive values. The difference being that the RMSE simply takes the root out of the MSE to directly interpret the error in terms of the measurement units.

$$MSE = \frac{1}{n} \times \sum_{n=1}^{n} (X_t - X_p)^2$$

$$RMSE = \sqrt{\frac{1}{n} \times \sum_{n=1}^{n} (X_t - X_p)^2}$$

(3)
Where $X_t$ is the target point and $X_p$ the models prediction of the point. The value $n$ being the amount of points to calculate the error for.

Another metric for regression analysis is the Pearson correlation coefficient [19], (Pearson $r$). The $r$-score gives an indication for linear correlation between two variable sizes $X$ and $Y$, with the range of [-1,1] indicating by range from either a fully negative correlation (-1), no correlation at all (0), or a full correlation (1) between the two variables. The value of the Pearson correlations does not depend on the specific units used for measurements, for example the correlation between two variables will be the same regardless if one variable changes its measurement unit. The Pearson correlation will not however give you any information about the slope of the line but will only tell you whether there is a relationship. To interpret the actual $r$-score one can approximately interpret the score based on the following rules:

- High Correlation: [0.5 to 1.0] or [-0.5 to -1.0]
- Medium Correlation: [0.3 to 0.5] or [-0.3 to -0.5]
- Low Correlation: [0.1 to 0.3] or [-0.1 to -0.3]

The Pearson correlation for $X$ and $Y$ can be calculated by the following formula.

$$r = \frac{n(\sum_{i=1}^{n} X_i Y_i) - \sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i}{\sqrt{[n \sum_{i=1}^{n} X_i^2 - (\sum_{i=1}^{n} X_i)^2] \cdot [n \sum_{i=1}^{n} Y_i^2 - (\sum_{i=1}^{n} Y_i)^2]}}$$  \hspace{1cm} (4)
4 Data gathering

The created model to determine emotional content in text was determined to use the dimensional approach, using the three dimension Valence, Arousal and Dominance (VAD) to determine emotional content. The underlying reasons behind the use of the dimensional model was that it provides real-valued values which works great as features for additional model learning such as similarity or content based recommendation models. It could also be combined with similar emotional models for audio or video which is in the progress where I did my project and finally the VAD approach has shown promising results in the text emotional domain (see section 2).

The reason behind using the three dimension Valence, Arousal and Dominance was due to a extensive dictionary with English VAD labeled word available to me. The dictionary was created by Warriner et al [10] and includes around 14000 English lemmas and their corresponding VAD rating. The dictionary was created from crowd sourcing and also includes word ratings for different gender and age groups. The created model however, uses the general ratings from all people who rated the words regardless of their gender or age, due to the ability of creating a more general model. Each word was rated in the range 1-9 for each value and Figure 2 show the distributions of the ratings for the three domains.

![Figure 2: The data distribution of the VAD dictionary](image)

As can be seen the value for arousal tends to be rated a bit lower than for
its corresponding Valence and Dominance ratings. As described in [10] this may be due to reading a single word most likely won’t invoke such a strong reaction but instead be rather neutral. The mean for the dimension in the Vad dictionary is for Valence 5.06, Arousal 4.21 and for Dominance 5.18 showing that the means for Valence and Dominances are close to neutral rating and the mean for Arousal being slightly below the neutral rating.

To be able to get a better understanding of the three dimension and how words impact them, two tables down below shows the top 5 highest rated in respectively dimension.

<table>
<thead>
<tr>
<th>Valence (R)</th>
<th>Arousal (R)</th>
<th>Dominance (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vacation (8.53)</td>
<td>Insanity (7.79)</td>
<td>Paradise (7.9)</td>
</tr>
<tr>
<td>Happiness (8.48)</td>
<td>Gun (7.74)</td>
<td>Win (7.86)</td>
</tr>
<tr>
<td>Happy (8.47)</td>
<td>Sex (7.6)</td>
<td>Incredible (7.74)</td>
</tr>
<tr>
<td>Christmas (8.37)</td>
<td>Rampage (7.57)</td>
<td>Self (7.74)</td>
</tr>
<tr>
<td>Enjoyment (8.37)</td>
<td>Lover (7.45)</td>
<td>Completion (7.73)</td>
</tr>
</tbody>
</table>

And the top 5 words with the lowest rating for each of the dimensions:

