university of groningen / faculty of arts

Sentiment Analysis on Movie Reviews

A comparison of Feature sets

Tom Kosse

Bachelor thesis
Informatiekunde
Tom Kosse
s2002167
June 5, 2017

ABSTRACT

Finding features in a movie review can be as difficult as finding a needle in a haystack. When it comes to sentiment classification on movie reviews, it is difficult to predict on which features a reviewer will express sentiment. This paper uses an ad-hoc approach to feature analysis: features are tested based on the number of times they occur in the review. The domain-specific features investigated include: actors, directors, genre and movie titles. Classification is performed by using both a Naive Bayes classifier and a Linear SVC classifier.

After performing classification, the results show that non-specific features perform far better than movie-specific features. More specifically, 89.2% accuracy is achieved by the 'text' feature when using bigrams and a Naive Bayes classifier. Only 63.2% accuracy is achieved by the movie-specific features when using unigrams and the Linear SVC algorithm. This score is achieved by combining all four domain-specific features together. Overall, unigrams seem to outperform both bigrams and trigrams. Also, the Linear SVC seems to outperform the Naive Bayes classifier in most instances.

A deeper analysis of the domain-specific features is also provided in this research; here we analyze why these features have less predictive power than expected. We find that if relative occurrence is taken into account rather than absolute occurrence, this should greatly improve predictive power. For movie titles, we find that if IMDb scores are taken into account, this should also have a beneficial effect on the predictive power.

CONTENTS

PREFACE

First and foremost, I would like to thank my supervisor Leonie Bosveld-de Smet for helping out a great deal during the production of this thesis. We sat down several times to discuss the shape of the research and the manner of interpretation of the results. Secondly, I would like to thank my lovely girlfriedn Sarah jenicek for supporting me throughout the writing of this thesis. She also helped a great deal with reviewing the text and tweaking it as much as possible. Also, I would like to thank Kevin Markham from the YouTube-channel Data School. His online courses and explanations have helped me a great deal at the start of this experiment. Last but not least, I woul like to thank my computer for having great processing power and not giving up on me during the production of this research.

1 INTRODUCTION

In the spectrum of sentiment analysis, movie reviews are as we say, a 'tough nut to crack'. Most types of reviews have a clear set of aspects on which the reviewer expresses sentiment. Intuitively, it is hard to determine what these aspects are when it comes to movie reviews. This paper proposes an ad-hoc approach to sentiment analysis on movie reviews, where occurrences of certain aspects are used as features. In essence, this research is a feature analysis, where different feature sets are tested for their predictive power.

According to Choi et al. (2009), sentiment analysis is the task of extracting positive or negative aspects from a free text. Sentiment analysis is also called opinion mining, sentiment mining, opinion extraction or review mining, to name a few (Liu, 2012). While performing sentiment analysis, machine learning is used to train a system and make predictions about unseen instances. In the case of movie reviews, the classes will be either positive or negative. Thus, this is a binary sentiment classification task. Our research question is:

Which feature sets work best when performing sentiment classification on movie reviews, using a supervised learning model?

With this research question, a full analysis of all the different feature sets is performed. There will be a complete overview of all the different features and their performance in this sentiment classification task. To provide for a thorough feature analysis, the main question has been divided into the following subquestions:

1. What feature or combination of features perform(s) best overall?;
2. Which movie-specific features perform best overall?;
3. Which classification algorithm performs best overall?;
4. Why do the movie-specific features yield the results that they do?.

The following report will be divided into four chapters in order to answer the main research question. In the first chapter, a complete theoretical background is given to introduce the reader to the subject. Also, previous research and results are examined and we will show how this investigation is related to previous work.

In the second chapter, the dataset that has been used for the experiment is discussed. A complete overview of the collection is given, including a graph to allow for better comprehension. The third chapter will discuss which algorithms, features and techniques are used for performing the task at hand. There is a section for the classification algorithm and a section for the feature analysis algorithm, which allows the features to be examined on a deeper level. In the final chapter, the same sections as in the previous chapter are included. In the classification section, the classification results are presented and interpreted. In the feature analysis section, movie-specific features are analyzed more in-depth to investigate why they performed in their specific manner.

2 BACKGROUND

In this chapter, the background of the investigation will be outlined. For the reader to become familiar with the subject, the first section contains the theoretical framework in which this research is embedded. In the second section, previous and similar research is discussed, which can also be seen as an overview of the current state of play. In the last section, the purpose of this paper is provided, where we discuss how this investigation contributes to existing research in the domain of sentiment analysis on movie reviews.

2.1 THEORETICAL FRAMEWORK

As mentioned briefly in the introduction, this investigation shall be done by means of machine learning. As Mohri et al. (2012) explain well in their work, there are several facets to the spectrum of machine learning, where different kinds of data are needed to perform the classification. Within this investigation, supervised learning will be used to make predictions about unseen instances, where annotated data is used to train a classification model and make predictions about sentiment in a text. In our case, this text will consist of movie reviews. There are three levels on which machine learning can be utilized: document-level, sentence-level and aspect-level (Manek et al., 2016). In this investigation we will focus on the aspect-level, where we will analyze different aspects, or features, of a movie review.

There are also several types of classification algorithms available, with which we can execute the investigation. While we will not delineate how they work, a quick overview will be given of the classification algorithms that are used for this investigation. The reader will find that there are results of two classification algorithms in this paper: the Naive Bayes algorithm and the Linear Support Vector Classification. The Naive Bayes algorithm was first introduced in the 1960's, and has since been widely used by information scientists to perform text categorization (Russell and Norvig, 1995). It is often used as a baseline method in modern research, where it can still perform very well in binary classification tasks with word frequencies as its features.

