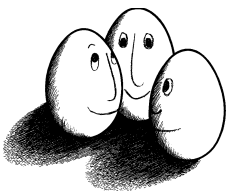
technische universität
dortmund

Diplom-Arbeit

Success prediction of upcoming movies through lexicon- and learning-based sentiment analysis algorithms on preview information in social media

Ahmet Celikkaya



Diplomarbeit
Fakultät Informatik
Technische Universität Dortmund

Dortmund, 13. April 2015

Betreuer:

Prof. Dr. Katharina Morik
Dr. Wouter Duivesteijn

Abstract

We are predicting movie success using sentiment analysis on extracted messages on Twitter. To determine the movie success we analyze the textual sentiment of texts retrieved with extraction from Twitter. We analyze and predict the movie success based on movie trailers of movies, which are not released yet and possibly have no IMDB rating. After modeling the prediction, the result will be set in comparison to the official box office sales and the later IMDB rating.

Unlike the related work about sentiment analysis, our work uses two approaches of sentiment analysis. Firstly, the lexical based approach with usage of the SentiStrength algorithm is being applied. This algorithm was trained on social media text (MySpace comments) and works without any need for labeled data and has proven successful with high accuracy especially on short informal texts [49]. The second part of the work on this approach is to optimize the SentiStrength algorithm by enhancing the lexical data sources with domain-specific idioms and language (e.g.: the phrase 'can't wait to see movie xy' is ignored in the original algorithm and gets the label 'neutral', whereas our enhancement would label that text as positive due to an anticipation sentiment of the user towards the upcoming movie).

The second part of the proposed thesis is the learning based approach. We apply SVM algorithm with data labeled by the enhanced SentiStrength algorithm to create a hybrid sentiment analysis approach. A comparison of both the lexical approach (original algorithm and enhanced) and the learning based approach in the mentioned variations are made to identify the most effective way to predict movie sales before release via tweets.

Unlike other related works, our work focuses solely on Twitter and uses not only a modified lexical approach but also a learning based approach to derive predictors. This work does not focus on trailers in particular but mostly on general movie information in social media before the release date. The assumption is, that most tweets before the release date contain trailer or preview information or user opinions on the upcoming movie.

After extracting and crawling a database of an amount of tweets, sentiment analysis / opinion mining is being run on that data to detect relations and dependencies between tweets and offline results: the box office sales. A big part of the work consists of the pre-processing of the database, since a tokenized, processed and labeled dataset is a challenge in these kinds of works and can have a big impact on the results [24]. Part of the work is also to map the retrieved sentiments to a comparable score with IMDB or find and show a relation between the IMDB score and the retrieved general sentiment of a movie on Twitter. The last step is a discussion about transferring this kind of work and analysis to other domains and discuss whether it is possible to use this approach as a generic social media model, which could also be applied on other domains.

Inhaltsverzeichnis

Abbildungsverzeichnis **vi**

Tabellenverzeichnis **viii**

1 Einleitung **1**
1.1 Einleitung und Motivation 1
1.2 Ziel der Arbeit 2
1.3 Forschungsfragen 2
1.4 Struktur der Arbeit 3
1.4.1 CRISP-DM: Cross Industry Standard Process for Data Mining . . 3

2 Verwandte Arbeiten **5**
2.1 Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment 5
2.2 Predicting IMDB Movie Ratings Using Social Media 5
2.3 Prediction of Movie Success using Sentiment Analysis of Tweets. 6
2.4 User rating prediction for movies 6
2.5 Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data 7
2.6 Predicting the Future with Social Media 7
2.7 Predicting movie success and academy awards through sentiment and social network analysis 8
2.8 Quantifying Movie Magic with Google Search 8
2.9 Twitter earthquake detection: earthquake monitoring in a social world . . 9
2.10 The Role of Preprocessing in Twitter Sentiment Analysis 10
2.11 The Role of Text Preprocessing in Sentiment Analysis 11
2.12 Predicting crime using Twitter and kernel density estimation 12
2.13 Sentiment Analysis in Twitter using Machine Learning Techniques 13

3 Grundlagen **14**
3.1 Machine Learning 14
3.2 Sentiment Analysis 15
3.3 NLP und Computerlinguistik 17
3.4 Data Mining Tools 18
3.4.1 RapidMiner 19
3.4.2 WEKA Framework 19
3.5 SentiStrength, Lingpipe, WordNet 20
3.5.1 SentiStrength 20

3.5.2 Lingpipe . . . 24
3.5.3 WordNet und SentiWordNet . . . 25
3.6 Support Vector Machine (SVM) . . . 25
3.6.1 Schlupfvariable . . . 27
3.6.2 Kernel-Trick . . . 28
3.7 Twitter und die Twitter API . . . 29
3.8 Twitter4j . . . 31

4 Extraktion der Datenbasis 34
4.1 Datenkollektion . . . 34
4.1.1 Datensätze in dieser Arbeit . . . 39
4.2 Problematiken der Twitter Extraktion . . . 40

5 Preprocessing der rohen Datenbasis 41
5.1 Preprocessing von Twitter Daten . . . 41
5.1.1 Das Verarbeitungsproblem des Datumformats bei Twitterdaten . . . 44
5.1.2 Retweets: Warum sie nicht entfernt werden . . . 46
5.2 Dimensionality Reduction . . . 46
5.3 Feature Subset Selection (FSS) . . . 48
5.3.1 Feature Selection: Filtering . . . 49
5.3.2 Feature Selection: Wrapper . . . 50
5.3.3 Ausgewählte Variante für SVM . . . 51

6 Prediction Modell und Extended Lexicon 52
6.1 Extended Lexicon . . . 52
6.1.1 Extended Lexicon Hintergrund . . . 53
6.1.2 Fehlerhafte Klassifizierungen in der Domäne ohne extended lexicon 54
6.1.3 Manuelle domänenspezifische Erweiterung des sentiment lexicons . 56
6.1.4 Automatische Lexicon-Expansion . . . 60
6.2 Modellierung . . . 61
6.2.1 SVM Modellierung in RapidMiner . . . 61
6.2.2 Regressionsmodell zur Vorhersage . . . 69
6.2.3 PT-NT Metrik . . . 74

7 Experimente: Stimmungsanalyse der Datenbasis 76
7.1 SentiStrength Experimente . . . 76
7.1.1 Experimente mit Originallexicon und expandiertem Lexicon . . . 78
7.2 Hybrides Experiment: SVM und SentiStrength Kombination . . . 91

8 Evaluierung / Success Prediction 95
8.1 Validierung des Extended Lexicon . . . 95
8.2 Vorhersage mittels ausgewählter Prädiktoren . . . 96
8.3 Vergleich mit IMDB Portal . . . 107

9 Zusammenfassung und Ausblick 108

Abbildungsverzeichnis

1.1 CRISP. (aus Vorlesung Prof. Morik, TU Dortmund, „KDD“, 2009) 4

3.1 SVM: Hyperebene H mit margin Distanzen d 26
3.2 Kernel-Trick 28

5.1 Schwerpunkt der Daten wird verschoben (ohne Berücksichtigung der Dimensionsreduktion) 47

6.1 SentiWordNet: Sentiment für „unexpected“ 56
6.2 Herangehensweise 62
6.3 Modell in RapidMiner 66
6.4 Prädiktoren für den Hercules Datensatz 69
6.5 Auszug aus Korrelationsmatrix des Datensatzes Hercules 72

7.1 Hercules box office 77
7.2 Hercules Intraday 20.07.2014 78
7.3 Hauptdarsteller; Tweet mit sehr hohem Retweet 79
7.4 Hercules Intraday 21.07.2014 80
7.5 Hercules Intraday 25.07.2014 81
7.6 Hercules score average over time 81
7.7 Hercules score average over time 82
7.8 Verteilungen der Stimmungen mehrerer Filme 84
7.9 Verteilungen der Stimmungen mehrerer Filme 85
7.10 Linke Seite: Originallexicon Sentiments, Rechte Seite: Extended Lexicon Sentiments 87
7.11 Linke Seite: Originallexicon Sentiments, Rechte Seite: Extended Lexicon Sentiments 89
7.12 Linke Seite: Originallexicon Sentiments, Rechte Seite: Extended Lexicon Sentiments 90
7.13 Zeitlicher Verlauf der URLs 91
7.14 SVM Konfusionsmatrix: Kreuzvalidierung 92
7.15 SVM Konfusionsmatrix: bigram 92
7.16 SVM Model Tree 94

8.1 Guardians of the Galaxy Statistiken 97
8.2 Green= Prediction, Black= Box Office 98
8.3 Prediction vs Real Box Office und Lexicon vs Extended Lexicon 100
8.4 Prediction vs Real Box Office und Lexicon vs Extended Lexicon 102
8.5 Prediction vs Real Box Office und Lexicon vs Extended Lexicon 103

8.6 Prediction vs Real Box Office: SVM Modell 105
8.7 Prediction vs Real Box Office und Lexicon vs Extended Lexicon 107

Tabellenverzeichnis

3.1 Tabelle: Vergleich der Algorithmen [49] 23

4.1 Film-Datensätze 39

6.1 Neue Idiomliste 60
6.2 Konfusionsmatrix 65

7.1 Vergleich: SVM und Lexicon 93
7.2 Auszug aus Hercules: SVM Ergebnisse 93

8.1 Scores der drei Herc-Datensätze 95
8.2 p-Values und adjusted R^2für Extended und Original Lexicon 104
8.3 p-Values und adjusted R^2 für SVM Modell 106

Notation und Abkürzungen

- WordNet: Hierarchisch aufgebautes digitales Wörterbuch in Baumstruktur.
- SentiWordNet: Verwendet WordNet und weist jedem Wort in WordNet einen Stimmungswert zwischen $1 < x < 0$ zu.
- PT-NT Ratio: Verhältnis der Anzahl positiver Tweets zu den negativen Tweets.
- IMDB: Internet Movie Database. Eine bekannte Plattform zur Bewertung von Filmen.
- Extended Lexicon: Erweiterung der Wortliste, welche zur Stimmungsanalyse verwendet wird.
- Tweets / Retweets: Öffentliche Kurznachrichten auf Twitter bzw. kopierte Weiterleitungen dieser.
- Feature / Prädiktor(-Variable): Attribut zur Vorhersage eines anderen Attributes.
- SVM: Support Vector Machine.
- Wrapper Funktion: Eine Funktion, welche eine andere Funktion umgibt und verwendet.
- SentiStrength: API-Tool zur Stimmungsanalyse mit Lexicon und Text Mining.
- NLP: Natural Language Processing. Digitale Verarbeitung von natürlicher Sprache.

- Box Office: Bezeichnung für die Verkaufszahlen eines Films. (Abgeleitet von der Ticketkabine „Box Office“).

1 Einleitung

1.1 Einleitung und Motivation

Twitter (angefangen 2006) ist heute eine weltweit bekannte Marke und mit Facebook der meist verwendete Dienst im Bereich des Social Media. Twitter erlaubt es dem Anwender Kurznachrichten (sogenannte *Tweets*) zu schreiben und zu veröffentlichen. Jeder Tweet darf eine maximale Zeichenanzahl von 140 haben und kann öffentlich oder aber in einer geschlossenen Gruppe von Abonennten (genannt *Follower* in der Twitter Terminologie) publiziert werden. Beim Verfassungsdatum dieser Arbeit hat Twitter laut offiziellen Angaben über 241 Millionen monatlich aktiver Benutzer und insgesamt über eine Millarde registrierter Nutzer (entsprechend des Quartalsberichts bei der New Yorker Börse NYSE:TWTR Q4 Twitter Report [36]). Von diesen werden täglich über 500 Millionen Tweets erstellt bzw. getwittert. Es ist inzwischen üblich, dass jede Werbekampagne zu einem neuen Produkt oder einem neuen Kinofilm auch über Twitter läuft. Auch werden seit einiger Zeit viele politische Wahlen (wie z.B. bei Barack Obama 2012 in den USA, siehe [50]) sehr stark durch Twitter-Kampagnen der jeweiligen Parteien und Wahlbüros unterstützt, um so Einfluss auf die potentielle Wählerschaft auszuüben.

Es gibt einige wenige große Unternehmen, die bezahlte Dienste für Twitteranwender anbieten. Diese Dienste sind unter anderem der Verkauf von Twitterdaten und Tweets, Vorhersagen von Wahlen, Werbekampagnen und Trends. Twitter unterstützt diese angebotenen Dienste, indem es den bezahlten sogenannten *Firehose Zugang* anbietet. Dieser Firehose Zugang erlaubt es dem Nutzer, mehr Twitter-Daten über die Twitter API abzugreifen als es dem kostenlos genutztem Twitter Zugang erlaubt ist. Dieser Zugang wird von Twitter auch einigen akademischen Arbeiten und Institutionen erlaubt.