<table>
<thead>
<tr>
<th>Valence (R)</th>
<th>Arousal (R)</th>
<th>Dominance (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedophile (1.26)</td>
<td>Grain (1.6)</td>
<td>Dementia (1.68)</td>
</tr>
<tr>
<td>Rapist (1.3)</td>
<td>Dull (1.67)</td>
<td>Alzheimers (2)</td>
</tr>
<tr>
<td>AIDS (1.33)</td>
<td>Calm (1.67)</td>
<td>Lobotomy (2)</td>
</tr>
<tr>
<td>Torture (1.4)</td>
<td>Librarian (1.75)</td>
<td>Earthquake (2.14)</td>
</tr>
<tr>
<td>Leukemia (1.47)</td>
<td>Soothing (1.91)</td>
<td>Uncontrollable (2.18)</td>
</tr>
</tbody>
</table>

4.1 Gathering VAD labeled data

Due to the nature of classifying emotional content in text is somewhat uncharted territory finding VAD movie data to validate the model was not possible. But to not only rely on own made assumption regarding the models validness other emotional texts with VAD labels was gathered. Since the model would be designed using a word dictionary with labeled VAD values for each word, it was decided to try to gather small texts with labeled VAD values. These text could then be used to test the model and validate the result. Two datasets with VAD labeled small text were used. The first dataset were from a self-made data gathering where small texts were rated by people. The second was a large, online available VAD dataset including text from different domains. These two datasets will be described in more detail in the following subsections.
4.1.1 Gathering own data

To be able to gather VAD labeled data, sentences from different movies was retrieved. The movies the sentences were taken from were self-picked with the notion that they should represent different genres and therefore different types of expression and sentences. The following movies were used to extract sentences.

1. Pulp Fiction
2. Collateral
3. Die Hard I
4. Get Hard
5. Interstellar
6. Warcraft
7. Whiplash
8. Up
9. Back To The Future I
10. The Shawshank Redemption
11. Requiem For A Dream
12. Good Will Hunting
13. Jurassic Park I

The sentences were then extracted by first selecting all sentences that at least included three words also present in the Warriner VAD dictionary. Finally 500 sentences from the 13 movies were picked at random to be used for rating purposes.

The data gathering was done by creating a website where users could log in and rate the sentences previously selected. The website Figure 3 4 was created using Flask [20] and provided an easily manageable interface for users to rate sentences and track their progress.
To be able to start rating, a user after signing up, needed to complete a simple tutorial. This tutorial explained the VAD rating system much like the explanation done in [10] for their data gathering. The Self-assessment manikin (SAM) was also explained, which was used to be able to defer the particular ratings for each emotional dimension. Finally to complete the tutorial the user needed to rate the sentence fully neutral, that is a 5 on Valence, 5 on Arousal and 5 on Dominance on a example text.

To be able to better understand the users rated data, control words were added to determine bias. These control words were simply words taken from the VAD dictionary which users also had to rate. Then a comparison between the actual values for the word in the dictionary compared to what the users rated it as could be done. In total 100 words where picked at random from the VAD dictionary to be used for control ratings.
For each sentence the user had the option to either skip the sentence if the user could not decide on a particular rating or to choose a rating for Valence, Arousal and Dominance by selecting the right SAM picture for each. Each user also had a progress or level bar to indicate how many sentences they had rated but to also motivate them to continue rating, working as a rewarding game system. Whenever a user rated or skipped a sentence the next sentence where picked at random from the pool of available and not yet rated sentences which was then presented for the user.

Data Distribution

The text rated data from the website included in total 9 users rating sentences which combined rated 775 control words plus sentences. From the 775 rated texts, 637 was sentences and 138 control words. To be able to decrease the amount of rating dependency from just a single person only sentences which were rated by more than one user was selected to be used for validation. In total 190 sentences and 33 control words had been rated by more than one user. The rating for each of these texts will then be the average rating for all users who rated it. The data distribution for the rated sentences for each dimension is shown in 5.
As Figure 5 shows the rating for arousal tends to be higher rated than for both Valence and Dominance. The mean for each value is for Valence at around 4.7, Arousal at around 6.2 and Dominance at around 5.2.

4.1.2 Using preexisting data

The data set called EmoBank [21][22] spans VAD annotated small text from multiple genres such as news articles or blog posts. In total around 10 000 small texts were rated and each text rated by using a 5 point VAD scale, making values range from 1 to 5 with 3 being the neutral value for respectively dimension. Each text is also split into ratings for both the writer of the text and the reader, that is the emotional impact it has on writer or reader.

To use the EmoBank dataset for the model the reader rating perspective was chosen, this due to actually showing the emotional impact of the reader which corresponds with how the model should work when classifying movie subtitles. Finally the dataset was filtered to only include those text which had at least 1 word in them that was also in the VAD dictionary (described in Section 5). This to avoid validating the model against text that had no words present in the text that could be used to get a VAD rating for. In total 9180 of the texts remained to be used for validation. The distribution of the texts in their corresponding emotional dimensions can be seen below.
Figure 6: The data distribution for the three dimensions Valence, Arousal and Dominance from the EmoBank rated texts

The distribution clearly shows that most of the text are rated neutral or very close to neutral rating for all the three dimensions. As argued in [21] this is due to as in most case the text is not invoking any particular weak or strong reaction and a neutral rating for it is chosen. Comparing it to the Warriner word VAD dictionary the same pattern emerges except for arousal where the average for words are lower than for the texts used in EmoBank.