The Linear Support Vector Classification (Linear SVC) algorithm is a form of Support Vector Machine that uses a linear kernel to perform its classification task. Support vector machines have a remarkably robust performance with respect to sparse and noise data, making it incredibly powerful for text classification purposes (Furey et al., 2000). The Support Vector Machine algorithm was first introduced in 1963, but has since been tweaked numerously by researchers and developers. The Linear SVC that is used in this investigation was introduced by Cortes and Vapnik (1995).

In light of the types of features that are investigated in this paper, a brief explanation of n-grams is in order: this type of feature is used a lot in computational linguistics, and it represents a sequence of n items from a given text. These items can include many different types of sequences, but the n-grams that are used in this investigation will always apply to a sequence of words. When n-grams refer to sequences of words, these can also be called 'shingles' (Broder et al., 1997). In this investigation, one will see unigrams, bigrams and trigrams being used for classification purposes. These definitions apply to sequences of one, two and three words respectively.

2.2 SIMILAR RESEARCH

Sentiment analysis on reviews is a widely studied subject, where product reviews are often seen as the baseline since the features are usually clearly defined. Products always have certain attributes, for which sentiment can be derived from the review. In the paper by Dave et al. (2003) features are identified by means of linguistic extraction of product attributes. To achieve this they use parsing methods and metadata, with a baseline of word frequencies. They attempt to perform classification on the sentence level, but they state that performance is limited due to noise and ambiguity.

In our specific domain of movie reviews, different kinds of classification methods and feature sets have been attempted over the last several years. Pang et al. (2002) performed sentiment classification on movie reviews, but without applying any domain-specific features. They found that machine learning outperforms the human-produced baselines by about 20% accuracy. Turney (2002) found movie reviews to be the most difficult on which to perform sentiment classification in the entire domain of review classification, due to the fact that positive and negative reviews both often describe plots and scenes.

Additionally, Chaovalit and Zhou (2005) stated that the challenges of movie review mining are that factual information is always mixed with real-life review data and ironic words are used in writing movie reviews. They make a comparison between semantic orientation and machine learning techniques, both supervised and unsupervised. They found that supervised machine learning outperformed both unsupervised machine learning and semantic orientation methods.

Furthermore, Anand and Naorem (2016) argue that a major part of movie reviews is devoted to describing the plot and yields no usable information for classification. They attempt to perform classification using aspect clue words, which are words associated with a certain feature, which they manually compiled. For instance, they have a list of words that are associated with an actors' performance and a list of words associated with the story of the movie. The results show that manually selecting the features outperforms clustering methods.

In Zhuang et al. (2006) they argue that summarizing a review might be the best way to perform sentiment analysis in movie reviews. They accomplish this by parsing the text and extracting feature-opinion pairs giving a certain sentiment on a feature. With this information, they performed classification and found that this type of feature selection performs rather poorly, with only marginal gains on the baseline.

In this investigation, a more ad-hoc approach to sentiment analysis of movie reviews will be performed. This paper attempts to perform the classification based on occurrences of features, rather than opinions expressed about certain aspects of a movie. As seen above, aspects of movie reviews are hard to derive from the text used by the author of the review. Hence, we take the approach of counting occurrences of features such as actor names or mentions of genre. This way, we will determine whether the mention of a certain actor is an indication of that review being either positive or negative.

This work is thus based on feature frequencies, rather than sentiment expressed on certain features. The results of this research will provide new insights into how features can be used to perform sentiment classification on movie reviews. Combining the ad-hoc features used in this investigation and features used in previous investigations that use parsing and semantic structures, a new and better baseline can be provided for performing similar classification tasks on movie reviews.

3 DATA AND MATERIAL

3.1 COLLECTION

The data collection that has been used for the classification task is the Large Movie Review Dataset. This is a pre-existing dataset that was compiled by Maas et al. (2011) for their research about learning word vectors for sentiment analysis. The dataset is compiled from reviews that exist on the IMDb website, where users can write reviews for movies. The dataset contains 50.000 movie reviews with their associated binary sentiment polarity labels (more on annotation in the next section). The dataset is divided evenly into a train and a test set, each containing 25.000 reviews. Within the dataset there are also an additional 50.000 unlabeled movie reviews, which are not used within the scope of this research. In the entire collection, no more than 30 reviews are allowed for any given movie, since reviews for the same movie tend to have similar ratings. In Figure 1 there is a visualization of the structure of the entire dataset.

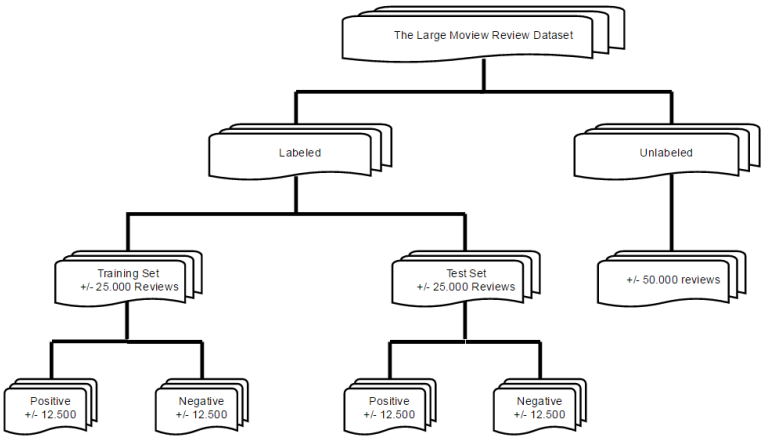

Figure 1: The Large Movie Review Dataset.

The reviews in the dataset are stored as text files, where each file has a unique ID and an associated rating within the title of the file. The file itself contains the review in raw text format, which means that the text in the file is exactly as the reviewer wrote it on the IMDb website. Because of this, there could be some noise since the reviewer might have misspelled words or made use of an unconventional way of punctuation. This has to be taken into account when evaluating the results of classification presented in this papar.