Twitter hat bis zum heutigen Zeitpunkt 4 Unternehmen als offiziell zertifizierte Datenverkäufer bekanntgegeben: Gnip, Data Sift, Topsy und NTT Data [47]. Diese Unternehmen bezahlen Twitter und verkaufen dann Daten und Dienste weiter an kleinere Unternehmen und Endnutzer. Dabei ist Topsy (`www.Topsy.com`) das einzige Unternehmen, das auch bis zu einem gewissen Grad kostenlose Dienste anbietet und mit Start von Twitter auch sämtliche Tweets bis zum gegenwärtigen Zeitpunkt extrahiert hat. Das heißt, mittels einer Datensuche in Topsy ist es möglich, Tweets von 2006 einzusehen und zu analysieren. Diese Dienste und Daten werden unter anderem für Marktanalysen, wissenschaftliche Arbeiten und Vorhersagen verwendet. Offensichtlich gibt es ein steigendes kommerzielles wie akademisches Interesse an dem stetig wachsendem Bereich der Stimmungsanalyse (***Sentiment Analysis***) mit Twitterdaten. Eine akademische Studie bzw. Arbeit über Sentiment Analysis bietet sich daher an.

1.2 Ziel der Arbeit

Das Ziel der vorliegenden Arbeit ist die Erfolgsprognose bzw. Vorhersage von aktuellen sehr bekannten und weniger bekannten Kinofilmen anhand der Stimmungsanalyse mittels Datenextraktion von Nachrichten aus Twitter. Zur Bestimmung des Erfolges und Erfolgsgrades wird der sogenannte textuelle *Sentiment* - also die textuelle (emotionale) Stimmung - eines Textes bzw. Kurztextes von Twitter entnommen. Ein grundlegendes Ziel ist es, dass die entsprechenden Tweetdaten zum Teil schon *vor* Erscheinen des Kinofilms als Datenbasis genommen werden. Dadurch ist für den Zeitraum vor dem offiziellem Kinotag gewährleistet, dass es sich bei den Daten zum großen Teil um Tweets bezüglich der Filmtrailer und Filmpreviews handelt. Allgemein ist auch davon auszugehen, dass in diesem Zeitraum auch noch gar keine oder zu mindest keine repräsentative Bewertung auf den größten Filmbwertungsportalen wie zum Beispiel IMDB gibt. Nach der Modellierungphase einer Prognose werden die Ergebnisse dieser in Relation zu den offiziellen Verkaufszahlen gesetzt. Es soll geprüft werden, ob es möglich ist, Variablen von Twitterdaten abzuleiten, um diese in einem statistischen Modell für Prognosen zu verwenden.

Ein Ziel ist es auch, Daten über den offiziellen Starttermin des Kinofilms hinaus zu sammeln und etwaige Schwankungen und Änderungen der Stimmung **vor** Starttermin und **nach** Starttermin aufzudecken. Abschließend wird diskutiert, inwieweit diese Methoden auch auf andere Domänen und Themengebiete transferiert werden können und in welchem Umfang eine Anpassung notwendig wäre.

1.3 Forschungsfragen

Im folgenden werden die behandelten Forschungsfragen aufgelistet:

- Wie können die Daten aus Twitter in einer für eine Analyse nutzbaren Form extrahiert werden?
- Wie kann der riesige Umfang an Daten in Twitter für Datenanalysen und Prognosen verwendet werden?
- Kann die Stimmung zum einem Thema a priori derart manipuliert werden, so dass eine negative Stimmung relativiert werden kann?
- Gibt es eine Diskrepanz der Stimmung vor Starttermin und nach Starttermin?
- Wie stehen Portale wie IMDB in Verbindung zu Tweets, gibt es einen Stimmungsunterschied oder eine Korrelation?
- Wie umfangreich und wichtig ist das Preprocessing?
- Kann man ein statistisches Modell mittels Twitterdaten bezüglich Verkaufszahlen aufstellen?
- Wie aussagekräftig wäre dieses Modell dann?

1.4 Struktur der Arbeit

Die Arbeit richtet sich vom Aufbau her nach der in wissenschaftlichen Arbeiten üblichen Struktur. Insbesondere wird die *CRISP-DM* Herangehensweise (siehe dazu Unterkapitel 1.4.1) verwendet. Nach einem englischen Abstract werden verwandte Arbeiten erwähnt und deren Herangehensweisen und Ansätze erläutert. Es werden Ähnlichkeiten und Unterschiede zur vorliegenden Arbeit erwähnt. Im Anschluss folgt das Kapitel **Grundlagen**. Die Grundlagen beeinhalten die Erklärung des Themas, Eigenschaften, Möglichkeiten und Grenzen der Stimmungsanalyse. Es werden außerdem in der Branche verwendete Programmbibliotheken und Tools wie Sentistrength und RapidMiner vorgestellt und gezeigt, aus welchem Grund eine bestimmte API für diese Arbeit ausgewählt wurde. SentiStrength ist ein von Prof. Mike Thelwall entwickelter lexikaler Algorithmus zur Stimmungsbestimmung in Texten [49].

Die Grundlagen erklären des Weiteren das soziale Medium Twitter und die Twitter Programmierschnittstellen, die auch in dieser Arbeit für die Extraktion und das Preprocessing verwendet werden. Abschließend wird die Computerlinguistik, auf die die Stimmungsanalyse basiert, erklärt.

Ein wesentlicher Bestandteil der Arbeit ist das Sammeln der Daten und die Speicherung dieser, um einen *Textkorpus* zu erstellen. Diese Thematik erhält daher ein eigenes Kapitel zur Datensammlung, Problemen der Extraktion mit der Streaming API von Twitter und einen Überblick des Datenmodells. Nachfolgend wird im Kapitel **Preprocessing der rohen Datenbasis** 5 aufgezeigt, wie die rohen Datensätze vor der Stimmungsanalyse aufbereitet werden.

Es findet in Kapitel 6 eine Erweiterung des Lexicons von SentiStrength statt, um die Domäne besser erfassen zu können.

Das Kapitel 7 *Experimente* umfasst die Sentiment Analysis und ist unterteilt in die Bereiche SentiStrength, SVM und hybride Analyse.

Darauffolgend werden dann die Ergebnisse evaluiert und mit den offiziellen Verkaufszahlen verglichen, um dann darauffolgend eine Methode zu entwickeln.

1.4.1 CRISP-DM: Cross Industry Standard Process for Data Mining

CRISP-DM steht für *Cross Industry Process for Data Mining* und ist ein EU gefördertes Prozessmodell, das den sequenziellen Ablauf der Einzelschritte im Data Mining skizziert[1]. CRISP ist kein wissenschaftlich bewiesenes Modell, sondern vielmehr eine Ansammlung von Vorgehensweisen, die sich in der Praxis bewährt haben, d.h. *best-practices*. Es gibt in der CRISP Initiative viele große Unterstützer wie IBM, SPSS, DaimlerChrysler etc.

Es gibt insgesamt 6 aufeinanderfolgende Phasen in CRISP-DM:

[1] http://www-ai.cs.uni-dortmund.de/LEHRE/VORLESUNGEN/KDD/SS09/1DMVintro.pdf

1. **Business Understanding:** Analyszeziele, Anforderungen, Projektplan
2. **Data Understanding:** Datenkollektion bzw. Extraktion, Beschreibung, Untersuchung der Rohdaten
3. **Data Preparation:** Datenaufbereitung, Preprocessing, Transformation
4. **Modeling**: Modellauswahl, Testdesignauswahl, Schätzung, Modellqualität
5. **Evaluation:** Modell bewerten, bisherigen Prozess überprüfen, Nächste Schritte
6. **Deployment:** Anwendungsplan, Präsentation, Bericht

Projekte im Bereich des Data Mining können sich an das CRISP-DM Prozessmodell orientieren. Allgemein wird CRISP als das am meisten verwendete Prozessmodell im Data Mining betrachtet[2]. Daher wird auch in dieser Diplomarbeit der Gesamtprozess entsprechend der Schritte im CRISP-Modell durchgeführt.

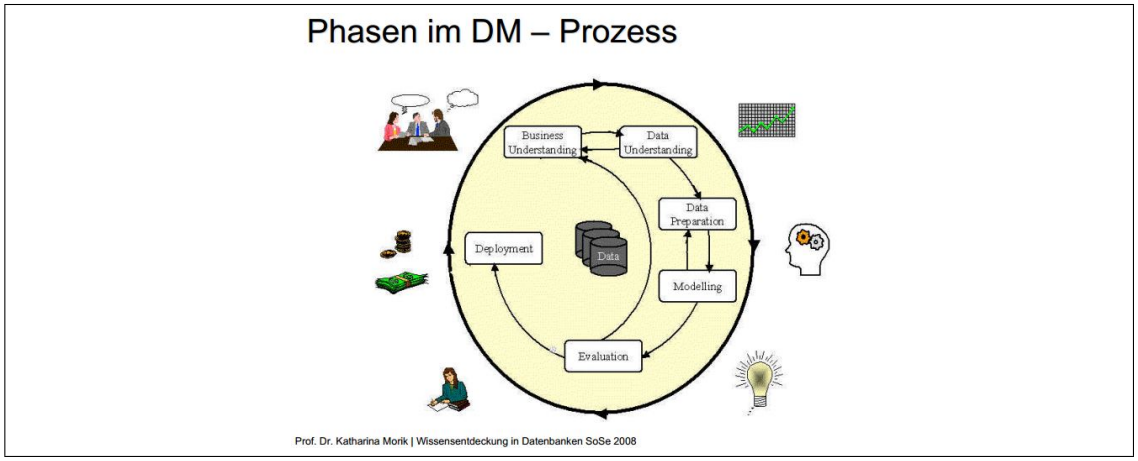


Abbildung 1.1: CRISP. (aus Vorlesung Prof. Morik, TU Dortmund, „KDD“, 2009)

Das Kapitel **Einleitung**(1) stellt die erste Phase dar und stellt den groben Arbeitsplan und die Ziele vor. Das Kapitel **Extraktion**(4) entspricht der 2.Phase, dem Data Understanding. Entsprechend der CRISP-Abfolge und Phase 3, wird dann eine Datenaufbereitung im Kapitel **Preprocessing der rohen Datenbasis**(5) vorgenommen. Die Phase **Modeling** in CRISP wird im Kapitel Prediction Modell(6) durchgeführt. Das darauffolgende Kapitel der Experimente gehört ebenfalls noch zu dieser Phase. Gemäß CRISP gibt es danach das Kapitel der **Evaluation**(8). Das letzte Kapitel entspricht Phase 6 in CRISP und präsentiert das Gesamtergebnis in Bezug auf die Ziele aus Phase 1.

[2] `http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm`

2 Verwandte Arbeiten

In diesem Kapitel werden verwandte und ähnliche Arbeiten im Bereich der Stimmungsanalyse vorgestellt.

2.1 Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment

Es gibt einige Arbeiten und Papers über Twitterdaten und Data Mining, die sich auf die politische Domäne spezialisiert haben [1]. In der Arbeit von Tumasjan et al. versucht der Autor die deutschen Bundestagswahlen von 2009 vorherzusagen. Dazu wurden 100.000 Tweets gesammelt und verarbeitet. In der Twitter API hat der Autor die Namen der deutschen Parteien als Schlüsselwörter verwendet. Als Sentiment Analysis Werkzeug verwendet der Autor die Software Linguistic Inquiry and Word Count (LIWC) [33]. Eine der Funde der Arbeit ist beispielsweise, dass bereits alleine mit der Tweetrate von Tweets pro Stunde über eine politische Partei eine Wahlgewinner-Vorhersage mit prozentualen Ergebnissen gemacht werden konnte. Die Vorhersage war sehr nah an den tatsächlichen Wahlergebnissen. So sagte die Analyse der Arbeit voraus, dass die CDU 30,1 % erhält. Das tatsächliche Wahlergebnis lag bei 29 %.

Die Autoren haben in ihrem Modell 12 verschiedene Kategorien definiert. Diese sind Traurigkeit, positive Emotionen, Geld, Erfolg, Wut, Angst und andere. Diesen Kategorien wurden dann die Tweets über die Parteivorsitzenden in der Wahl verteilt. Dazu verwendeten sie LIWC, um die Stimmung und den Sentiment Score zu klassifizieren und jeden Tweet mit einem der Kategorien zu kennzeichnen. Auf die Art konnten sie die generelle Wahrnehmung der Twitter Benutzer gegenüber den Politikern bestimmen. Insgesamt konnten die Autoren den Wahlsieger und mit einem MAE von 1,65% auch die prozentualen Wahlanteile der Parteien vorhersagen.