4.2 Finding and parsing movie subtitles

To be able to analyze text emotional impact on movies, subtitles were required. To find subtitles the website subscene [23] was used which provides a comprehensive list over subrip files (srt) for several movies. To decrease the potential misspelling errors for the subtitle, Subscenes ranking system was used to pick the highest ranked subtitle to use for a movie.

When a subrip file was selected it was first converted to utf-8 format to avoid compatibility issues and then parsed and split into several parts to act as the scenes used to parse for emotional impact. A SubRip file has the following format

Subrip file format

# Subtitle index
# Time range
Which makes it easy to parse a subrip file for its content and split it into parts based on the time it should appear in a movie. To parse the subtitle the library pysrt [24] a python library for parsing subrip files was used.

Finally all the movie parts were added to a list which each element corresponding to the entire text (all subtitles) for a particular movie part. The list was then sent to the emotional model to determine the distinct V,A,D scores for all the elements in the list.
5 Method

5.1 Building a dimensional model

The model to determine the emotional impact from text was built using python and used the library NLTK\[25\] to be able to utilize their tokenization, stemming and other NLP utilities.

The emotional model itself takes several argument at run start. The first being the VAD dictionary to use, in this project only the Warriner VAD dictionary was used. The second argument being as mentioned the list with elements containing the text to process an emotional VAD score for. Each of the element in the list is processed as a document being that the weighting schemes used are processed for each of the elements and the model returns a VAD score for that particular element.

Each elements text is processed in the model by first splitting up the text into sentences using the NLTK sentence parsing. For each sentence the sentence is then split into the words present (tokenization) and parsed through the dictionary to see which words in the sentence also occurs in the VAD dictionary used. All the words not present will be removed to create a final list only with words found in the VAD dictionary. To check if a word is in the dictionary the word in its low-case original form is first used to perform a dictionary look up. If the word is not in the dictionary we stem the word in hope its root form is present in the dictionary. If the stemmed word is still not in the dictionary the word is not added to the final list. When all elements in the original list have been parsed and the new list with all words found is created, the model parses each word separately to compute its appropriate weight and its VAD score found in the dictionary. Figure 5.1 shows the models steps in a flow chart. To create a final score with score for Valence, arousal and dominance for each of the elements sent to the model, we first sum up all the VAD scores for words in each of the sentences found in each particular element together with the word weight (See section 5.1.1) and compute the score from the following formula.

\[
(V, A, D)_{sentence} = \frac{\sum_{i=1}^{n} (\delta_i \times W_i (V_i, A_i, D_i))}{\sum_{i=1}^{n} \delta_i}
\]  

(5)

Where $\delta_i$ is the weight for a particular word and the $W_i$ the words particular scores for valence, arousal and dominance. Finally the V,A,D scores from all the elements sentences are summed up using the following formula.
Computing the total weight sum for all words weight in sentence $i$

$$S_i = \sum_{i=1}^{n} \delta_i$$  \hspace{1cm} (6)

$$\left( V, A, D \right)_{\text{final}} = \frac{\sum_{i=1}^{n} (S_i \times (V, A, D)_{\text{sentence}})}{\sum_{i=1}^{n} S_i}$$

This assures that the $V,A,D$ rating calculated for the particular element remains in the same range for the three dimensions as the $V,A,D$ dictionary used. Finally a list with the computed valence, arousal and dominance scores for each element is returned.
Emotional Model

For each element

Parse sentences

For each sentence

Tokenization

For each word

Is word in dictionary?

no

Ignore word

Can word be stemmed?

no

stemmed?

yes

Stem word

yes

Append to final list element

For each element in final list

Parse text

For each word

Compute word weight

Compute V,A,D scores

Return list with V,A,D scores

24
5.1.1 Weighting schemes

Due to emotional analysis being rather unexplored an interesting notion was to explore the impact of different weighting schemes as done in [4], that is different ways to rank the importance of a particular word in the overall V,A,D score. It was decided to use 4 different types of weighting schema to evaluate the model with. Regardless of weighting schema used the other steps in the emotional model remains the same, but with an overall different results due to the word weights being different. To devise word weights each element in the list was used as a separate document. The 4 types of word weights used was:

1. **Equal weight (AVG)**: Give each word the weight of one, the overall result will be the average V,A,D scores for all words.

2. **Term-Frequency (TF)**: Word weights defined by the frequency of the word present in all document.