To collect the movie-specific features, the IMDb 5000 Movie Dataset was used. This is a dataset that contains information about more than 5000 distinct movies, including their title, genre, director and top 3 actors. The features were gathered from this dataset and saved as lists in text-files to be used as feature vectors during the classification task. The IMDb 5000 Movie Dataset was gathered by using a Python scraping tool to scrape the IMDb website for the information. In the dataset there are 5043 movie titles, 2400 director names and thousands of actor names. Altogether, this dataset provides us with quite an extensive collection of feature sets.

3.2 ANNOTATION

The Large Movie Review Dataset was compiled by Maas et al. (2011) as an annotated dataset, which means that the reviews were gathered based on their ratings. The reviewer can rate the movie on a 1 to 10 scale, giving an indication of the opinion of the reviewer about the movie. These ratings are used for the annotation of the dataset, where a rating of 4 or lower is considered negative and a rating of 7 or higher is considered positive. As a result, we obtain a polarized dataset that only contains reviews that have either a convincing negative or a convincing positive rating. The binary polarized nature of the dataset makes it extremely suitable for binary classification purposes. The train and test are divided evenly, with 12.500 positive reviews and 12.500 negative reviews per set.

3.3 PROCESSING

Since the reviews all contain raw text files, some pre-processing is needed before the classification task can be performed. The first step in the processing of the data is the creation of a pandas dataframe within Python. Pandas is a library in Python which makes it possible to create dataframes which can be interpreted by classification algorithms. The created dataframe contains the text of every review and the label of every review, with the unique ID of the file as the index.

The second step is to perform feature extraction, where all the different features are extracted from the text of the movie reviews. The results are stored in the same dataframe as mentioned above. To extract a feature from the text, a text file is used in which all the keywords for every feature set are stored. As an example, the file for the 'genre' feature set contains words like 'action', 'adventure' and 'horror'. On every new line in the text file, one of these words is stored. In Figure 2 the first few rows of the created dataframe can be seen. There is a column for every feature, and in every row the information that is extracted per feature for a given movie review.

	Label	Text	Sentiment	Actors	Directors	Genre	Titles
File							
0_3	negative	Story of a man who has unnatural feelings for ...	absurd, bs, crazy, cry, die, err, insane, irk,...	sally kirkland, frederic forrest		comedy	
10000_4	negative	Airport '77 starts as a brand new luxury 747 p...	ail, anger, bad, badly, bland, boring, crash, ...	kathleen quinlan, robert foxworth, christopher...	jerry jameson	action, adventure, thriller, war	the doors
10001_4	negative	This film lacked something I couldn't put my f...	din, ding, dire, disappoint, disappointed, exp...				anything else, the prince
10002_1	negative	Sorry everyone,,, I know this is supposed to b...	absence, blow, brutal, bs, complain, complaini...				
10003_1	negative	When I was little my parents took me along to ...	babble, boring, contempt, crap, damage, dark, ...	woody allen	ingmar bergman	drama, family, music	

Figure 2: An etract from the pandas dataframe.

The algorithm loops through all the different feature lists and extracts all the features from the given reviews. Once this has been completed, the created dataframe can be used to fit to a classification algorithm. Because all the features are stored in the same dataframe, combining different feature sets is very convenient. The next chapter explains how this process works in more detail. When the text and the features are stored in one dataframe, the pre-processing of the raw data is complete.

4 METHOD

As discussed in the second chapter, this research will focus on evaluating the performance of different feature sets for movie review classification. To this end, several different feature sets have been gathered. Essentially, there are two types of feature sets that are considered in this paper, namely domain-specific and non-specific features. The non-specific features in this paper are *n*-grams and a sentiment lexicon. The domain-specific (movie-specific) features are actors, directors, genre and movie titles. The investigation is divided into two parts: classification and feature analysis. In the classification part of the research all the different feature sets will be tested and the results evaluated. In the feature analysis part, an in-depth look into the movie-specific features will be given to study exactly why these feature sets yield their specific results. To give a short overview, these are all feature sets that will be evaluated during this investigation:

1. *N*-grams (the text in the review);
2. Sentiment words (polarized words that indicate a certain sentiment);
3. Actors (names of actors that are mentioned in the review);
4. Directors (names of directors that are mentioned in the review);
5. Genre (mentions of genre in the review);
6. Movie titles (mentions of movie titles in the review).

As mentioned in the previous chapter, the movie-specific features are all extracted from the IMDb 5000 Movie Dataset. There is a small overlap between the list of actors and the list of directors, since some actors of movies were also directors for other movies in the dataset. For this reason, these two feature sets were made mutually exclusive to make the classification results easier to interpret. For every person that was mentioned in both the actors and the directors feature set, he or she was removed from the directors list. We decided to treat the data in this way since it is more common for an actor to start directing at some point in his career than the other way around. Thus, we made the naive assumption that people that appear in both lists are more likely to be an actor first and a director second.

4.1 CLASSIFICATION

In this part of the investigation, all the different feature sets will be tested separately. After this, the movie-specific feature sets will also be combined to investigate if this leads to an increase in performance. It might be the case that certain feature sets perform better when combined with each other than just on their own merits. To determine how well the different feature sets perform we will analyze the results with the use of four different evaluation scores. The most intuitive of these scores is accuracy, which indicates how many of the predicted reviews are predicted to be the right class. This is possible due to the supervised learning nature of this experiment. The F1-score is another good measure of performance in classification, since this is the harmonic mean of precision and recall. The higher the accuracy and F1-scores are, the better a certain feature set performed in predicting the right class for the review.