2.2 Predicting IMDB Movie Ratings Using Social Media

In [18] wird eine Analyse über die Vorhersage von einer Filmbewertung auf IMDB gemacht. Die Arbeit setzt den Fokus stark auf eine Information Retrieval Perspektive und betrachtet die Aufgabe als eine *Cross-Channel* Vorhersage, in der Signale von verschiedenen Kanälen entnommen werden. In der Arbeit sind diese Kanäle der Signale Twitter und YouTube. Es soll eine Korrelation zwischen IMDB Bewertungen eines Films und den sogenannten Aktivitätsindikatoren des entsprechenden Films gefunden werden. Die

Aktivitätsindikatoren sind in diesem Fall beispielsweise die Tweets. Ein Beispiel wird in dem Paper gegeben, in dem ein Film mit einer sehr hohen IMDB Bewertung von 8.9/10 Punkten ca. 167.000 Tweets hat und ein anderer Film mit einer Bewertung von 1.1/10 nur 25.000 Tweets. Vorgeschlagen werden 2 verschiedene Aktivitätsindikatoren: Quantitative Indikatoren (z.B. die Gesamtzahl an Tweets zu einem Film) und qualitative Indikatoren wie der extrahierten Stimmung eines Tweets oder YouTube Kommentars. In dem Experiment werden 1.600.000 Tweets und 55.000 YouTube Kommentare als Datenbasis verwendet. Es wird lineare Regression mit 10-fold Cross-Validation von 60 Filmen in Weka verwendet. Insgesamt wird im Ergebnis der Arbeit gezeigt, dass mit einer Kombination der Anzahl von Likes/Dislikes in YouTube Kommentaren und den extrahierten Stimmungen von Twitternachrichten die besten Resultate erzielt werden konnten.

2.3 Prediction of Movie Success using Sentiment Analysis of Tweets.

Eine weitere Arbeit über Erfolgsprognosen von Kinofilmen mittels Sentiment Analysis ist die von Vasu Jain [27]. Das Ziel der Arbeit ist es, den Erfolg des Films an den offiziellen Verkaufszahlen (der sogenannte Box Office Sale) zu messen. Dazu wurde ein Datensatz an Tweets extrahiert, verarbeitet und basierend auf der Stimmung des Textes als positiv, negativ, neutral oder irrelevant (z.B. Tweets ohne Kontextbezug zum betrachtetem Film) gekennzeichnet. Als Werkzeug bediente sich der Autor der Lingpipe [32] Sentiment Analysis API in Java. Die Filme wurden anhand dieser und einer Metrik namens *PT-NT ratio* als Hit, Flop oder Durchschnitt klassifiziert. Das PT-NT ratio ist das Verhältnis der Gesamtzahl der positiven bzw. negativen Tweets.

In der Arbeit werden auch Experimente mit Zeitintervallen und Tweetzeiten durchgeführt. Der Autor identifiziert daraus einen kritischen Zeitraum für Tweets und nennt solche Tweets die “critical period tweets”. Jene Tweets werden laut Autor immer in der ersten Woche vor Erscheinungsdatum des Films gesendet und sind nach seiner Behauptung deutlich zahlreicher als Tweets, welche vor oder nach dieser kritischen Zeitspanne geschickt werden. Die Arbeit schließt dann mit der erfolgreichen Vorhersage der Box Office Sales Performanz von insgesamt 5 der betrachteten 7 Filme (es gab einen 8. Film, der aber auf Grund fehlender Verkaufszahlen nicht in die Arbeit einfloss).

2.4 User rating prediction for movies

In [54] wird ein gewisser Score kalkuliert, der aus den einzelnen Scores der Schauspieler, des Regisseurs, des Produzenten und des Drehbuchautors zusammengesetzt wird. Es wird das Ziel verfolgt, den Erfolg des Kinofilms und die Bewertungen auf IMDB in Relation zu den genannten Attributen - also Scores bzw. Punktzahl - zu setzen. Jedes Attribut (Casting und die Mitarbeiter am Set) erhält dafür eine Gewichtung. Diese Gewichtung beruht hier nicht auf einer mathematischen statistischen Berechnung, sondern auf der persönlichen Einschätzung des Autors gemessen an der Bekanntheit. Entspre-

chend des Scores eines Schauspielers beispielsweise und den Filmen, in denen er eine Rolle nahm, wird die Gewichtung mit der Gesamtbewertung auf IMDB multipliziert. Das Ergebnis ist dann der gesamte Score des Schauspielers. Diese Scores werden dann als Input für ein neurales Netzwerk verwendet, welches die Bewertung und den Erfolg des Films vorhersagen.

2.5 Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data

Mestyán, Yasseri und Kertész [5] analysieren den Box Office Erfolg ebenfalls auf eine andere Art und Weise. Sie messen den sogenannten *Activity Level* (Aktivitätsgrad) eines Bewertungsautors und der Betrachter eines Filmeintrags auf Wikipedia. Als Kennzahlen für den Aktivitätsgrad werden die Anzahl der Seitenaufrufe des Film-Artikels, die Anzahl der Benutzer, welche einen Beitrag zum Artikel leisten und die Anzahl an Nachbearbeitungen des Artikels verwendet. Ein anderer betrachteter Faktor, der in dieser Form in keine der verwandten Arbeiten auftaucht, ist die Einbeziehung der Anzahl der Kinosäle, die den Film aufführen. Das verwendete Modell ist eine multivariate Linearregression.

Insgesamt wurden für die Analyse eine gewisse Menge an Daten gesammelt. Diese Daten umfassen insgesamt 312 Filme aus dem Jahre 2012. Unter Verwendung der definierten Messzahlen in der Formel und mit der Charakterisierung der Güte der einzelnen Datensätze *(Goodness of different Sets)*, erreichen die Autoren bei der einen Hälfte des Experiments $R^2 = 0.94$ oder 0.98 bei einem anderen Experiment mit einem Datensatz. In der Arbeit werden Vergleiche zu den Ergebnissen aus einer anderen wissenschaftlichen Arbeit von [8] gezogen. Als Endresultat erreicht der Wikipedia-Ansatz dieser Arbeit $R^2 = 0.98$ und gibt ein Herausstellungsmerkmal der Arbeit an, dass es erlaubt mittels der in der Arbeit verwendeten Methoden den Erfolg eines Films schon einen Monat vor Erscheinungsdatum vorherzusagen, was in der verglichenen Arbeit [8] nicht möglich sei.

2.6 Predicting the Future with Social Media

Wegen der Hervorherhebung und den Vergleich der Arbeit von [8] durch [5] wird hier auch auf diese Arbeit als verwandte Arbeit hingewiesen. Die Arbeit wurde von B. Huberman und S. Asur verfasst. Sie schreiben über das große Prognosepotential in den sozialen Medien im Allgemeinen und über Twitter im Speziellen. Die beiden Autoren haben sich zum Ziel gesetzt, allgemeine Vorhersagen über reale Ereignisse in der Welt machen zu können. Dafür verwenden sie die Domäne der Kinofilme, um den Ansatz und die Herangehensweise dann abstrahieren und auf andere Domänen anwenden zu können. Für diesen Zweck extrahierten sie insgesamt 2.8 Millionen Tweets zu 23 verschiedenen Filmen. Mit diesem großen Datenbestand fanden sie einen Korrelationskoeffizienten von 0.64 für Tweets, die eine Filmbezogene URL im Text enthielten (z.B. ein Bild oder eine Webseite mit Bildern und Previews) zu der Box Office Performanz bzw. den Verkaufszahlen [13].

Die Autoren zeigen des Weiteren einen Tweetratenkorrelationswert von 0.90 mit dem Box-Office Verkauf und einen Korrelationskoeffizienten von 0.97 im Vergleich zum HSX Hollywood Stock Exchange [46], welcher in Form einer virtuellen Börse als der goldene Standard der Erfolgsprognose von Hollywoodfilmen gelte. Durch die virtuellen Börsenkurse der Filme können die wahrscheinlichen Verkaufszahlen abgelesen werden. Die Stimmungsanalyse wird im Paper als alternative Methode der Vorhersage angewandt. Nach dem Preprocessing der Tweets und der manuellen Stimmungs-Kennzeichnung mit Amazon Mechanical Turk [7] als Vorbereitung für das Trainieren mit den Daten, wird von Hubermann et al. festgestellt, dass die Stimmungsanalyse gute Ergebnisse abliefert aber ihr Ansatz mit der Tweetrate dieser zu bevorzugen ist. Insgesamt ist das Ergebniss der Arbeit, dass mit den Aktivitäten in den sozialen Medien und der Tweetrate auf Twitter effektive Indikatoren für reale Vorhersagen über Ereignisse in der Welt existieren.

2.7 Predicting movie success and academy awards through sentiment and social network analysis

Krauss et al. schlagen einen anderen Ansatz vor, um Vorhersagen in der Filmdomäne zu machen. Sie geben Foreneinträgen und Diskussionen zwischen Benutzern eine Gewichtung [3]. In der Arbeit konnten 9 Oskarnominierungen zwei Monate vor der Verleihung erfolgreich vorhergesagt werden. Als Datensätze werden extrahierte Diskussionen und Beiträge auf IMDB (alle Einträge zwischen Dezember 2005 und Dezember 2006) und box office mojo verwendet [13]. Es werden die Diskussionen in Vergleich zu den Verkaufszahlen gesetzt. Das Modell beeinhaltet 3 Parameter. Diese sind Intensität der Diskussion, Positivität und Zeitrahmen.

Die Positivität wird hierbei als schwierigster zu bestimmender Parameter genannt. Die Intensität der Diskussion wird als die Anzahl der Erwähnungen definiert und ist eine Zahl zwischen 0 und 1. Die Positivität ist die Stimmungskalkulation (Sentiment Analysis) mittels des Analysetools Condor [15]. Condor erlaubt es dem Nutzer, Verbindungen zwischen Teilnehmern und Aktivitäten auf einer Webseite oder einem Forum aufzudecken. Des Weiteren kommt in dieser Arbeit auch LIWC [33] zur Stimmungsanalyse zum Einsatz. Die Autoren experimentierten auch mit Datensätzen, die aus Film-Preview Diskussionen stammen. Dadurch waren sie in der Lage zu zeigen, dass eine hohe Kommunikationsintensität auch zu einem größeren Erfolg und höheren Verkaufszahlen korrelierte.

2.8 Quantifying Movie Magic with Google Search

Das Unternehmen Google hat ein Whitepaper zum Thema Erfolgsvorhersage bei Kinofilmen veröffentlicht [2]. Die Datengrundlage bilden die gesammelten Daten von Google Search Anfragen. Die Autoren sind Mitarbeiter bei Google und haben daher uneingeschränkten Zugriff auf alle möglichen Google Search Statistiken, Datenquellen und Anfragen. Das Whitepaper sagt aus, dass 61% aller Kinobesucher angaben, dass sie sich von Onlinequellen bedienten, um sich über einen Film zu informieren. Mögliche Onlinequel-

len sind hier die Google-Suche und YouTube. Die Arbeit zeigt eine Korrelation zwischen filmbezogenen Suchaktivitäten auf Google und dem Box Office Verkauf. Die Genauigkeit einer Filmprognose ist laut Google 92% in der Eröffnungswoche basierend auf dem Suchanfragevolumen und dem bezahlten Klickvolumen. Unter Einbeziehung von Trailerbezogenen Suchtrends kombiniert mit dem Markenstatus des Films (*Franchise*) und des Saisoneinflusses, schaffen die Google Mitarbeiter eine Vorhersage der Verkaufszahlen mit einer Accuracy von 94%.

Ein anderer Punkt in der Arbeit ist, dass das Suchvolumen sieben Tage vor Starttermin der mächtigste Indikator für die Box Office Performanz eines Films ist. Laut dem Paper können 70% der Performanz im Box Office mit dem Suchvolumen auf Google ergründet werden. Es ist auch wichtig zu erwähnen, dass in dem Whitepaper von Google die Aussage gemacht wird, dass Trailersuchen auf Google Schlüsselindikatoren für die Erfolgsprognose sind. Die vorliegende Diplomarbeit geht auch von der im Laufe der Arbeit aufzuzeigenden Annahme aus, dass Preview und Trailer Informationen (in diesem Fall der Sentiment von Einträgen in den sozialen Medien über den Filmtrailer) sehr wichtige Indizien für die Prognose sind.

2.9 Twitter earthquake detection: earthquake monitoring in a social world

Eine innovative Arbeit über Sentiment Analysis wurde für Erdbebenvorhersagen verfasst [17]. Die Arbeit wurde von der U.S. Geological Survey(USGS) subventioniert. In dem Projekt geht es um die Verbesserung der Fähigkeit des USGS bei der Vorhersage von Erdbeben. Die Software die im Rahmen des Projekts entwickelt wurde trägt den Namen TED - Twitter Earthquake Detection. Das Projekt basiert auf der Tatsache, dass sehr viele ortsansässige Menschen Twitter verwenden, um Mitteilungen über gespürte Erdbeben zu machen. Diese Informationen könnten zur Lokalisierung des Epizentrums oder aber als Alarmmechanismus verwendet werden. In der Arbeit wurden insgesamt 48 Erdbeben innerhalb von 5 Monaten weltweit detektiert. Nur 2 der 48 entdeckten Erdbeben waren Fehlinterpretationen von TED. Obwohl Seismographen im selben Zeitraum weit über 5000 Erdbeben feststellten, hat die Vorhersage mittels TED und Twitterdaten große Vorteile. 75% der Detektierungen konnten innerhalb der ersten 2 Minuten des Erdbebens gemacht werden, was sehr viel schneller ist als die meisten seismographischen Feststellungen in den technisch schlecht ausgestatteten Regionen der Welt. TED hat auch den Vorteil sehr günstig zu sein. Für den Betrieb reicht ein handelsüblicher Laptop mit Internetverbindung und der TED Software anstatt eines teuren Seismographensystems. Der Autor weist darauf hin, dass der Twitter Detektor kein vollständiger Ersatz für ein seismisches Netzwerk sein kann. Dafür könne TED aber ein Hilfstool zur Frühwarnung und Entdeckung von Erdbeben in den schlechter ausgestattenen Regionen der Welt sein. Auch könnte TED zum Sammeln von Informationen und Kurzberichten der Ortsansässigen verwendet werden. Ein anderer Vorteil von TED sei die Möglichkeit, dass TED als System anderen Frühwarnsystemen vorgeschaltet werden könnte um die

beobachtenden Stellen vorab zu warnen, da TED ein Erdbeben potentiell schneller als ein Seismograph feststellen könne.