3. **TFIDF**: Frequent word but only appearing in few documents giving a higher weight

4. **TFIDF-ES** Own devised extension of the TFIDF to also take into account the emotional score of a word. That is a word is given a higher weight if it is far away from the neutral score in valence, arousal and dominance.

The Term-frequency and TFIDF weighting schemes are explained more thoroughly in 3.2.3, the TFIDF-ES is an extension using also the emotional rankings of a word to determine the weight. If a word is far from having a neutral score in the valence, arousal and dominance dimensions (5 being neutral in our Warriner dictionary case) it is given a higher weight and therefore a higher impact on the overall V,A,D score. The argument behind this extension is that it would be reasonable to assume that the emotional score of a small text is determined by the words that are more emotional or less emotional intense rather than the ones being closer to neutral. The formula for computing the TFIDF-ES weight is shown down below.

\[
TFIDFES = TFIDF \times \sum_{i=1}^{3} |K_i - N_i|
\]

\[
K \in (V_i, A_i, D_i), N \in (5.0, 5.0, 5.0)
\]
5.2 Validating the model

As described in 4 to be able to validate the accuracy for the model, classifying emotional content, specific small text with labeled Valence, Arousal and Dominance ratings were gathered. These data sets could then be used to not only validate the model but to also test the four different weighting schemes and see which performed best. To be able to get a score-metric and to compare the four weighting schemes it was decided to use the root mean squared error (RMSE) together with the Pearson correlation (r-score). The weighting schema with the lowest RMSE and the r-score closest to one would be determined to be the best. Using the two metrics together work well where the Pearson correlation gives an indication of how well the predicted values correlate with the actual labeled values but no indication of the closeness of them. However also using the RMSE which gives an direct error metric the closeness can also be established.

However looking at the data distribution especially for the EmoBank dataset which should perhaps be the most important one due to the much higher degree of labeled text, the data is very centered around neutral. This could potentially be a problem when evaluating the model using only these two metrics. The reason behind this is that a very simple model that only predicted values close to neutral (3 in the case of EmoBank) for valence, arousal and dominance would succeed in having low RMSE due to most labeled text having rating close to neutral. The model could also be created such that the r-score would be decent enough by only predicting V,A,D ratings close to neutral. It could also be argued that text with an emotional rating further away from being neutral could be more interesting to accurately predict, especially in movies where we would perhaps want to find interesting parts of a movie, which most likely won’t have a neutral emotional rating. Therefore for each of the predicted ratings for each weighting scheme a scatter plot showing the actual target values (Y-axis) and the predicted values (X-axis) will be added. A perfect model would then show a linear graph where (Y=X). This by inspection avoids the problem of a model that is only good at predicting text with neutral rating.
5.2.1 Analyzing movies

In terms of analyzing actual commercial movies it was decided to predict V,A,D values for the movie over time. That is detailing the emotional impact from the movie over the span of the movie. Just giving each movie a V,A,D rating is not sufficient to capture the essence of the movie and the different parts of the movie and their emotional content.

To be able to do this each movie needed to be split into parts with their corresponding subtitles and then sent through the model, where each subtitle part could work as one of the element passed to the model. It was first decided to split the movie based on the several scenes that appears in a movie, this since one scene usually fully details one incident in the movie. However since this model should work in an automatic fashion and be a part of a larger movie analyzing model this was not feasible. The reason behind this is that currently there are no decent enough software that automatically can parse a movie and finds its corresponding scenes. This would leave the only option remaining to be manually shifting through a target movie finding all its corresponding scenes which would be too inefficient. Instead we decided to split a movie into 1 minute parts, hereby referred to as a scene, with the subtitle part which was given a V,A,D score to show the emotional content of a movie over the movie duration. This would give, if a movie on average would be 1 hour and 30 minutes long, 90 ”scenes” that were given a V,A,D rating. By doing this interesting segment of the movie could be found. For example a segment with very high arousal would indicate a intense scene where the viewer is most likely engaged in the movie defining it as a important part of the movie for enjoyment, but other interesting patterns could also be found in a particular movie.

Another way to validate the model against movies was to look for similarities between movies. Movies are commonly remembered by a set of interesting scenes that happens during the movie, if by finding a interesting scene in a movie (in this case 1 minute scenes) we could look into finding a similar scene from another movie by using Manhattan distance for the three dimensions. The closest scene from another movie could then be manually determined if its content where similar to its comparison scene. This opening up not only being able to analyze one movie but also several movies together to find similarities between them. To do this one scene was picked and worked as a seed to generate the 9 closest scenes to it. The scene chosen was from Die hard, detailing when the police, Al Powell arrives at the Nakatomi building to check out the disturbance and then getting attacked by the terrorist. By analyzing the 9 VAD similar scenes we could determine if they actually are similar to the scene from Die hard. Which in that case
would indicate success for the model. In total 20 movies were used for this and they are shown below. These were movies which were available to me not only with subtitles but also the actual movies, which was needed to be able to manually validate how well the model performs.