Because this research also takes n-grams into account, all the feature sets are tested three times. As can be seen in the theoretical framework, we consider unigrams, bigrams and trigrams in this paper. For the classification algorithms, we also use two different types: the Naive Bayes and the Linear SVC. Thus, in total, we performed the classification experiment six times, with three different n-grams and two different classifiers. Hence, we will not only examine which feature sets perform best compared to each other, but also which feature sets perform best when used in a certain classification algorithm. This gives us a more complete overview of the performance of feature sets for movie reviews.

To visualize the classification process, a flow chart for the algorithm used to perform this task has been included below. As can be seen in Figure 3 the algorithm takes a set of documents as its input. It then extracts the needed information (text and features) and performs classification on them with both a Naive Bayes and a Linear SVC algorithm. The accuracy, precision, recall and F-scores are automatically computed and reported by the algorithm. After this, certain features can be combined and the classification will again be performed on these new combinations of feature sets.

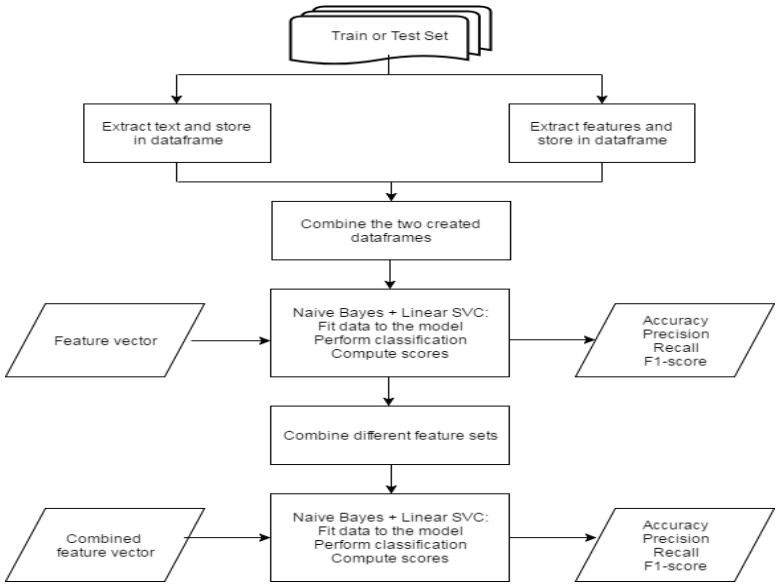

Figure 3: Flow chart of the classification task.

The goal of the feature sets is to outperform the baseline, which we have set at 50% accuracy for this experiment. This baseline comes from randomly guessing, which yields 50% accuracy. This is due to the fact that it is a binary classification experiment, where we have two possible outcomes: positive or negative. Since the data collection is evenly divided into these two categories random guessing would yield an accuracy score of around 50%.

4.2 FEATURE ANALYSIS

Since we want to have a deeper understanding of the results of movie-specific features, a more in-depth analysis will be done in this section of the investigation. To do this, we will count the occurrences for all keywords in each feature set, in both the positive and negative reviews. After this, we can closely study which actors, directors, genres and movie titles are mentioned in positive and negative reviews respectively. With this information, it is possible to provide better insights into the feature vectors that are used by the classifier to make predictions.

Since this part of the investigation has no relation to the classification task itself, both the training and the testing data is taken into account when counting all the different features. This way there is more data on which to draw conclusions.

An algorithm has been developed which can be seen beloew in Figure 4. When all occurrences of a certain feature set have been counted, the features that have less than five occurrences will be taken out of consideration. Once this is done, there are three ways of analyzing the data more closely.

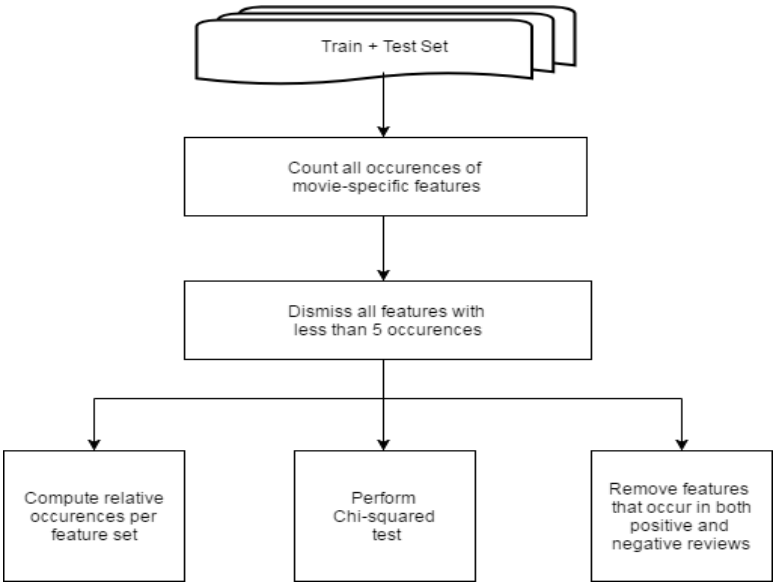

Figure 4: Flow chart of the feature analysis.

One of these is the relative occurrences of a certain feature. For some features there might be more overall occurrences in positive or negative reviews. To make proper comparisons between positive and negative reviews, the data will be normalized by computing their relative occurrences compared to the overall total.

Secondly, we can perform the Chi-squared test to test whether the class of a review is dependent on a certain feature. For this concept version, we have not yet succeeded in performing this in the right way. Hopefully, we will get this right for the final version of the thesis.

The last thing that can be done to provide more insight in the data is excluding features if they appear in both positive and negative reviews. This way, we obtain lists of movie-specific features that appear exclusively in positive or negative reviews. This information might be useful in feature classification tasks on movie reviews.

To have a better understanding of the scripts included with this paper, two examples have been included in the appendix. In Figure 5 an example of the classification code can be seen, and in Figure 6 an example of the feature analysis code.