2.10 The Role of Preprocessing in Twitter Sentiment Analysis

Eine Grundlagenarbeit zum Thema der Relevanz von Vorverarbeitung in der Sentimentanalyse wurde von Bao et al. verfasst [10]. Bao et al. stellen in ihrer Arbeit heraus, dass der Prozess der Vorverarbeitung einen starken Einfluss auf das Endergebnis der Sentimentanalyse haben kann, da Tweets sehr viele irreguläre sprachliche Elemente enthalten können. Dafür haben die Autoren insbesondere die in Tweets oft vorkommenden Textbausteine und Elemente wie URLs, umgangssprachliche Elemente wie Suffixverlängerungen, Stemming usw. in Betracht gezogen. Die Experimente von Bao et al. wurden in folgender Reihenfolge mit dem Stanford Twitter Sentiment Dataset durchgeführt:

1. Rauschentfernung: Als *Denoising* wird dieser erste Schritt bezeichnet. Dafür werden zu erst die Benutzernamen im Tweet komplett entfernt und mit Leerzeichen ersetzt. Ausserdem werden alle sogenannten Hashtags (#), welche in Twitter als Topic oder als Tag fungieren, ebenfalls mit Leerzeichen ersetzt. Des Weiteren werden alle Emoticons, die keinem bestimmten Sentiment zugeordnet werden können, auch entfernt. Offensichtliche Emoticons wie ":)" werden beibehalten. Als letzten Punkt werden alle Zahlen, Einzelbuchstaben und Symbole entfernt.
2. Nach der Rauschentfernung werden in dem Experiment folgende Preprocessing Schritte angewandt: URL feature reservation (Beibehaltung der URLs als Attribute), Negationstransformation und Normalisierung von Buchstabenwiederholungen, Stemming, Lemmatization. Mit Stemming wird die Wortwurzel eines Wortes anstatt des Wortes selbst genommen. Dadurch hat die Endung des Wortes keinen Einfluss mehr auf die Bedeutung des Wortes und eine Wortredundanz wird vermieden. Lemmatization wandelt die Worte in ihre Ursprungsform bzw. in den Infinitv um.
3. Auswahl der Attribute mittels Term Frequency (Anzahl Vorkommen des Attributs), Information Gain, X^2 Statistics (Interdependenz zwischen Attribut und Klasse).
4. Klassifikation mittels Liblinear, einem linearen Klassifizierer.

Als Framework verwendeten die Autoren unter anderem WEKA [38]. Als Resultat der Experimente stellen die Autoren heraus, dass bei der Rauschentfernung die URL feature reservation, Negationstransformierung, Normalisierung der Buchstabenwiederholungen einen positiven Einfluss auf die Accuracy haben, wohingegen Stemming und Lemmatization sich eher negativ auswirkten. Mit allen Schritten der Rauschentfernung erreichten die Autoren eine Accuracy von 81,62 %, wohingegen bei einer Beibehaltung von URLs die Accuracy auf 82,73 % anstieg. Laut Bao et al. könnte der Grund für diesen Effekt sein, dass manche URLs Attributinformationen enthalten könnten. Es kann insgesamt also festgehalten werden, dass bewiesen wurde, dass Vorverarbeitungsprozesse notwen-

dige Schritte für eine ausführliche Sentimentanalyse sind. Wenngleich in dem Paper kein allzugroßes prozentuales Wachstum festzustellen ist, sollte doch darauf hingewiesen werden, dass es noch viele andere Preprocessing Schritte gibt, die hier nicht alle angewandt wurden, wie z.B. POS Tagging.

2.11 The Role of Text Preprocessing in Sentiment Analysis

Eine andere Arbeit über die Signifikanz von Preprocessing wurde von Haddi et al. verfasst [24]. Die Autoren Haddi et al. analysieren in ihrer Arbeit, wie stark der positive Einfluss von verschiedenen Vorverarbeitungsschritten im Hinblick auf die Accuracy sein kann. Für die Experimente verwenden die Autoren Support Vector Machine (SVM) Algorithmen. Wichtig und interessant an dieser Arbeit ist, dass die Autoren beweisen, dass mittels geschicktem Preprocessing und SVM ähnlich hohe Accuracy Werte erzielt werden wie beim Topic Modelling bzw. Topic Categorisation, obwohl die Sentimentanalyse eine schwierigere Textminingaufgabe sei.

Es wird in dem Paper darauf eingegangen, dass vor allem Unternehmen mit großen Daten an Reviews für ihre Produkte umgehen müssen. Durch die Größe der Datensätze und dem Datenrauschen (*Noise*) sei die Aufgabe aber vor allem in Echtzeit kaum zu schaffen. Dadurch sei es allein schon aus Performanzgründen notwendig, ein umfangreiches Preprocessing der Datensätze durchzuführen, um eine Echtzeit-Sentimentanalyse durchführen zu können. Das Preprocessing umfasst folgende Transformationen: Online text cleaning, Leerzeichenentfernung, Expandieren von Abkürzungen beziehungsweise Abkürzungen, Stemming, Entfernung von für den Textsentiment irrelevanten Stoppwörtern, Negationsbehandlung und Attributselektion. Insbesondere auf die Attributselektion wird in dem Paper viel Wert gelegt. Zur Gewichtung der Attribute werden *Feature Frequency* (FF), Term Frequency Inverse Document Frequency (TF-IDF) und Feature Presence (FP) verwendet. Die Features sind hier die Attribute. Als Datensätze werden zwei Sets von Filmbewertungensdokumenten verwendet, wobei ein Datensatz 1400 Dokumente enthält und der andere 2000 Dokumente. Bei Dokumentdatensätze bestehen zur einen Hälfte von positiven und zur anderen Hälfte von negativen Filmbewertungen.

Interessant ist in der Arbeit von Haddi et al. auch die domänenspezifische Definition und Anwendung von zu filternden beziehungsweise entfernenden Stoppwörtern. Da die Arbeit in der Filmdomäne angesiedelt ist, wurden Worte wie Film, Movie, Actor, Actress, Scene als irrelevante Stoppwörter definiert. Dadurch haben sie keinen Einfluss auf den Textsentiment. Durch das Stemming und die Redundanzentfernung wurde in Datensatz eins die Anzahl von Attributen von 10450 auf 7614, in Datensatz zwei von 12860 auf 9058 reduziert. Danach wurden die Attributsmatrizen für den SVM Algorithmus erstellt. Für die Evaluierung werden die in der Klassifikation üblichen Kennzahlen Precision, Recall und F-Measure verwendet. Als Baseline Vergleichswerte werden die Experimente mit Preprocessing und ohne Preprocessing verglichen.

Als Ergebnis erhalten die Autoren bei dem Datensatz mit TF-IDF ohne Preprocessing 78.33 % und mit Preprocessing 81.5 %. Bei der Anwendung von FF erreichen sie in einem

Fall eine Accuracy von 83 % im Vergleich zu 72.7 % ohne Preprocessing. Zusammenfassend wurde aufgezeigt, dass mittels Preprocessing und SVM sehr hohe Accuracy Werte erzielt werden können, die vergleichbar sind mit Topic Modelling.

2.12 Predicting crime using Twitter and kernel density estimation

Es gibt seit einigen Jahren Ansätze das Data Mining auch zur Bekämpfung gegen Kriminalität anzuwenden. Es sollen Vorhersagen über zukünftige Verbrechen gemacht werden, um sie vor dem Geschehen zu vereiteln. Dazu sollen Vorhersagemethoden des Data Mining verwendet werden. Neben noch juristisch ungelösten Problematiken dieses Ansatzes (Ist es legal massenhaft Daten von Bürgern zu speichern und abzurufen, um dadurch etwaige Verbrechen zu vereiteln?), gibt es in Deutschland noch keine etablierte Software für diese Technologie. In den USA in Los Angelges beispielsweise wird das sogenannte *Predictive Policing* [11] bereits angewandt und es konnte eine Minimierung der Kriminalität in dem betroffenen Stadtteil um bis zu 25 % festgestellt werden.

Einen Beitrag zum Predictive Policing leistet Matthew S. Gerber mit seiner Arbeit *Predicting crime using Twitter and kernel density estimation* [21]. In seinem Experiment basierend auf Kernel Density Estimation[1] verwendet er Polizeidaten angereichert mit Tweetdaten. Er sucht eine Antwort auf die Frage, ob mittels Twitterdaten von US Bürgern mögliche lokale Verbrechen vorhergesagt werden können. Dazu stellt Gerber fest, dass eine der größten Hürden das Format, die verwendete Umgangssprache und andere textuelle Gründe sind. Auch sei es schwer ein Verbrechen auf einen Block mittels Twitter Geo-Daten einzugrenzen. Trotzdem werden so viele GPS-Daten wie möglich von den Tweets extrahiert, um so weit wie möglich das Gebiet eingrenzen zu können. Der Author beschränkt sich mit einem Filter auf die Stadt Chicago, da dort die zweitgrößte Mordrate in den USA vorherrsche. Ein anderer Grund für die Wahl von Chicago ist die weltweit wahrscheinlich größte Datensammlung an aktuellen und vergangenen Verbrechen in und um Chicago. Das Chicago Police Department hat nach aktuellem Verfassungsstand dieser Diplomarbeit 5.625.482 Crime-Records als Datensatz öffentlich online gestellt. Die Datensätze für Verbrechen beginnen mit dem Jahr 2001 und werden fast täglich mit gegenwärtigen Fällen aktualisiert. Die Daten können in vielen verschiedenen Dateiformaten heruntergeladen werden (`https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2`).

Gerber hat aus dem Datenportal alle Verbrechen zwischen Januar 2013 und März 2013 extrahiert (60.876 records). Parallel hat er für den selben Zeitraum sämtliche Tweets mit Filtereingrenzung (Koordinatendaten) und GPS-Daten nach den Stadtgrenzen vom Großraum Chicago gesammelt (1.528.184 Tweets). Als Trainingsdaten wurden die Chicago Police Department Datensätze für den gesamten Januar verwendet. Der Autor stellt

[1] kernel density estimation (KDE); auf Deutsch: Kerndichteschätzung; Dies ist ein Verfahren zur Schätzung der Wahrscheinlichkeitsverteilung einer Zufallsvariable

eine logistische Funktion[2] F und eine KDE Formel mit Feature Daten $f_2(p), ..., f_n(p)$ auf. Mit der KDE Formel berechnet er die historische Dichte, mit der Funktion F die Wahrscheinlichkeit eines Verbrechenstyps T an einer Lokation p. Die Features wurden aus den Twitternachrichten extrahiert. Damit beschreibt das Paper seinen Hauptbeitrag in der Vohersageforschung mittels Social Media, nämlich der Nutzung und Einbindung von den Zusatzfeaturefunktionen aus den Twitterdaten in die Standard KDE-Formel.

Insgesamt konnten durch die Arbeit von Gerber in 19 von 25 Verbrechensarten eine höhere Vorhersagegenauigkeit erzielen, wenn Twitterdaten und Features mitverwendet wurden. Der Autor stellt in Aussicht, dass die Verwendung dieser und verbesserter Methoden eine Hilfe in der Verbrechensbekäpmfung sein könnten.

2.13 Sentiment Analysis in Twitter using Machine Learning Techniques

Neethu und Rajasree haben in ihrer Arbeit Tweets zu Elektronikprodukten gesammelt und mit einer Stimmungsanalyse analysiert [35]. Um den Effekt von Domäneninformationen zu beaobachten, wurde bewusst eine spezifische Domäne selektiert. Die Autoren weisen daraufhin, dass die Performanz von Sentimentklassifizierern stark von dem Thema und dem Inhalt abhängt. Zunächst wird ein Preprocessing durchgeführt, woraufhin ein Feature Vektor erstellt wird. Als letzter Schritt werden diverse Algorithmen (u.a. SVM) zur Klassifizierung verwendet. Die Datensätze für Training und Test wurden mit der Twitter API gesammelt. Insgesamt wurden 1200 Tweets in Betracht gezogen. Diese wurden vom Autor manuell annotiert. Davon werden 1000 (500 negativ, 500 positiv für das Trainingsset und 100 positiv / 100 negativ für das Testset). Im Preprocessing werden URLs, Rechtschreibfehler und Slangausdrücke entfernt.