1. Zootopia
2. Magnificent Seven
3. Nocturnal Animal
4. Whiplash
5. Gifted
6. The Revenant
7. Captain America - Civil War
8. 10 Cloverfield Lane
9. Collateral Beauty
10. Assassins Creed
11. Bridge of Spies
12. Deepwater Horizon
13. The Jungle Book
14. John Wick 2
15. Stardust
16. Land of Mine
17. Moana
18. Logan
19. Star Trek Beyond
20. Die hard I
6 Result

6.1 Small text results

The 2 datasets, EmoBank and movie text where used to validate the model with RMSE, r-score and scatter plots. Both datasets where changed to span V,A,D ratings from 1-9 to also be able to compare them with each other. That means the EmoBank which originally had ratings spanning from 1-5 was changed to instead span ratings from 1 to 9.

To be able to compare the model a random classifier was also used which simply picks a random score of 1-9 for each of the three dimensions for each text. Obviously our model should substantially outscore the random predictor to be of any use.

6.1.1 EmoBank text result

<table>
<thead>
<tr>
<th>Weight</th>
<th>R-V</th>
<th>R-A</th>
<th>R-D</th>
<th>R-Avg</th>
<th>RMSE-V</th>
<th>RMSE-A</th>
<th>RMSE-D</th>
<th>RMSE-Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>0.409</td>
<td>0.188</td>
<td>0.124</td>
<td>0.240</td>
<td>0.864</td>
<td>0.862</td>
<td>0.749</td>
<td>0.825</td>
</tr>
<tr>
<td>tf</td>
<td>0.408</td>
<td>0.185</td>
<td>0.122</td>
<td>0.238</td>
<td>0.866</td>
<td>0.867</td>
<td>0.752</td>
<td>0.828</td>
</tr>
<tr>
<td>tfidf</td>
<td>0.415</td>
<td>0.190</td>
<td>0.126</td>
<td>0.244</td>
<td>0.874</td>
<td>0.871</td>
<td>0.764</td>
<td>0.836</td>
</tr>
<tr>
<td>tfidf-es</td>
<td>0.439</td>
<td>0.200</td>
<td>0.140</td>
<td>0.260</td>
<td>0.890</td>
<td>0.906</td>
<td>0.792</td>
<td>0.863</td>
</tr>
<tr>
<td>Random</td>
<td>-0.009</td>
<td>0.008</td>
<td>-0.014</td>
<td>-0.005</td>
<td>2.737</td>
<td>2.671</td>
<td>2.666</td>
<td>2.691</td>
</tr>
</tbody>
</table>

Table 1: EmoBank dataset accuracy
Figure 7: The scatter plots for EmoBank
6.1.2 Movie text results

Here follows the accuracy measured on the self-gathered move text dataset with Table 2 detailing the accuracy on the control words, that is how our users rated compared to the data gathering in the Warriner dictionary. Table 3 show how well the model performed on the movie data set.

<table>
<thead>
<tr>
<th>Weight</th>
<th>R-V</th>
<th>R-A</th>
<th>R-D</th>
<th>R-Avg</th>
<th>RMSE-V</th>
<th>RMSE-A</th>
<th>RMSE-D</th>
<th>RMSE-Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>0.716</td>
<td>0.451</td>
<td>0.551</td>
<td>0.573</td>
<td>1.092</td>
<td>1.496</td>
<td>1.139</td>
<td>1.243</td>
</tr>
</tbody>
</table>

Table 2: Control words accuracy

<table>
<thead>
<tr>
<th>Weight</th>
<th>R-V</th>
<th>R-A</th>
<th>R-D</th>
<th>R-Avg</th>
<th>RMSE-V</th>
<th>RMSE-A</th>
<th>RMSE-D</th>
<th>RMSE-Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>0.328</td>
<td>0.372</td>
<td>0.017</td>
<td>0.239</td>
<td>1.684</td>
<td>1.339</td>
<td>1.713</td>
<td>1.579</td>
</tr>
<tr>
<td>tf</td>
<td>0.324</td>
<td>0.364</td>
<td>0.022</td>
<td>0.237</td>
<td>1.689</td>
<td>1.346</td>
<td>1.713</td>
<td>1.583</td>
</tr>
<tr>
<td>tfidf</td>
<td>0.331</td>
<td>0.365</td>
<td>0.034</td>
<td>0.243</td>
<td>1.677</td>
<td>1.355</td>
<td>1.724</td>
<td>1.582</td>
</tr>
<tr>
<td>tfidf-es</td>
<td>0.353</td>
<td>0.341</td>
<td>0.048</td>
<td>0.247</td>
<td>1.669</td>
<td>1.385</td>
<td>1.724</td>
<td>1.593</td>
</tr>
<tr>
<td>Random</td>
<td>-0.114</td>
<td>0.0241</td>
<td>-0.131</td>
<td>-0.074</td>
<td>3.171</td>
<td>2.924</td>
<td>3.095</td>
<td>3.063</td>
</tr>
</tbody>
</table>