5 RESULTS AND DISCUSSION

5.1 EXPECTATIONS

Before discussing the results of the classification task and the feature analysis, the expectations that we had before the experiment need to be considered. For the classification algorithm, we would expect that the Linear SVC algorithm would outperform the Naive Bayes algorithm in every way, since Linear SVC is a lot more robust to noise in data. Also, the *n*-grams and sentiment lexicon are expected to perform well, especially for unigrams and bigrams.

For movie-specific features like actors and directors, the bigrams should perform better than the unigrams and trigrams, due to the fact that names usually consist of two words. Overall, the movie-specific features are expected to perform fairly well, since certain genres or certain actors/directors are assumed to be a sign of a film being of higher quality. If a 'good' actor like for instance Denzel Washington is mentioned in a review, this could be a good indication of a positive review since Denzel Washington usually stars in highly rated movies. Similarly, if the genre 'horror' is mentioned this could be an indication of a negative review, since horror movies are generally seen as worse movies than drama films, as an example.

5.2 CLASSIFICATION

During the development phase of the investigation different train-test split parameters have been tested to see what percentage of training versus testing data performed best overall. In most cases, a situation where 90% of the data is used for training and 10% of the data is used for testing is ideal. Therefore, all the results that are provided in this chapter were produced under this train-test ratio.

Within this chapter, we provide two different tables containing the results for unigrams (Table 1) and bigrams (Table 2) for both classification algorithms, containing all tested feature sets. The trigrams performed worse compared to the unigrams and bigrams, so will therefore not be discussed here. The results for the trigrams can be found in Table 6 in the appendix.

As can be seen in Table 1 containing the unigram results, some features outperform the baseline by a large margin. Overall, for nearly all features except the sentiment lexicon, the Linear SVC algorithm outperforms the Naive Bayes algorithm with a margin of about 2-5%. The highest accuracy and F1-score comes from the 'text' feature performed by the Linear SVC algorithm. The sentiment lexicon is the only feature that performs better with the Naive Bayes classification, by a margin of about 2-3%. This is a peculiar result since we would expect all features to perform better with the Linear SVC as mentioned above, due to its robustness to noise.

The movie-specific features do not perform as well as we expected. However, if we combine all the movie-specific features together the accuracy score is about 13% above the baseline. Each time another movie-specific feature is added, the results become slightly better. Despite this, adding all the movie-specific features to the sentiment feature has a negative effect on the overall performance.

For the bigrams results in Table 2, the most noticeable is the high score for the 'text' feature in the Naive Bayes classification, which is close to 90% accuracy and F-score. This time, Naive Bayes outperforms the Linear SVC on this feature, and it also outperforms its unigram equivalent. This is also the highest accuracy and

Table 1: Unigrams

Naive Bayes Features	Accuracy	Precision	Recall	F1-score
Text	85.6%	84.4%	87.8%	86.1%
Sentiment	85.2%	86.0%	84.6%	85.3%
Actors	54.0%	68.0%	17.0%	27.2%
Directors	51.2%	64.9%	7.9%	14.1%
Genre	53.7%	57.1%	34.2%	42.8%
Titles	54.8%	62.7%	26.4%	37.2%
Actors + Directors	54.9%	67.2%	21.3%	32.4%
Actors + Directors + Genre	58.6%	62.8%	44.5%	52.1%
Actors + Directors + Genre + Titles	61.0%	64.9%	49.8%	56.4%
Actors + Directors + Genre + Sentiment + Titles	85.2%	85.2%	85.5%	85.4%
Linear SVC Features	**Accuracy**	**Precision**	**Recall**	**F1-score**
Text	89.1%	89.4%	89.1%	89.2%
Sentiment	82.9%	83.4%	82.8%	83.1%
Actors	58.4%	55.3%	92.5%	69.2%
Directors	54.6%	52.8%	97.0%	68.4%
Genre	55.4%	54.9%	67.0%	60.3%
Titles	56.9%	55.4%	75.8%	64.0%
Actors + Directors	60.3%	56.7%	91.1%	69.9%
Actors + Directors + Genre	62.4%	60.0%	77.4%	67.6%
Actors + Directors + Genre + Titles	63.2%	61.6%	72.8%	66.7%
Actors + Directors + Genre + Sentiment + Titles	82.8%	83.8%	82.0%	82.9%

Table 2: Bigrams

Naive Bayes Features	Accuracy	Precision	Recall	F1-score
Text	89.2%	88.8%	90.0%	89.4%
Sentiment	83.2%	84.1%	82.4%	83.2%
Actors	54.6%	72.3%	16.5%	26.9%
Directors	51.2%	66.7%	7.3%	13.1%
Genre	51.2%	57.3%	14.0%	22.5%
Titles	53.7%	59.8%	26.1%	36.3%
Actors + Directors	56.0%	71.9%	21.4%	33.0%
Actors + Directors + Genre	57.4%	68.5%	29.2%	41.0%
Actors + Directors + Genre + Titles	61.1%	68.3%	43.1%	52.8%
Actors + Directors + Genre + Sentiment + Titles	83.9%	84.8%	83.2%	84.0%
Linear SVC Features	**Accuracy**	**Precision**	**Recall**	**F1-score**
Text	87.9%	89.4%	86.3%	87.9%
Sentiment	80.0%	82.2%	77.1%	79.6%
Actors	58.4%	55.3%	93.3%	69.4%
Directors	54.5%	52.7%	97.1%	68.4%
Genre	54.2%	53.1%	81.5%	64.3%
Titles	56.4%	55.0%	76.0%	63.8%
Actors + Directors	60.4%	56.7%	92.0%	70.2%
Actors + Directors + Genre	61.2%	58.3%	81.8%	68.1%
Actors + Directors + Genre + Titles	63.0%	60.8%	76.0%	67.6%
Actors + Directors + Genre + Sentiment + Titles	81.2%	82.4%	80.1%	81.2%

F-score acquired during this investigation, closely followed by the unigram score for the 'text' feature in Linear SVC.