Der erwähnte Feature Vektor wird in zwei Schritten erzeugt. Zuerst werden Twitterspezifische Attribute extrahiert. Das sind zum Beispiel Hashtags und Emoticons. Diese werden entweder positiv oder negativ gelabeled. Im zweiten Schritt werden die Twittereigenen Features entfernt, um den Kurztext als normalen Text zu bearbeiten und erneut Features zu extrahieren. Anschließend werden SVM, Maximum Entropy, Naive Bayes und ein Ensemlbe Classifier verwendet.

Als Ergebnisse liefern die Autoren 89.5 % Precision für Naive Bayes und um die 90 % für SVM, Maximum Entropy und Ensemble Klassifizierer. Insgesamt wird festgestellt, dass mit dieser Art und Weise der Erstellung des Feature Vektors bei allen verwendeten Algorithmen ähnliche Ergebnisse erhalten werden.

[2] $F(f_1(p), f_2(p), ..., f_n(p)) \wedge f_1(p)$ als KDE und historische Dichtefunktion

3 Grundlagen

3.1 Machine Learning

Dieses Unterkapitel erklärt die Grundlagen des maschinellen Lernens. Die Stimmungsanalyse verwendet das maschinelle Lernen und muss daher begrifflich geklärt werden. Das Lernen allgemein kann folgendermaßen beschrieben werden:

Lernen ist jeder Vorgang, der ein System in die Lage versetzt, bei der zukünftigen Bearbeitung derselben oder einer ähnlichen Aufgabe diese besser zu erledigen. [1]

Diese Definition kann sowohl auf Maschinen als auch auf Menschen angewendet werden. Unabhängig vom System kann das Lernen in mehreren Schritten geschehen. Der erste Schritt ist die Kategorisierung aller Elemente, welche gemeinsame Merkmale aufweisen. Im nächsten Schritt müssen diese Kategorien spezifiziert werden, d. h. die Kategorien müssen abgegrenzt werden und die Anzahl der Merkmale muss minimiert werden. Wenn dies geschehen ist, kann von einem Begriff der Klassifikation gesprochen werden, bei der jedes Element einer anderen Menge von Kategorien zugeordnet (klassifiziert) wird.

Das System unterteilt den Lernprozess mittels *Klassifizierung* in zwei sequenzielle Abschnitte: Das *Trainieren* (Lernen anhand von Beispielen) und das *Testen*. Das Testen ist die Anwendung und Ableitung des Trainingswissens auf eine bislang unbekannte Datenmenge um diese zu klassifizieren.

Arthur Samuel definiert 1959 Machine Learning als „*Field of study that gives computers the ability to learn without being explicitly programmed*“[2].

Im Machine Learning spielt die Klassifikation eine große Rolle. Die Klassifikation verwendet Beispiele, die wiederum aus den *Beobachtungen* bzw. Messungen entstehen. In diesem Fall sind die Beobachtungen die Tweets ohne Label. Zunächst wird der Begriff der Beispielmenge formal definiert[3]:

Definition 1. (Beispielmenge):

Ein Beispiel der Form $e := (\vec{x}, y)$ *ist ein Tupel mit Merkmalsvektor* $\vec{x} \in X$ *mit Vektorraum* $X := \mathbb{R}^d$ *und Label* $y \in Y$ *aus einer endlichen Menge* L *von Labels. Die Menge* $E := \{b_1, ..., b_1\}$ *wird Beispielmenge genannt.*

[1] Simon 1983 aus Vorlesungsunterlagen Prof. Morik, TU Dortmund.
[2] Phil Simon (March 18, 2013). Too Big to Ignore: The Business Case for Big Data. Wiley. p. 89.
[3] siehe dazu auch Schlitt2010:Tag prediction in micro-blogging systems, TU Dortmund LS8

Falls $y_i \in \{1, -1\}, \forall i \in \{1, ..., N\}$, wird von einer *binären Klassifikation* gesprochen. Die Gesamtheit der Beispielmenge wird im Data Mining und in dieser Arbeit auch *Datensatz* genannt. Die Klassifikation kann die Beispielmenge zum Training und Test verwenden.

Die Klassifikation ist folgendermaßen definiert:

Definition 2. (Klassifikation):

Gegeben sei die Beispielmenge $(\vec{x}_1, y_1), ..., (\vec{x}_n, y_n)$. *Es gibt eine Klassifikationsfunktion*

$$h : X \to L,$$

so dass $h(\vec{x}_i) = y_i \; \forall \; (\vec{x}_i, y_i)$ *aus der Beispielmenge.*

H wird auch als Hypothesenfunktion bezeichnet, die also aus der Beispielmenge (Datensatz) gelernt wird. Jeder Eintrag im Datensatz ist ein Tupel der Form $(\vec{x}, y)$, wobei x ein Merkmalsvektor ist, welcher aus einer Reihe von Merkmalen (*Features*) besteht. Jeder Merkmalsvektor $\vec{x}$ wird dann einer Zielvariablen bzw. Klasse y zugeordnet.

Bei Lernalgorithmen gibt es zudem die Unterscheidung zwischen *Supervised Learning* (überwachtes Lernen) und *Unsupervised Learning* (unüberwachtes Lernen). Beim Supervised Learning sagt der Lernalgorithmus ein oder mehrere abhängige Variablen y zu einer Menge von unabhängigen Variablen bzw. Prädiktoren x_i voraus. Die mögliche Ausgabemenge ist also bereits bekannt. Bei der Stimmungsanalyse könnte dies beispielwese das Label $y = \{positive, negative\}$ sein. Mittels eines Trainingsdatensatzes wird ein Modell gebaut, das dann auf Testdaten angewendet werden kann. Dies ist bei der SVM Variante auch die Vorgehensweise in dieser Diplomarbeit.

Beim unüberwachten Lernen hingegen haben wir keine Zielvariable y mit bekanntem potenziellem Output. Es gibt also auch kein Vorhersagemodell wie beim überwachtem Lernen. Ziel ist vielmehr das Entdecken von Zusammenhängen und Mustern.

3.2 Sentiment Analysis

Sentiment Analysis oder die etwas weniger bekannte deutsche Bezeichnung *Stimmungsanalyse* ist eine Subdisziplin des Data Mining bzw. Text Mining und Opinion Mining (Im der folgenden Arbeit werden Sentiment Analysis und Opinion Mining synonym verwendet). Es gibt zahlreiche Arbeiten über das Opinion Mining und Text Mining. Allerdings gibt es bis heute keine einheitliche Zuordnung der Stimmungsanalyse und der Begrifflichkeiten, da diese Disziplin des Text Mining noch relativ jung ist.

Entstanden ist die Stimmungsanalyse aus den Bereichen des Natural Language Processing (NLP, siehe Kapitel 3.3) und dem Machine Learning. Die Stimmungsanalyse verwendet linguistische bzw. computerlinguistische Erkenntnisse, um Wissen in einer großen Datenmenge zu entdecken und diese mit einer Bewertung bzw. *Label* zu kennzeichnen. Die Daten für die Stimmungsanalyse können sehr lange Texte wie Produktbewertungen,

Filmkritiken, Blogeinträge oder aber kurze Texte wie Tweets auf Twitter oder YouTube Kommentare sein.

Eine übliche Definition von Opinion Mining und der Sentiment Analysis ist die Extraktion von positiven und negativen Meinungen aus unstrukturierten Texten gemäß Pang et al. in [40].

Oft genutzte Labels sind beispielsweise {*positiv, negativ, neutral*}, in manchen Fällen noch zusätzlich {*irrelevant*}. Diese Kennzeichnungen werden in der Stimmungsanalyse und im Weiteren synonym *Polarität* genannt. Andere komplexere Polaritäten und Stufen sind auch möglich und unterscheiden sich je nach Anforderung an die zu bewältigende Aufgabe. Es ist auch üblich eine Skala für die Polaritäten zu verwenden. In dem Fall gibt es beispielsweise ein Intervall von $[-10, 10]$, wobei -10 für sehr negativ und 10 für sehr positiv steht. Hier ist auch der Unterschied der Sentiment Analysis zur Klassifikation sichtbar. Während die Sentiment Analysis *subjektiv* ist und eine Bewertung des Textinhalts anstrebt, ist die Klassifikation *objektiv* und eindeutig. Die Detektion von Stimmung ist also die Auffindung von subjektiven Empfindungen und nicht Fakten. Subjektive Empfindungen können aber von Person zu Person stark variieren.

Bei einer Klassifikationsaufgabe wie der Zuordnung von Fahrzeugen mit bestimmten Eigenschaften zu einer Automarke beispielweise, ist die Automarke faktisch eindeutig, selbst wenn der Klassifikationsalgorithmus sich irrt und eine falsche Klassifizierung durchführt. Es gibt also keinen Spielraum was die Deutung angeht, ob ein Auto ein BMW oder ein Ford ist. Bei der Sentiment Analysis hingegen kann der Satz graduell anders interpretiert werden, je nachdem wieviel Bedeutung einzelnen Worten beigemessen wird. Das bedeutet in einem Algorithmus könnte ein Satz einen Score von $+4$ erhalten und ein anderer Algorithmus könnte dem selben Satz ein Sentiment von $+6$ vergeben. Es wäre schwer und eher eine linguistische Frage zu beeurteilen welcher Algorithmus richtiger liegt, falls man davon ausgeht, dass beide Algorithmen fehlerfrei funktionieren. Auch Menschen deuten Stimmungen subjektiv und nicht immer gleich.

Ein Beispiel für die Stimmungsanalyse ist ein großer Waschmittelhersteller, der wissen wollte, wie die Resonanz bei der Kundschaft auf das neue Produkt ist [44]. Dazu wurde Rapid-I, der Hersteller von Rapidminer, beauftragt. Als kostengünstige Methode wurde mittels Stimmungsanalyse mit Daten aus Internetquellen wie Foren die Stimmung und Resonanz der Kundschaft geprüft. Es wurde festgestellt, dass viele Kunden den Geruch des Waschmittels als unangenehm empfanden und der Hersteller konnte dank der Stimmungsanalyse sehr früh darauf reagieren und den Geruch anpassen. Die Stimmungsanalyse eignet sich also auch sehr gut für kommerzielle Interessen, um frühzeitig als Unternehmen zu erkennen, ob bezüglich eines Produkts Handlungsbedarf besteht. Die Stimmungsanalyse ist im Vergleich zu umfangreichen Marktanalyse von Agenturen deutlich günstiger.

In der Stimmungsanalyse gibt es zwei grundsätzlich verschiedene Wege der Bestimmung des Sentiments und der Polarität. Man unterscheidet zwischen *lernbasierten* Ansätzen [51] und *lexikonbasierten* Ansätzen [41]. Es gibt einige wissenschaftliche Abhandlungen,

die auch eine Hybridform verwenden und beide Verfahren in Experimenten je nach Bedarf kombinieren [34].

Lexikonbasierte Methoden basieren auf einem Wörterbuch, welches entweder bereits in vollständiger Form vorliegt, oder mittels eines sogenannten kleinen *Seed-Lexicons* und einem manuellem oder automatisiertem Erweiterungsmechanismus expandiert wird. In weiter entwickelten Lexicon-Verfahren werden zusätzlich Heuristiken, orthographische Regelwerke und Negationen erkannt. SentiStrength ist ein solches System 3.5.1.

Maschinelles Lernen bezeichnet generell lernbasierte Verfahren, welche Entscheidungen basierend auf *gelernten* Trainingsdaten treffen (siehe formale Definition in 3.1). Ausgehend von diesen Daten versucht dieser Ansatz beispielsweise Vorhersagen zu machen oder zu mindest auffällige Muster zu entdecken. Dieses maschinelle Lernen hat einige grundlegende Unterschiede zur lexikonbasierten Polaritätsklassifikation, da die sogenannten *Features* oder *Merkmale* und Eigenschaften eines Objekts in Vektorform eine wichtigere Rolle spielen, als die textuellen Vorkommen von Worten. Die Merkmale können qualitativ oder quantitativ sein. Qualitative Merkmale (auch nominale Werte genannt) sind endliche Mengen von Zuständen oder Eigenschaften wie z.B. Farben oder die Anzahl der Türen eines Autos oder aber der binären Polarität eines Textsentiments (positiv/negativ). Quantitative Merkmale (oder numerische Werte) können theoretisch unendlich sein und sind beispielsweise Werte wie die Absätze eines Produktes. Die Merkmale müssen im Gegensatz zur Stimmungsanalyse immer erst vektorisiert werden.

Lexiconbasierte Verfahren sind schnell und benötigen keine Trainingsdaten. Nachteile liegen in der Problematik der Erkennung von mehrdeutigen Worten, Umgangssprache und Sarkasmus. Außerdem muss immer ein Lexicon generiert werden oder bereits vorliegen.