Table 3: Movie dataset accuracy
<table>
<thead>
<tr>
<th>Weight</th>
<th>Valence</th>
<th>Arousal</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
</tr>
<tr>
<td>tf</td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
</tr>
<tr>
<td>tfidf</td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
</tr>
<tr>
<td>tfidf-es</td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
<td><img src="image" alt="scatter plot" /></td>
</tr>
</tbody>
</table>

Figure 8: The scatter plots for Movie small texts
6.2 Movie results

6.2.1 Single Movie

Figure 9 details the VAD variation for the movie Die hard split into 1-minutes scenes for the duration of the movie.

![Figure 9: The VAD score for the movie Die hard](image)

While specific scenes might be hard to see in this figure, general interesting patterns can be found. For example the large spikes of the dimension Intensity and low Dominance and Valence at the end details the hostage scene where Hans has taken Holly as a hostage. From this plot interesting segment of the movie can be seen.

6.2.2 Comparison of movie scenes

This section show the scene from Die hard and its 9 closest scenes from other movies. Table 4 gives an insight into which scenes were found, their dimensional ratings and a short description of what is happening in the scene. Table 5 extends on the other table to show which words were found in each scene and which where parsed for dimensional ratings. Finally a complementary picture detailing Die hard and its closest scene can be seen in Figure 10.
<table>
<thead>
<tr>
<th>Movie</th>
<th>Time Frame</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Die Hard</strong></td>
<td>0:56:55 - 0:57:55</td>
<td>V = 3.85, A = 6.57, D = 4.51</td>
<td>The Police, Al Powell is sent to check out the disturbance at the Nakatomi building. When driving away a body falls on his car and the terrorist start shooting at him. He then tries to flee from the building while calling for backup.</td>
</tr>
<tr>
<td><strong>The Revenant</strong></td>
<td>0:11:04 - 0:12:04</td>
<td>V = 3.85, A = 6.56, D = 4.55</td>
<td>The Americans are getting attacked by the natives. The natives are shooting arrows. The Americans are firing back while trying to flee to their boat. Scene ends with them getting on the boat.</td>
</tr>
<tr>
<td><strong>The Magnificent Seven</strong></td>
<td>1:45:18 - 1:46:18</td>
<td>V = 3.88, A = 6.60, D = 4.55</td>
<td>The Magnificent seven are defending a village against a group of people. One of the seven arrives at horse and yells that they enemy has a gatling gun. The run to hide while the gatling gun start firing.</td>
</tr>
<tr>
<td><strong>The Magnificent Seven</strong></td>
<td>1:46:18 - 1:47:18</td>
<td>V = 3.81, A = 6.52, D = 4.48</td>
<td>Continue of previous scene, the gatling gun keep firing while people run to safety. The man on the horse keep warning people to get inside a building to hide. People dies left and right from the firing gatling gun.</td>
</tr>
<tr>
<td><strong>Logan</strong></td>
<td>0:34:00 - 0:35:00</td>
<td>V = 3.83, A = 6.48, D = 4.45</td>
<td>Logan and Professor X are surrounded inside their car by people with guns. Logan confronts the main bad guy asking him what they did to his friend. The bad guy tells him he killed him and Logan tries to attack him. Logan gets beaten up by the bad guys.</td>
</tr>
<tr>
<td><strong>Moana</strong></td>
<td>1:25:55 - 1:26:55</td>
<td>V = 3.74, A = 6.63, D = 4.50</td>
<td>Maui fights with the fire monster while Moana tries to get away to an island. Maui transforms himself to different animals but eventually gets beaten. The fire monster launches a fire ball towards Moana but it’s absorbed by the water.</td>
</tr>
<tr>
<td><strong>Nocturnal Animals</strong></td>
<td>1:16:25 - 1:17:25</td>
<td>V = 3.87, A = 6.54, D = 4.33</td>
<td>Tony and Bobby (the cop) is driving with Ray (the criminal) held at gun point. They drive to a cabin where they confront Ray about raping and killing Tonys wife. Ray acts innocent saying he didn’t rape anyone.</td>
</tr>
<tr>
<td><strong>Captain America Civil War</strong></td>
<td>0:07:01 - 0:08:01</td>
<td>V = 3.73, A = 6.50, D = 4.54</td>
<td>Captain America and team tries to interrupt Crossbones who has a biological weapon. Natasha fights her way through his gun armed lackeys to get to the weapon. She gets caught by Crossbones and thrown into a car.</td>
</tr>
<tr>
<td><strong>Whiplash</strong></td>
<td>0:17:20 - 0:18:20</td>
<td>V = 3.67, A = 6.63, D = 4.52</td>
<td>Fletcher introduces Andrew to the band while calling him young. Fletcher then makes the band start to play Whiplash.</td>
</tr>
<tr>
<td><strong>Nocturnal Animals</strong></td>
<td>1:30:25 - 1:31:25</td>
<td>V = 3.98, A = 6.64, D = 4.46</td>
<td>Bobby is holding Ray and his companion at gun point in a cabin. Bobbly then gives the gun to Tony to kill Ray and his companion since they raped and killed his wife and daughter. Tony hesitates, Bobby start coughing up blood and has to leave the room. Tony can’t kill them so they escape from the cabin.</td>
</tr>
</tbody>
</table>