Once again, combining the movie-specific features yields the best result. There is no significant improvement made with the bigrams compared to the unigrams (a margin of at most 3%). Both the F-score and accuracy score are fairly similar in unigrams and bigrams. As mentioned before, the sentiment lexicon seems to perform best when using Naive Bayes, while the other features perform better with the Linear SVC. This is the case for both the unigrams and the bigrams.

5.3 FEATURE ANALYSIS

For the sentiment lexicon, there are more sentiment words in positive reviews (885.101) than negative reviews (866.802). In negative reviews, 34.7% of sentiment words have a positive sentiment and 65.3% have a negative sentiment. This makes sense, since we would expect that a negative review contains more negative words. However, in positive reviews 45.8% of sentiment words is positive and 54.2% is negative. This is counter-intuitive, but the sentiment feature still performs fairly well. The feature would have performed even better if there were more positive words than negative words in positive reviews.

Judging from Table 1 and Table 2, most movie-specific features perform up to 8% above the baseline. Before an in-depth analysis of movie-specific features is done, the average word count per review class is computed. A positive review has an average of 114 words per review, while a negative review has an average of 112 words per review. Thus, there are no significant differences in review length between positive and negative reviews.

In the actors feature set, the total number of occurrences of actors in both negative and positve reviews is 18.228. In terms of percentages, 67.8% of actors are found in positive reviews, while 32.2% are found in negative reviews. This indicates that reviewers write more about actors when the review is positive When actor occurences are studied more closely, numerous actors that occur in positive reviews also occur in negative reviews, as can be seen in Table 3.

Table 3: Top 10 Actors

Positive Actors	Absolute	Relative	Negative Actors	Absolute	Relative
Clint Eastwood	108	0.59	Woody Allen	79	0.43
Meryl Streep	90	0,49	Robin Williams	58	0.32
Sean Connery	80	0.44	Kevin Spacey	55	0.3
Al Pacino	78	0.43	Clint Eastwood	54	0.3
Tom Hanks	74	0.41	Morgan Freeman	53	0.29
Brad Pitt	74	0.41	Adam Sandler	53	0.29
Robin Williams	60	0.33	Christopher Lee	52	0.29
Bruce Willis	60	0.33	Al Pacino	51	0.28
Morgan Freeman	59	0.32	Bruce Willis	49	0.27
Rock Hudson	58	0.32	Tom Hanks	47	0.26

In the top 10 list of mentioned actors, a big overlap can be seen between the positive and negative reviews, where half of the actors occur in both review classes. In the relative columns, where occurrences are expressed in percentages compared to the total, there is a big difference between actors in positive reviews and the same actors in negative reviews. Al Pacino for instance, occurs almost twice as many times in positive reviews than in negative reviews. This information could be useful in future research, but needs to be utilized by the classifier.

For the directors feature set similar results were found compared to the actors feature set. Thus, these movie-specific feature sets have the same problem, namely

the overlap. Because of this, they lose their predictive power in a machine learning experiment like the one performed in this paper. The top 10 directors can be found in Table 7 in the appendix. Also, two tables have been created in which actors and directors are mentioned exclusively in either positive or negative reviews. This information can be seen in Table 8 and Table 9, which are also included in the appendix. This data could be useful in future experiments for classifying movie reviews.

Table 4: Top 10 Genres

Positive Genre	Absolute	Relative	Negative Genre	Absolute	Relative
War	6956	11.69	War	5971	10.03
Music	3471	5.83	Action	3105	5.22
Action	3381	5.68	Horror	2649	4.45
Comedy	2583	4.34	Music	2384	4.01
Drama	2543	4.27	Comedy	2154	3.62
Family	2524	4.24	Drama	1749	2.94
Horror	1752	2.94	Family	1527	2.57
History	1274	2.14	History	882	1.48
Musical	893	1.5	Thriller	732	1.23
Thriller	874	1.47	Mystery	629	1.06

The analysis of the genre feature can be seen in Table 4. Out of the total number of occurrences of genre, 55.4% were in positive reviews and 44.6% were in negative reviews. It is important to note here that not all occurrences of a genre refer to the genre itself. However, in order to do the analysis the naive assumption has been made that every mention is a reference to a genre. As we expected, a genre like 'horror' occurs comparably more often in negative reviews than positive reviews. Although, as with the other movie-specific feature sets, genres occur too many times in both positive and negative reviews to have significant predictive power when utilized as absolute numbers.

As mentioned before, movie titles also do not have the predictive power that was expected. However, something interesting can be seen when the list of titles that appear exclusively in either positive or negative reviews is considered. This list of movie titles can be seen in Table 5, where the corresponding IMDb scores have also been included. Judging from the table, movie titles that are exclusively mentioned in positive reviews tend to have much higher IMDb ratings than movie titles that are exclusively mentioned in negative reviews. This could be valuable information in future research, if IMDb ratings of mentioned movie titles are taken into account when performing the classification task.

Table 5: Top 10 Titles

Positive Titles	IMDb score	Negative Titles	IMDb score
The young Victoria	7.3	Son of the mask	2.2
Best in show	7.5	The omega code	3.5
Black snake moan	7.0	Half past dead	4.6
Pitch perfect	7.2	The grudge 2	5.0
Gentleman's agreement	7.4	Soul survivors	3.9
Driving lessons	6.8	Freddy got fingered	4.5
Punch-drunk love	7.3	Hanging up	4.7
Return to me	6.9	20,000 leagues under the sea	7.2
Diamonds are forever	6.7	Red Sonja	5.0
The sea inside	8.0	Exit wounds	5.5

6 CONCLUSION

Overall. there is a significant difference in predictive power for the movie-specific features and non-specific features investigated in this paper.