3.3 NLP und Computerlinguistik

Die Sentimentanalyse verwendet viele Erkenntnisse der Linguistik bzw. Computerlinguistik. Die Erkennung von Wortarten (POS-Tagging bzw. Part-of-Speech Tagging) beispielsweise wird in der Sentimentanalyse ebenfalls verwendet und entstammt der Computerlinguistik. Daher ist es erforderlich einen kurzen Einblick in diese wissenschaftliche Disziplin zu geben, um die Stimmungsanalyse besser einordnen zu können. Die Computerlinguistik kann folgendermaßen definiert werden [26]:

> Computerlinguistik (engl. *Computational Linguistics*) ist jene Disziplin, die sich mit der maschinellen Verarbeitung natürlicher Sprache beschäftigt.

Nach dieser Definition ist das Natural Language Processing (NLP) im Weiteren ein fachsprachliches Synonym für die Computerlinguistik, wobei NLP in der Literatur nicht immer eindeutig der Computerlinguistik zugeordnet werden könne. Die Linguistik ist älter und der Ursprung des NLP. Die Linguistik allgemein ist die Disziplin der Sprachwissenschaften und hat das Ziel der Beschreibung und Erklärung natürlichsprachiger Phänomene [26].

Die Computerlinguistik hat folgende Aufgaben [42]:

- Theorie und Modellierung von Automatismen für die digitale Sprachverarbeitung
- linguistische Datenverarbeitung
- maschinelle Sprachverarbeitung (Realisierung natürlichsprachlicher Phänomene)
- Entwicklung von Sprachtechnologien und Software

Es sollte außerdem zwischen der angewandten Computerlinguistik und der theoretischen Computerlinguistik unterschieden werden. Während in der Anwendung konkrete Algorithmen gemeint sind, ist die theoretischen Computerlinguistik eine Teildisziplin der Linguistik, welche Modelle von natürlichen Sprachen entwickelt.

In der theoretischen Informatik hat die Computerlinguistik eine sehr wichtige Rolle, da die Automatentheorie ein Teilbereich dieser Wissenschaft ist. Die Automatentheorie wird in der Komplexitätstheorie und den Turingmaschinen verwendet.

3.4 Data Mining Tools

Data Mining beschreibt verschiedene Methoden zur Auffindung und Detektion von für den Kontext relevanten Erkenntnissen aus einer gegebenen (zu meist großen) Datenbasis. Laut einer aktuellen Studie des Analiystenhauses IDC [30] wird das gesamte Datenvolumen des digitalen Universums (Gesamtdigitaldaten der Erde) von 2005 bis 2020 von 130 Exabytes auf 40.000 Exabytes steigen. Das entspricht einer ungefähren Verdopplung alle zwei Jahre nach 2005. 1 Exabyte entsprechen einer Trillionen (10^{18}) Bytes oder 1 Millarde Gigabyte. In Anbetracht dieser stetig wachsenden Datenwelt steigt auch die Relevanz von Methoden und Werkzeugen, die wichtige Daten von den unwichtigen trennen können. Data Mining setzt genau an dieser Stelle an. Durch dieses immense Wachstum ist man auf Algorithmen angewiesen, die die benötigten Informationen aus der Gesamtdatenbasis extrahieren können - und das möglichst automatisiert. Die extrahierten Informationen müssen einen Vorteil für den Anwender erbringen.

Anhaltspunkte zur Auffindung wichtiger Informationen im Data Mining sind *Patterns* - auffällige Muster. Der Prozess sollte automatisiert oder zu mindest semi-automatisiert sein. Diese gefundenen Muster erlauben es, Vorhersagen über neue Daten zu machen. Unterschieden wird zwischen strukturierten und unstrukturierten Mustern. Strukturierte Muster erlauben es genauere Aussagen über den Inhalt zu machen.

Zur vereinfachten Unterscheidung der Begrifflichkeiten von Machine Learning 3.1 und Data Mining, kann das Data Mining als die Anwendung von Erkenntnissen aus dem Machine Learning betrachtet werden. Das Data Mining verwendet Algorithmen und Tools, die aus den Theorien des Maschine Learning abgeleitet werden und bezieht sich auf das Bearbeiten von großen Datenmengen (insbesondere Datenbanken), um daraus wichtige Erkennisse zu gewinnen. Machine Learning hingegen ist die Wissenschaft, aus einer Datenmenge neue Regeln und Gesetzmäßigkeiten zu lernen. Insbesondere das *Algorithm*

Engineering zur Problemlösung ist Bestandteil des Machine Learning. Es gibt in der Literatur allerdings keine einheitliche Unterscheidung der Begriffe.

3.4.1 RapidMiner

RapidMiner ist ein aktuelles in Praxis und Wissenschaft beliebtes Open-Source Data Mining System. Laut Entwickler ist es das derzeit am meisten verbreitete und verwendete Data Mining tool.[45]

RapidMiner wurde 2001 als YALE (Yet Another Learning Environment) von der TU Dortmund (LS Künstliche Intelligenz) entwickelt und 2007 in RapidMiner umbenannt. Es beeinhaltet alle gängigen Mining-Verfahren wie Text-Mining, Web-Mining etc. Die interne Darstellung in Standard XML ermöglicht Datenaustausch und Weiterverarbeitung. Da RapidMiner in Java geschrieben wurde, ist es weitestgehend plattformunabhängig und durch Plugins erweiterbar. Das WEKA Tool und die WEKA Bibliothek sind ebenfalls integriert worden.

RapidMiner stellt für das Data Mining sogenannte *Operatoren* bereit. Diese Operatoren haben oft einen oder mehrere Inputports und einen oder mehrere Outputports. Der Operator kann viele verschiedene Aufgaben erfüllen. Beispielsweise gibt es für jeden Typ von Support Vector Maschinen verschiedene Operatoren wie dem Standardoperator für SVM oder aber andere Implementierungen wie LibSVM. Die meisten Operatoren sind parametrisiert und erlauben Änderungen an den Parametern.

Es gibt regelmäßig neue Operatoren oder Extensions, die auf der Herstellerseite und dem Market Place heruntergeladen werden können[4]. RapidMiner ist in Java geschrieben und erlaubt das Implementieren eigener Operatoren in Java.

3.4.2 WEKA Framework

Das WEKA Framework ist eine Ansammlung von zahlreichen Data Mining Algorithmen und Methoden. Das WEKA Tool beeinhaltet eine große Menge an statistischen Lernverfahren, Methoden, Modellen und kann bis zu einem bestimmten Grad den eigenen Bedürfnissen angepasst werden können. Es ist möglich das WEKA Tool als .jar Ausführungsdatei zu verwenden oder in einer Entwicklungsumgebung(IDE) die WEKA Source-Datei einzubinden. Dadurch ist es möglich den WEKA Source-Code umzuschreiben oder eigene Algorithmen anzubinden. WEKA ist komplett in der Programmiersprache JAVA geschrieben und daher weitestgehend plattformunabhängig. WEKA ist Open-Source und wird von der University of Waikato entwickelt [38].

[4] `http://marketplace.rapidminer.com`

3.5 SentiStrength, Lingpipe, WordNet

3.5.1 SentiStrength

SentiStrength ist eine Software und Java API für die Stimmungsanalyse [49], die von Prof. Mike Thelwall entwickelt wurde. SentiStrength ist Teil des EU geförderten CyberEmotions Projekts [16], das im Februar 2009 ihren Anfang nahm und am 10 Juli 2013 abgeschlossen wurde. In dem Projekt CyberEmotions ging es um die kollektiven Emotionen in den sozialen Medien.

SentiStrength verwendet den lexikalen Ansatz der Textklassifikation von Inhalten. Das Programm gibt es als Windowsapplikation oder als Java Bibliothek und kann programmatisch in anderen Javaprogrammen verwendet werden. Für nichtkommerzielle Zwecke ist SentiStrength JAVA kostenlos.

SentiStrength wurde in der Vergangenheit bereits in einigen wissenschaftlichen Arbeiten für die Stimmungsanalyse verwendet. Es gibt auch einige bekannte kommerzielle Einsätze wie bei Yahoo! und bei den Olympischen Spielen 2012. Dort wurde Sentistrength zur Sentiment-abhängigen Echtzeitbeleuchtung des EDF Energy London Eye verwendet, dem größten Riesenrad Europas. Abhängig vom durchschnittlich berechneten Sentiment von Tweets über die Olympia wurde dort die Beleuchtung dynamisch angepasst. Zuletzt wurde beim Super Bowl 2014 das Empire State Building in New York City mit einer Sentiment-abhängigen Lichtshow beleuchtet[5]. Dabei sollten Twitternutzer vor den Spielen unter dem Hashtag *#WhosGonnaWin* zu ihren Teams twittern. SentiStrength wurde verwendet, um diese Tweets zu klassifizieren und die Stimmung zu extrahieren. Dynamisch und in Echtzeit wurde dann während des Spieltages die Lichtshow beim Empire State Building in Abhängigkeit der Daten beleuchtet.

Die Hauptaufgabe von SentiStrength ist die Berechnung des Sentiment Scores von Texten bzw. Kurztexten. Für gegebenen Input gibt Sentistrength standardmäßig folgenden Output aus:

-1(nicht negativ) bis -5(extrem negativ) 1(nicht positiv) bis 5(extrem positiv)

Dabei ist zu beachten, dass für jeden Text zwei Scores ausgegeben werden statt eines Gesamtscore. Damit wird dem entgegengesteuert, dass ein natürlicher Text sowohl negative als auch positive Sentiments simultan beeinhalten kann. Sollte diese duale Skala nicht erwünscht sein, ist es auch möglich, andere Skalen für den Output zu verwenden wie z.B. binär (positiv/negativ), trinär (positiv/negativ/neutral) oder Single Scale (-4 bis +4), wie sie auch in dieser Arbeit verwendet wird.

Insgesamt unterstützt SentiStrength bisher 14 Sprachen: Finnländisch, Deutsch, Holländisch, Spanisch, Russisch, Portugiesisch, Französisch, Arabisch, Polisch, Persisch, Schwedisch, Griechisch, Wälisch, Italienisch und Türkisch. Mittels SentiStrength ist es also auch möglich, manche fremdsprachigen Texte zu klassifizieren.

[5] `http://www.verizonwireless.com/news/article/2014/01/super-bowl-week-kickoff-empire-state-building.html`, Video:`https://www.youtube.com/watch?v=kvy8n2tLHV0`

SentiStrength hat insgesamt 2310 Sentiment Worte, die unter anderem vom Linguistic Inquiry and Word Count (LIWC) Programm stammen. In dem Lexikon bzw. der Nachschlagtabelle sind alle Stimmungsworte mit Scores versehen. Diese Scores wurden nach linguistischen Methoden von Menschenhand erstellt und basierten anfangs auf eineM Korpus von 2600 Kommentartexten von Myspace. In einer Evaluierungsphase wurden diese Scores von SentiStrength dann gegen zufällige neue Datensätze getestet. Dadurch eignet sich SentiStrength auch für Social Media Mining und Kurztexte aus dem Internet. Die Worte im Lexikon sind gewichtet und können im Rahmen einer lernbasierten Analyse auch angepasst werden[6].

Im Folgenden werden die Eigenschaften und Funktionsweise der zu verwendenden Bibliotheken von SentiStrength beschrieben. Eine detaillierte Beschreibung der in der Arbeit vorkommenden Funktionen ist wichtig, da viele Sentiment Scores sich aus verschiedenen Regeln zusammensetzen und diese dem Betrachter nicht widersprüchlich erscheinen sollen, denn Sentiments und Emotionen in Texten sind immer auch Teil von subjektiver Empfindung.

Sentistrength kann automatisch umgangssprachliche Texte und Redewendungen erkennen und diese entsprechend bewerten. Ein großer Vorteil ist für englische Analysen auch, dass Rechtschreibfehler oder bewusst künstlich in die Länge gezogene Worte wie *„niceeeeeeee“* von SentiStrength identifiziert und bewertet werden. In diesem Fall würde das Wort zu *„nice“* geändert werden. Zusätzlich würde der Score des Ursprungswortes um 1 erhöht werden. Der Grund für die Erhöhung ist, dass es in den sozialen Medien oft vorkommt, dass einem Emotionswort durch die Verlängerung und Wiederholung von Buchstaben mehr Gewicht verliehen werden soll. Das Wort *„niceeeeee“* würde also immer +1 mehr Score erhalten als das Wort *„nice“*. Das Wort „nicee“ hingegen würde keinen Sentiment boost von SentiStrength erhalten, da die Software bei einer einfachen Wiederholung des Buchstabens davon ausgeht, dass es sich um einen Tippfehler handelt und korrigiert diesen. Für den Sentiment boost muss der Buchstabe mindestens zweimal wiederholt werden. Dabei spielt es keine Rolle, wie oft der Buchstabe wiederholt wird, denn der Score würde immer nur um 1 erhört werden (aufgerundet von 0.6 Spelling Emphasis, siehe Beispiel).