Table 4: Details of the Die Hard scene and the 9 closest scenes to it from other movies.
Table 5: Same scenes as in table 4 but instead showing the words found for each scene

<table>
<thead>
<tr>
<th>Movie</th>
<th>Words Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die Hard</td>
<td>wild goose chase plaza damn goddamn welcome party pal goddamn mean goddamn policeman automatic rifle fire need backup assistance goddamn</td>
</tr>
<tr>
<td>The Revenant</td>
<td>fucking savage get fucking boat help cut rope cut rope wait go fire cannon leave wait</td>
</tr>
<tr>
<td>The Magnificent Seven</td>
<td>go block come devil breath goddamn gun let run get fuck damn</td>
</tr>
<tr>
<td>The Magnificent Seven</td>
<td>take cover go miss get gun gun get go gun gun move get way go get get</td>
</tr>
<tr>
<td>Logan</td>
<td>tell girl first ask bald seem quit friendly motherfucker ditch leave wolverine see like break damn heart rip chest fuck go get</td>
</tr>
<tr>
<td>Mosana</td>
<td>shark head</td>
</tr>
<tr>
<td>Nocturnal Animals</td>
<td>place know place honest god get shall look come let take look right right rape bed suppose rape stop fucking record know girl drop rape know</td>
</tr>
<tr>
<td>Captain America Civil War</td>
<td>biological weapon radio screech grunt groan</td>
</tr>
<tr>
<td>Whiplash</td>
<td>people nineteen year old cute right gang whiplash</td>
</tr>
<tr>
<td>Nocturnal Animals</td>
<td>cry cry feeling uncomfortable get take go right fuck right right right feeling uncomfortable come go free bird say shall two son come come let go gun shoot shoot come back little fucker</td>
</tr>
</tbody>
</table>

Movie scenes

Figure 10: Detailed view of the scene from Die Hard and the closest scene to it from The Revenant
7 Conclusion and Discussion

In terms of the data gathering, two datasets were collected. The own gathered data set over movies sentences did as mentioned include ratings from 9 different users. This is unfortunately not a lot of people which has the risk of making the data not representative. It does however give an indication when used as a validation set and together with the other larger dataset, conclusions can still be drawn about the performance of the created model.

The problem with the EmoBank dataset in terms of my task is that is not really related to the movie domain in particular and instead includes rather general text from different genres. The data distribution of the EmoBank set is also not optimal for testing how well the model performs with very non-neutral affective ratings, that is a very high V,A,D or low V,A,D. That because there is still too few data points in those domains.

Looking at the self made data gathering, control words were added to measure how well the 9 users rated and understood the three affective categories. Control words which already had a rating in the Warriner dictionary which came from a larger and more robust study. Looking at the Pearson correlation from these control words compared to their values in the Warriner dictionary, the average Pearson correlation for the 9 users is pretty good and actually corresponds with previous differences in cross-studies [10] Which at least indicate that the users where able to understand the different ratings and how they are used and therefore the actual movie sentences they rated should be viable to use.

Looking at the result of the model on the test set with labeled affective scores, compared to the random scoring, our model substantially outscores the random scoring. This both in terms of Pearson correlation but also in the RMSE score. This regardless of which weighting scheme used or data set. Our model performs the best on the EmoBank dataset in terms of average RMSE, and performs similar in terms of average Pearson correlation on both datasets with the interesting difference of a very low Pearson correlation for the movie text dataset regarding dominance which is even outsored by the random model, but with a higher arousal r-score than for the EmoBank dataset. While I argue that the EmoBank is a better dataset compared with the movie dataset to be used to validate the model due to its extensive size and a higher amount of people who contributed to give the VAD score it still introduces the problem of having a data distribution with a high neutral score. This could explain some of the much lower RMSE that can be seen for the EmoBank dataset compared to the movie dataset. This is also the reason I introduced scatter plots to be able to also determine the effect of the different weighting schemes on the models performance. Regardless of
weighting schema used for the model the performance in terms of Pearson Correlation and RMSE is similar. In the scatter plots you can however see the problem of the average weighting schema where it classifies more values closer to neutral than the others. The TFIDF-ES is the best performing scatter plot in terms of replicating the \( Y = X \) and was therefore chosen for analyzing movie subtitle content.