The 'text' feature set has proven to have the best predictability when it comes to sentiment classification of movie reviews. More specifically, 89.2% accuracy was achieved when using bigrams and a Naive Bayes classification algorithm. This score is closely followed by the 'text' feature with the use of unigrams and a Linear SVC algorithm (89.1%). Thus, the 'text' feature set performs best overall.

Furthermore, on average the Linear SVC performed slightly better than the Naive Bayes algorithm. Thus, the classification algorithm that provided us with the best results overall is the Linear SVC algorithm. Accuracy scores and F-scores are closely related, with only slight differences between them. Differences in the results of the two algorithms seem to be similar across unigrams, bigrams and trigrams.

For the movie-specific features, the results show that when all four are combined they yield the best results.This is accomplished by using unigrams with the Linear SVC algorithm. The results of the movie-specific features overall are much lower than the non-specific features of text and the sentiment lexicon. The Linear SVC seems to perform better on all the domain-specific features than the Naive Bayes algorithm.

After a deeper feature analysis, most movie-specific features seem to lose a lot of their predictive power due to the fact that most of them appear in both positive and negative reviews. Although most features like actors and directors appear more in positive reviews than in negative reviews, this does not seem to benefit the classification.

In conclusion, non-specific features perform considerably better than domain-specific feature sets when performing sentiment analysis on movie reviews. However, there is room for improvement when using these feature sets in classification. The results presented in this paper are limited due to the ad-hoc approach that has been taken.

In future work, if relative occurrence is taken into account by the classifier for the actors, directors and genre feature sets, significant improvement on predictive power should be achieved. For the movie titles, the IMDb rating could be taken into account by the classifier, which should improve the predictive power of this feature set as well. Overall, there is much room for improvement when performing sentiment classification on movie reviews. Domain-specific features can become better predictors, when utilized in the right way by the classifier.

BIBLIOGRAPHY

Anand, D. and D. Naorem (2016). Semi-supervised aspect based sentiment analysis for movies using review filtering. *Procedia Computer Science 84*, 86–93.

Broder, A. Z., S. C. Glassman, M. S. Manasse, and G. Zweig (1997). Syntactic clustering of the web. *Computer Networks and ISDN Systems 29*(8-13), 1157–1166.

Chaovalit, P. and L. Zhou (2005). Movie review mining: A comparison between supervised and unsupervised classification approaches. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pp. 112c–112c. IEEE.

Choi, Y., Y. Kim, and S.-H. Myaeng (2009). Domain-specific sentiment analysis using contextual feature generation. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pp. 37–44. ACM.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297.

Dave, K., S. Lawrence, and D. M. Pennock (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528. ACM.

Furey, T. S., N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics 16*(10), 906–914.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies 5*(1), 1–167.

Maas, A. L., R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 142–150. Association for Computational Linguistics.

Manek, A. S., P. D. Shenoy, M. C. Mohan, and K. Venugopal (2016). Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and svm classifier. *World wide web*, 1–20.

Mohri, M., A. Rostamizadeh, and A. Talwalkar (2012). *Foundations of machine learning*. MIT press.

Pang, B., L. Lee, and S. Vaithyanathan (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86. Association for Computational Linguistics.

Russell, S. and P. Norvig (1995). Artificial intelligence: a modern approach.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417–424. Association for Computational Linguistics.

Zhuang, L., F. Jing, and X.-Y. Zhu (2006). Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 43–50. ACM.

APPENDIX

Table 6: Trigrams

Naive Bayes Features	Accuracy	Precision	Recall	F1-score
Text	88.4%	87.0%	90.5%	88.7%
Sentiment	78.4%	80.1%	76.2%	78.1%
Actors	50.5%	75.9%	3.2%	6.2%
Directors	49.5%	60.0%	0.5%	0.9%
Genre	49.4%	50.0%	6.3%	11.2%
Titles	51.1%	57.9%	12.4%	20.4%
Actors + Directors	51.1%	76.9%	4.7%	8.9%
Actors + Directors + Genre	53.0%	68.3%	13.4%	22.5%
Actors + Directors + Genre + Titles	55.7%	64.5%	27.8%	38.9%
Actors + Directors + Genre + Sentiment + Titles	79.2%	80.7%	77.2%	78.9%
Linear SVC Features	**Accuracy**	**Precision**	**Recall**	**F1-score**
Text	83.6%	86.6%	80.0%	83.2%
Sentiment	75.4%	75.7%	75.7%	75.7%
Actors	53.8%	52.3%	99.2%	68.5%
Directors	51.1%	50.8%	99.8%	67.4%
Genre	52.4%	51.7%	92.5%	66.3%
Titles	53.2%	52.2%	89.4%	65.9%
Actors + Directors	55.0%	52.9%	98.9%	69.0%
Actors + Directors + Genre	57.3%	54.7%	91.1%	68.3%
Actors + Directors + Genre + Titles	58.2%	56.0%	80.9%	66.2%
Actors + Directors + Genre + Sentiment + Titles	76.2%	76.4%	76.8%	76.6%