Beispiel (Unäre Skalabewertung von 4 bis -4 wird in diesem Beispiel verwendet):

Testausdruck *nice*:

The Text 'nice' has scale result 2 . Approximate classification rationale: nice[3] [sentence: 3,-1] [result: max + and - of any sentence][scale result = sum of pos and neg scores] (Detect Sentiment)

Testausdruck *niceeee*:

The Text 'niceeeee' has scale result 3 . Approximate classification rationale: niceeeee[3] [+0.6 spelling emphasis] [sentence: 4,-1] [result: max + and - of any sentence][scale result = sum of pos and neg scores] (Detect Sentiment)

[6] `http://sentistrength.wlv.ac.uk/documentation/SentiStrengthChapter.pdf`

Hier ist in der SentiStrengthausgabe zu erkennen, dass der Algorithmus das Ursprungswort trotz Social Media Umgangssprache korrekt erkannt und diesem eine Erhöhung des Scores wegen Betonung (Spelling Emphasis) erteilt hat.

Ein anderer Vorteil von SentiStrength ist es, dass gewisse Worte als Booster (zusätzliche Adjektive) auch für die Score Berechnung in Betracht gezogen werden. Diese Worte tauchen in einer sogenannten **Booster Word List** auf und erhöhen den Score von nachfolgenden Worten, die eine Stimmung beeinhalten. Bei den Worten *extremely good* würde der Score des Wortes *good* um 2 erhöht werden. Bei den Worten *extremely bad* hingegen würde der Score niedriger werden.

Mittels einer Negativliste erkennt die Software auch Negierungen von Emotionen. Eine Negation vor einem Ausdruck bewirkt eine Vorzeichenänderung und ignoriert alle Worte der boosterlist, die zwischen der Negation und dem Sentiment Wort stehen.

Beispiel (Binärskala wurde in diesem Beispiel verwendet):

Testausdruck *very bad*:

The Text 'very bad' has binary result -1. (1 is positive, -1 is negative) Approximate classification rationale: very bad[-2] [-1 booster word] [sentence: 1,-3] [result: max + and - of any sentence][overall result = -1 as pos<-neg] (Detect Sentiment

Testausdruck *not very bad* liefert:

The Text 'not very bad' has binary result 1. (1 is positive, -1 is negative) Approximate classification rationale: not very bad[-2] [-1 booster word] [=0 negation] [sentence: 1,-1] [result: max + and - of any sentence][binary result = default value as pos=1 neg=-1] (Detect Sentiment)

Durch die Entsprechung von *„not“* mit einem Eintrag aus der Negativliste wurde hier das Vorzeichen des Ausdrucks umgekehrt. Aus dem Binäregebnis -1 (d.h. negatives Sentiment) wurde durch die Negation eine 1, also positiv.

SentiStrength kann auch die in den sozialen Medien und Twitter üblichen Emoticons bewerten. Die Sentimentstärke eines Ausdrucks wird durch das Emoticon beeinflusst. So kann die Verwendung eines traurigen Emoticons wie :(einen Score deutlich verschlechtern oder eine fröhliches Icon wie z.B. :) verbessern.

Es gibt eine Reihe anderer Regeln in SentiStrength wie z.B. dass der Gesamtscore um 1 erniedrigt wird, falls zwei starke negative Emotionsworte in Folge auftreten. Des Weiteren bewertet SentiStrength es als Sentiment Boost, falls in einem Satz mehrere Punkte und ein Ausrufezeichen folgen.

Zur Validierung der Funktionen und der Accuracy von SentiStrength wurden Tests durchgeführt. Bei den Tests ging es unter anderem um *Positive Strength Detection*. Dazu wurden 1041 neue MySpace Kommentare als Testset bestimmt. Es wurde die 10-fold cross-validation Methode mit extended feature set verwendet. Verglichen wurden im Experiment die Algorithmen aus Tabelle 3.1.

Algorithm	Optimal features	Accuracy	Accuracy +/- 1 class	Corr.	MAE
SentiStrength (standard configuration, 30 runs)	-	60.6%	96.9%	.599	22.0%
Simple logistic regression	700	**58.5%**	96.1%	**.557**	**23.2%**
SVM (SMO)	800	**57.6%**	**95.4%**	**.538**	**24.4%**
J48 classification tree	700	**55.2%**	**95.9%**	.548	24.7%
JRip rule-based classifier	700	**54.3%**	96.4%	**.476**	**28.2%**
SVM regression (SMO)	100	**54.1%**	97.3%	**.469**	**28.2%**
AdaBoost	100	**53.3%**	**97.5%**	**.464**	**28.5%**
Decision table	200	**53.3%**	96.7%	**.431**	**28.2%**
Multilayer Perceptron	100	**50.0%**	*94.1%*	**.422**	**30.2%**
Naïve Bayes	100	**49.1%**	**91.4%**	**.567**	**27.5%**
Baseline	-	**47.3%**	**94.0%**	**-**	**31.2%**
Random	-	**19.8%**	**56.9%**	**.016**	**82.5%**

Tabelle 3.1: Tabelle: Vergleich der Algorithmen [49]

Das Experiment zeigt, dass SentiStrength eine hohe Genauigkeit erzielt. Der einzige Algorithmus, der in dem Sentimentanalysetest sehr nah an SentiStrength herankommt ist Simple logistic regression.

Es ist zu beachten, dass es verschiedene Parameter in SentiStrength gibt und der Algorithmus daher stark variiert und angepasst werden kann. Es können beispielsweise diverse Texterkennungen und Bewertungen eingeschaltet oder abgeschaltet werden, um eventuell bessere Testergebnisse zu erzielen. Beispielsweise können im Falle von Texten, bei denen Emoticons ignoriert werden sollen, die Erkennung von Emoticons abgeschaltet werden. Oder die automatische Rechtschreibkorrektur mit anschließender Sentimentbewertung kann deaktiviert werden. In einer Auflistung ([49], S.18f) werden von Thelwall die unterschiedlichen SentiStrength Variationen aufgelistet und verglichen. Die Ergebnisse der Tabelle enstanden durch Tests auf positiven Sentiments als Durschnitt von dreißig 10-fold cross-validations gegen 1041 MySpace Kommentare.

Es ist zu beobachten, dass SentiStrength in der Standardvariante die besten Ergebnisse erzielt. In der Standardvariante liefert der Algorithmus mit 21.66 % den 3. niedrigsten Mean Square Error (MAE) bei der höchsten Anzahl an korrekten Klassifikationen (61.03 %). Thelwall führte selbigen Test auch zugeschnitten auf negative Sentiments durch und kam wieder zu dem Ergebnis, dass die Variationen nur minimal unterschiedliche Ergebnisse lieferten.

Insgesamt variieren die Ergebnisse der unterschiedlichen SentiStrength Algorithmen also nicht sehr stark und es kann im Rahmen dieser Arbeit weitestgehend ohne größere Performanzeinbußen mit der Standardkonfiguration von SentiStrength gearbeitet werden.

SentiStrength kann evaluiert werden, indem ein manuell gelabelter Korpus erstellt wird und SentiStrength Ergebnisse gegen diesen Korpus geprüft werden. Der Entwickler von SentiStrength schlägt dabei vor, einer Gruppe von Menschen jeweils 100 Textabschnitte zu geben und von den 3 ähnlichsten Bewertern die Durchschittsscores zu verwenden. Als Vergleichsmetrik sei die Pearson Korrelation am besten geeignet.

Für diese Arbeit wurde die SentiStrength .jar eingebunden und wird mit einigen Parametern aufgerufen. Beispielsweise wurde der *Idiomlist* eine höhrere Priorität zugewiesen, so dass in einem Tweet der Gesamtscore mit der Polarität des Idioms überschrieben wird. Dadurch wird gewährleistet, dass einzelne Worte in einer Redewendung nicht zu einer Fehlklassifikation führen, weil die Gesamtpolarität eine andere ist.

3.5.2 Lingpipe

Lingpipe ist eine quelloffene Java API zur Textklassifikation mittels Computerlinguistik [32]. Lingpipe gehört zur Familie der Machine Learning Klassifikationsalgorithmen, wohingegen SentiStrength den lexikalen Ansatz verfolgt. Das heißt, Lingpipe erwartet das Antrainieren eines Trainingsdatensatzes mit gelabelten Daten, um basierend auf diesen Trainingsdaten Vorhersagen bzw. Klassifikationen auf einem ungelabeltem Testdatensatz zu führen. Lingpipe ist für den nicht-kommerziellen Einsatz kostenlos (Daten müssen verfügbar gemacht werden).

Lingpipe ermöglicht dem Anwender eine automatische Klassifikation von Suchergebnissen bei Twitter oder die Entdeckung von bestimmten Tokens in Texten. Der Vorteil von Lingpipe ist, dass die API mächtig ist und sich für verschiedene Domänen, Sprachen, Genres eignet. Folgende Funktionen werden von Lingpipe bereitgestellt: Tokenization, POS Tagging, Named Entity Detection, Topic Classification, DB Text Mining, Spell Checker, Sentiment Analysis, Sentence Detection und Language Detection.

Lingpipe wird intensiv im akademischen und militärischen Bereich genutzt. Die Autoren werden unter anderem vom U.S. Department of Defense für diverse Forschungsarbeiten subventioniert und beauftragt. Eines der entwickelten Softwaretools mit Lingpipe ist beispielsweise der *ThreatTracker* zur Gefahrenerkennung (`http://alias-i.com/lingpipe/web/customers.html`) oder der *Osama bin Laden Tracker*, der weltweit von Nachrichtendienstmitarbeitern und Regierungen zur Terroristenerkennung genutzt wurde. Auch wurde Lingpipe bei einer Software zur Ausbrucherkennung von ansteckenden

Krankheiten genutzt. Insgesamt wird die Java API von vielen renommierten Instituten im professionellen Bereich verwendet.

Da Lingpipe für die Textklassifikation eine Trainingsphase benötigt, eignet sich die API nur für einen lernbasierten Ansatz. Folglich wird sie im Rahmen dieser Arbeit als Alternative zu bekannten lernbasierten Klassifikationsalgorithmen wie SVM, Naive Bayes betrachtet, und nicht verwendet werden. Lingpipe wird wegen der Bekanntheit in der Textklassifikation zwecks Vollständigkeit erwähnt.

3.5.3 WordNet und SentiWordNet

WordNet ist eine in den USA von der National Science Foundation geförderte semantische lexikale Datenbank der englischen Sprache. Die Besonderheit von WordNet ist, dass die verfügbaren Worte in einer hierarchischen Form in der Datenbank gehalten werden. Nomen, Verben, Adjektive und Adverben werden in der Datenbank derart verknüpft, so dass Worte mit ähnlichen Bedeutungen zu sogenannten *Synsets* zusammengefasst werden – eine Art Synonymgruppe. Diese Synsets sind untereinander verknüpft und können mit einem Browser durchforstet werden. Dadurch eignet sich WordNet vor allem für computerlinguistische Arbeiten bzw. Natural Language Processing (siehe Kapitel 3.3). Inzwischen gibt es WordNet auch für andere Sprachen (unter anderem für die deutsche Sprache durch GermaNet)[7].

SentiWordNet ist eine große Wort-Datenbank mit Zusatzinformationen zu den lexikalen Ressourcen aus WordNet. Diese Zusatzinformationen sind die sogenannten Sentiment Scores bzw. Polaritäten wie Positivity, Negativity und Objectivity. Das bedeutet, dass in SentiWordNet alle Wörter in der englischen Sprache bereits wortweise ein Sentiment enthalten und dadurch in der Sentimentanalyse verwendet werden können. Durch diese Annotationen können Text-Datensätze ohne Trainingsphase direkt mit einer Polarität klassifiziert werden ([9] und [19]). SentiWordNet kann in Verbindung mit WordNet zur automatischen Lexicon-Expansion verwendet werden. Eine ähnliche Methode wird in Kapitel 6.1.4 gezeigt.

3.6 Support Vector Machine (SVM)

Dieses Kapitel erklärt die Stützvektormethode. Sie findet oft Anwendung bei Stimmungsanalysen und anderen Klassifikationsproblemen. SVM wird auch in dieser Arbeit als Teil der hybriden Analyse[8] verwendet und soll daher in der Funktionsweise erklärt werden. In der Arbeit wird das in RapidMiner implementierte LibSVM als Operator verwendet[9]. Einige SVM Beschreibungen wurden aus der Lehrveranstaltung

[7] `http://www.sfs.uni-tuebingen.de/GermaNet/`

[8] Gemeint ist hier das Verwenden von Trainingsdaten für SVM, welche mit Lexiconmethoden erstellt wurden.

[9] Der Standard SVM Operator erkennt nur 2 Klassen. LibSVM erlaubt mehr als 2 Klassen. Allerdings führte dies in den Experimenten zu einer Verschlechterung der Genauigkeit. Näheres dazu im Kapitel Experimente.