Taking a look on how the model performed for movies, that is the created affective time lapse over a movie but also how well similar scene based on affective content can be found from a seed movie. The model actually creates some interesting patterns. Looking at the time lapse for the movie Die hard as seen in Figure 9 it details the movie well in terms of what is happening. I noticed that values for Valence and Arousal actually corresponded well to what was going on in the movie while the third domain Dimensional was harder to defer. For example scenes with low valence but high arousal usually detailed a fighting scene between the protagonist and one or more terrorist. This was also seen in other movies where arousal and valence values showed some properties of the scene in hand. Worth noting however is that the Movie Die hard includes a protagonist, John McClane who is very vocal and uses a lot of profanity even when fighting. This could mean this movie is better than others for my model to determine affective content. Therefore not to strong conclusions should be based only on this movie and the models performance but instead additional studies including more movies should be done.

From the Die hard scene seed most of the closest scenes from other movies are related and heavily reference the protagonists getting attacked or being in very stressed environments. The main divergent scenes are the Whiplash scene and the Moana scene which isn’t as similar to the Die hard scene as the rest. The Moana scene is also from an animated Disney movie whose content is made for children unlike the others and my model only found 2 words to use. This is however not something my model takes into considerations when processing for affective content. The Die hard scene however also includes some comedic elements which is not captured and not seen in the other scenes, I would however argue that these are not captured in the subtitles but rather seen through visual clues. But they still show a limitation of only having a model determine affective content through text.

Due to the nature of emotions and the vague definition of emotions its hard to quantify the exact impact my model has in the emotional classification domain. Few studies have also been done in terms of dimensional affective content labeling making it difficult to compare my model to a baseline. I do think however that my model shows some potential in labeling text which can be shown when looking at movies but also when comparing movie
scenes. Where the result is actually able to gather information about the content of a movie, labeling its affective content in a way that corresponds with the movie but also finding similar scenes from different movies. However as of the current state when processing movies I’d argue my model would work best as a fine-graining step for selection of similar movies or finding user patterns. Since my model disregards genres it has the unfortunate potential to group together movies which caters to different age groups. By instead having more genre based algorithms such a collaborative filtering as a first step selecting a set of similar movies, my model could be utilized to more closely find similar movies from that set by looking at the affective content and how it varies during a movie. Also extending my model with the same done for audio and video I believe to be necessary and can prove to become a powerful fine-graining step to group movies together and give users better suggestions for what to watch. This is something I think the streaming services today lack and incorporation of efficient labeling of affective content is a good step to extend the services with.
8 Future Work

This project only touches on the surface of what is classification of affective content. To be able to increase the classification of affective content for text more dataset needs to be created. If the VAD approach is taken datasets with more data points with a non-neutral score would be important to make available. With more data available a more advanced machine learning model could be created. Currently the automatic text processing is a hot topic and several models utilizing neural networks have shown to get good performance [26]. A neural approach could potentially be incorporated to increase the accuracy of the affective classification.

In terms of performance on movies and the validation of the model thereof, studies into emotional reactions from movies need to be done. This allows validation on a larger scale and more training data to be used. One could also look into how different genres in movies differ in emotional content over time and perhaps be able to find a generic genre pattern on emotional content. Building on a larger scale the potential to incorporate emotional classification in movies not only based on the subtitles but also from the audio and the visual content could create a better and more substantial model. From this user patterns can be analyzed by looking at the emotional content of a movie. Perhaps certain user groups can be found which prefers movies with a specific emotional variance, for example user groups who prefer low intense movies or the opposite. Movies could then be suggested based on a particular user group to create a better service.
Acknowledgements

I would like to thank everyone at VionLabs, the company I did my thesis on, for welcoming me in a really good way and helping me all the way trough my project. Special thanks to Arash and Adren for helping me with my project and the structure of it. I’d like to thank Yuqing Li for being the best rater for my data gathering and rating a large set of movie sentences.

A big thanks to my university supervisor Kristiaan Pelckmans for helping me structure the project and providing insightful ideas on how the approach the affective labeling problem.

Finally I’d like to thank my family and my girlfriend for their support and help during my thesis.
References