Table 7: Top 10 Directors

Positive Directors	Absolute	Relative	Negative Directors	Absolute	Relative
David Lynch	80	1.23	Uwe Boll	87	1.34
John Ford	79	1.21	David Lynch	81	1.24
Dustin Hoffman	69	1.06	Steven Seagal	78	1.2
John Carpenter	62	0.95	Dennis Hopper	50	0.77
Orson Welles	62	0.95	Wes Craven	46	0.71
Alfred Hitchcock	61	0.94	Alfred Hitchcock	42	0.64
Dennis Hopper	58	0.89	Tim Burton	42	0.64
Sean Penn	57	0.87	John Carpenter	42	0.64
Danny Devito	57	0.87	Matt Dillon	40	0.61
William H. Macy	56	0.86	Ben Affleck	38	0.58

```
# Caution! This function is the workhorse of this experiment
# With an i5 Skylake CPU with 3.5 GHz and 16gb of DDR4 RAM it takes around 12 minutes to run
# This might be a lot more on a system with lesser hardware
features = createFeatureFrame('test')
```

```
act_dir = combineFeatures(['actors', 'directors'])
```

```
all_feat = combineFeatures(['actors', 'directors', 'genre', 'sentiment', 'titles'])
```

```
performClassification(ngram=1, df=features, mode='sentiment')
```

```
=====Naive Bayes=====    =====Linear SVC=====
Accuracy score          Accuracy score
85.2                    82.9

Precision               Precision
86.0                    83.4

Recall                  Recall
84.6                    82.8

F1-score                F1-score
85.3                    83.1
```

```
performClassification(ngram=2, df=act_dir)
```

```
=====Naive Bayes=====    =====Linear SVC=====
Accuracy score          Accuracy score
56.0                    60.4

Precision               Precision
71.9                    56.7

Recall                  Recall
21.4                    92.0

F1-score                F1-score
33.0                    70.2
```

```
performClassification(ngram=3, df=all_feat)
```

```
=====Naive Bayes=====    =====Linear SVC=====
Accuracy score          Accuracy score
79.2                    76.2

Precision               Precision
80.7                    76.4

Recall                  Recall
77.2                    76.8

F1-score                F1-score
78.9                    76.6
```

Figure 5: Example use of classification code.

```
In [2]: # Count all occurences per actor in positive reviews
        pos_actors = countFeature('positive', 'actors', 'full')

In [3]: # Count all occurences per actor in negative reviews
        neg_actors = countFeature('negative', 'actors', 'full')

In [4]: pos_actors

Out[4]: OrderedDict([('clint eastwood', 108),
                     ('meryl streep', 90),
                     ('sean connery', 80),
                     ('al pacino', 78),
                     ('tom hanks', 74),
                     ('brad pitt', 74),
                     ('robin williams', 60),
                     ('bruce willis', 60),
                     ('morgan freeman', 59),
                     ('rock hudson', 58),
                     ('errol flynn', 57),
                     ('woody allen', 57),
                     ('denzel washington', 56),
                     ('gary cooper', 56),
                     ('marlon brando', 55),
                     ('will smith', 52),
                     ('myrna loy', 50),
                     ('michael douglas', 49),
                     ('julia roberts', 47),

In [5]: # Exclude all actors that are mentioned less than 5 times
        highest_pos_actors = removeLowest(pos_actors)
        highest_neg_actors = removeLowest(neg_actors)

In [6]: total_actors = (totalCount(highest_pos_actors) + totalCount(highest_neg_actors))

In [7]: # Compute relative occurences of actors comapred to the total number, that were mentioned at least 5 times
        per_pos_actors = calcPercentage(highest_pos_actors, total_actors)
        per_neg_actors = calcPercentage(highest_neg_actors, total_actors)

In [8]: per_pos_actors

Out[8]: OrderedDict([('clint eastwood', 0.59),
                     ('meryl streep', 0.49),
                     ('sean connery', 0.44),
                     ('al pacino', 0.43),
                     ('tom hanks', 0.41),
                     ('brad pitt', 0.41),
                     ('robin williams', 0.33),
                     ('bruce willis', 0.33),
                     ('morgan freeman', 0.32),
                     ('rock hudson', 0.32),
                     ('errol flynn', 0.31),
                     ('woody allen', 0.31),
                     ('denzel washington', 0.31),
                     ('gary cooper', 0.31),
                     ('marlon brando', 0.3),
                     ('will smith', 0.29),
                     ('myrna loy', 0.27),
                     ('michael douglas', 0.27),
                     ('julia roberts', 0.26),

In [9]: # Exclude all actors that are mentioned at least 5 times and appear in both classes
        ex_pos_actors, ex_neg_actors = excludeNames(highest_pos_actors, highest_neg_actors)
```

Figure 6: Example use of feature analysis code.

Table 8: Top 10 Exclusive Actors

Positive Actors	Absolute	Relative	Negative Actors	Absolute	Relative
Victor Mclaglen	38	0.21	Jessica Simpson	29	0.16
Jeremy Northam	35	0.19	Joe Don Baker	28	0.15
Emily Watson	32	0.18	Mira Sorvino	22	0.12
Jean Peters	30	0.16	Faye Dunaway	21	0.12
Ray Charles	28	0.15	Sarah Michelle Gellar	21	0.12
Leslie Caron	28	0.15	Anne Heche	20	0.11
Robert Blake	28	0.15	Rosanna Arquette	20	0.11
Eleanor Parker	27	0.15	Telly Savalas	19	0.1
Ralph Richardson	26	0.14	Steve Guttenberg	18	0.1
Sam Waterston	25	0.14	Melanie Griffith	18	0.1

Table 9: Top 10 Exclsusive Directors

Positive Directors	Absolute	Relative	Negative Directors	Absolute	Relative
Terry Gilliam	31	0.48	Uwe Boll	87	1.34
Hayao Miyazaki	27	0.41	Andy Garcia	12	0.18
Christopher Nolan	24	0.37	George A. Romero	12	0.18
Lars Von Trier	22	0.34	Richard Donner	10	0.15
Elia Kazan	22	0.34	Ruggero Deodato	10	0.15
Michael Curtiz	21	0.32	J. Lee Thompson	10	0.15
Paul Thomas Anderson	21	0.32	Richard Kelly	10	0.15
Richard Benjamin	21	0.32	Michael Crichton	9	0.14
Mike Judge	20	0.31	Edward Burns	9	0.14
Anthony Minghella	19	0.29	Martin Lawrence	8	0.12