„Knowledge Discovery in Databases“ von Prof. Morik entnommen. Diese sind unter `http://www-ai.cs.uni-dortmund.de/LEHRE/VORLESUNGEN/KDD` einsehbar.

Die Support Vector Machine wird in der Literatur unter Anderem auf die Theorien von Vladimir Vapnik und Alexey Chervonenkis zurückgeführt[10]. In der Arbeit von Vapnik et al. wird die sogenannte Vapnik-Chervonenkis (VC) Dimension definiert. Die VC-Dimension ist das Maß der Kapazität eines Klassifikators bzw. die Abschätzung des Risikos, die sich als Summe des empirischen und des strukturellen Risikos ergibt[11].

Das Support Vector Machine Modell wird in dieser Arbeit als 2-Klassenproblem verstanden. Es ist auch möglich SVMs für N-Klassenprobleme zu verwenden. In dieser Arbeit wird ein 3-Klassenproblem mit SVM behandelt. Siehe dazu das Unterkapitel **SVM Experimente**(7.2).

Eine SVM trennt mit einer Hyperebene Merkmalsvektoren derart, so dass möglichst viele zusammengehörige Merkmalsvektoren auf der selben Seite liegen. Das Ziel der SVM ist die Klassifikation dieser Vektoren anhand ihrer Lage zur trennenden Hyperebene. Es handelt sich hierbei um ein Optimierungsproblem, da die **optimal** linear trennende Hyperebene gesucht wird. Das Kriterium dieser Optimalität ist die Identifikation genau **der** Hyperebene H aus der Anzahl der unendlich vielen Hyperebenen, welche den Abstand d (*optimal margin*) zu den nächstgelegenen Merkmalsvektoren bzw. *Stützvektoren*(*support vectors*) $\vec{x}_i$ maximiert. Idealerweise gibt es dann jeweils einen vektorfreien Trennbereich zwischen der Trennebene und den beiden getrennten Bereichen mit den 2 verschiedenen Klassen. Dieses beschriebene Prinzip ist in Abbildung 3.1 dargestellt.

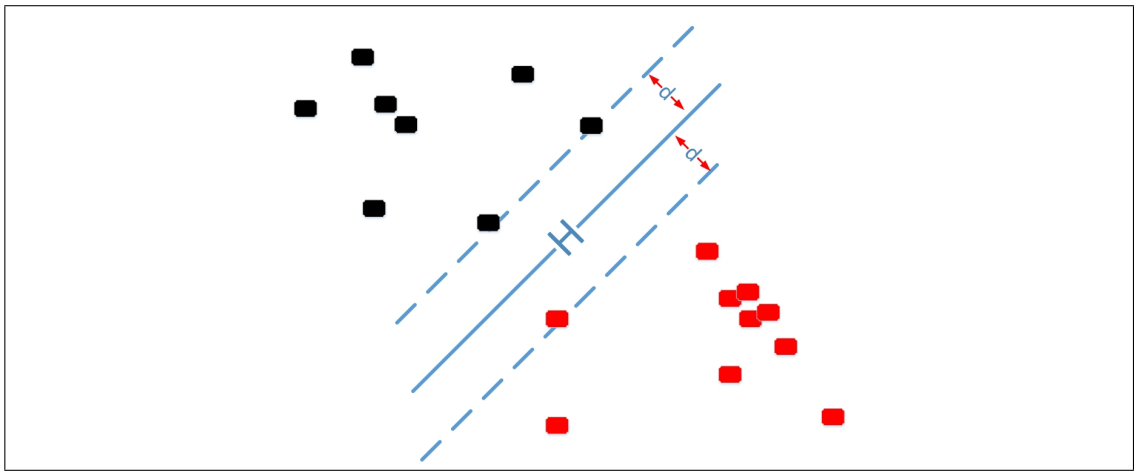


Abbildung 3.1: SVM: Hyperebene H mit margin Distanzen d

Es existiert eine eindeutige optimale Hyperebene der Form:

$$H := \{\vec{x} | \langle \vec{w}, \vec{x} \rangle \ + b = 0\} \tag{3.1}$$

[10] V. Vapnik, A. Chervonenkis: Pattern Recognition Theory, Statistical Learning Problems, Nauka, Moskau, 1974. Arbeit wurde später in das Englische übersetzt.

[11] Für Details zu VC-Dimension siehe: Prof.Fink TU Dortmund in LV Mustererkennung 2014, Skriptum zur Vorlesung “Mustererkennung”, S.81

$\vec{w}$ ist der Normalenvektor zur Hyperebene H, $\vec{x}$ ist ein Merkmalsvektor. b bestimmt den Abstand zur Hyperebene. Unter der Annahme der linearen Separierbarkeit gilt:

$$\langle \vec{w}, \vec{x} \rangle \ + b \begin{cases} > 0 & \text{falls Merkmalsvektor im positiven Raum} \\ = 0 & \text{falls Merkmalsvektor auf H} \\ < 0 & \text{falls Merkmalsvektor im negativen Raum} \end{cases} \tag{3.2}$$

Der Abstand zum Ursprung ist bestimmt durch $\frac{b}{\|w\|}$, wobei $\|w\|$ die Länge des Vektors $\vec{w}$ ist. Der jeweilige Abstand der zur trennenden Hyperebene nächstgelegenen positiven und negativen Beispiele ist der *margin* und beträgt insgesamt $\frac{2}{\|w\|}$. Zur Auffindung der optimalen eindeutigen Hyperebene muss dieser margin maximiert werden. Dazu kann $\|w\|^2$ minimiert werden. Es entsteht also das Optimierungsproblem:

$$\min \|w\|^2 \tag{3.3}$$

mit der Nebenbedingung:

$$\forall (\vec{x}_i, y_i) \in E : y_i(\langle \vec{w}, \vec{x} \rangle \ + b) - 1 \geq 0 \tag{3.4}$$

Nach der Bestimmung der eindeutigen Hyperebene, kann diese dazu verwendet werden, bisher ungesehene neue Beobachtungen zu klassifizieren[12].

3.6.1 Schlupfvariable

Die Sätze 3.3 und 3.2 gelten unter den Annahmen der linearen Separierbarkeit der vorliegenden Datengrundlage E. Allerdings sind in der Praxis (häufig auch verstärkt durch *Rauschen* oder *Ausreißer*) die Daten nicht linear trennbar. Zur Überwindung dieser Problematik gibt es zwei Ansätze. Der erste, simplere Ansatz ist die Einführung einer Schlupfvariablen (*slack variable*) ξ. Die Schlupfvariable toleriert einen gewissen Fehler, so dass nicht eindeutig linear trennbare Daten dennoch mittels SVM klassifiziert werden können. Dadurch wird eine weich trennende Hyperebene erzeugt. Das Optimierungsproblem sieht dann umgeformt mit der Schlupfvariablen folgendermaßen aus:

$$\min \|w\|^2 + C \sum_{i=1}^{n} \xi_i \text{ für ein festes } C \in \mathbb{R}_{>0} \tag{3.5}$$

mit der neuen Nebenbedingung:

$$\forall (\vec{x}_i, y_i) \in E : y_i(\langle \vec{w}, \vec{x} \rangle \ + b) \geq 1 - \xi_i \ \forall i \in \{1, ..., N\}$$
$$\text{mit } \xi_i \geq 0 \ \forall i \in \{1, ..., N\}$$

Durch Gleichung 3.5 können nun weichere Hyperebenen mit Fehlerklassen erzeugt werden.

[12] Das Optimierungsproblem des margins kann durch die Lagrange-Funktion gelöst werden. Darauf wird in dieser Arbeit aber nicht weiter eingegangen. Siehe dazu `http://www-ai.cs.uni-dortmund.de/LEHRE/VORLESUNGEN/KDD/SS08/`

3.6.2 Kernel-Trick

Daten liegen in der Praxis oft mit Rauschen, Ausreißern u.Ä. vor. Das Verwenden einer Fehlertoleranz für die Klassifizierung reicht in vielen Fällen nicht, um die Separierbarkeit zu erhöhen. Zur Behandlung dieses Problems gibt es den sogenannten Kernel-Trick bei SVMs. Beim Kernel-Trick handelt es sich um eine Kernel-Funktion, welche die gegebene ursprüngliche Inputdimension $\mathbb{R}^p$ zu einer höheren Zieldimension $\mathbb{R}^{p+k}$ im Hilbert-Raum mit $k \in \{1, ..., n\}$ überführt. Die Intention hierbei ist, dass zuvor nicht trennbare Beispiele dadurch besser trennbar werden. Dieser Prozess wird auch Φ Transformation genannt, wobei Φ eine wählbare Kernel-Abbildungsfunktion ist (*mapping*). Der Kernel $K(x, x')$ entspricht dem Skalarprodukt $(\vec{x}, \vec{x'})$.

Definition 3. (Kernel Funktion):

Merkmalsvektor $(\vec{x}_1, \vec{x}_2)$ *und die Abbildung* $\Phi : X \to K$ *sind gegeben, wobei* K *eine höhere Dimension ist. Die Funktion* $k(\vec{x}_1, \vec{x}_2) = \Phi(\vec{x}_1), \Phi(\vec{x}_2)$ *wird Kernel Funktion genannt.*

Ein Beispiel für den Kernel-Trick ist in Abbildung 3.2 zu sehen. Durch die Projektion der Dimension in die höhere Ebene werden die Datenpunkte trennbar. Nach der Trennung können die Daten wieder in die Inputdimension zurückprojeziert werden.

Es gibt einige oft verwendete Kernel-Funktionen, die auch von Rapidminer untersützt werden. Ein bekannter Kernel ist die Radialbasisfunktion (RBF) mit

$$K(\vec{x}_i, \vec{x}_j) = exp(-\gamma \left\| \vec{x}_i - \vec{x}_j \right\|^2)$$

und die Polynomfunktion

$$K(\vec{x}_i, \vec{x}_j) = \langle \vec{x}_i, \vec{x}_j \rangle^d .$$

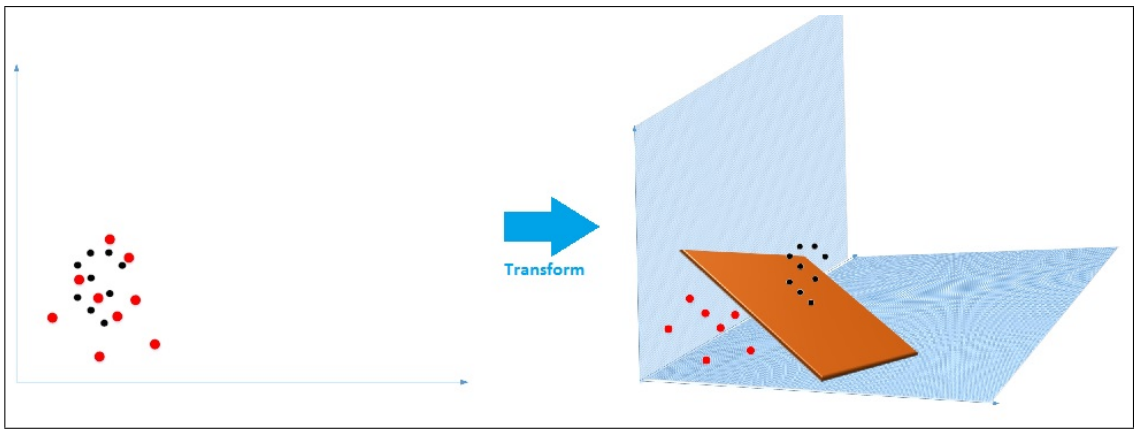


Abbildung 3.2: Kernel-Trick

In dieser Arbeit wurden experimentell beide Varianten in der verfügbaren Rapidminer Variante ausprobiert.

3.7 Twitter und die Twitter API

Die Twitter API ist eine Programmierschnittstelle, die von Twitter bereitgestellt wird. Mit der Twitter API ist es Entwicklern möglich, auf viele Programmierfunktionen in Twitter zuzugreifen. Es kann auf vergangene Tweets, auf aktuelle Trends, Benutzerinformationen und andere Kernfunktionen zugegriffen werden. Unter der Twitter API sind generell mehrere APIs gemeint, die unter diesem Begriff zusammengefasst werden. Die darunterliegenden APIs sind:

- Twitter Search API: Beeinhaltet Query-Funktionen
- Twitter REST API: RESTful Schnittstelle
- Twitter Streaming API: Schnittstelle für Big Data

Mit der Search API können in anderen Applikationen Suchfunktionen über Twitter bereitgestellt werden. Anwender können dadurch nach Schlüsselwörtern oder anderen Benutzern suchen.

Die Twitter REST API ist eine Schnittstelle, die auf einem Programmierkonzept für Anwendungen im world wide web basiert. REST steht für *Representational State Transfer* und wird als Abstraktion des Aufbaus des World Wide Web verstanden. Mittels REST wird versucht, eine Standardisierung von Webanwendungen in Bezug auf die Art und Weise der Adressierbarkeit und Programmierung zu erreichen. Es gibt einige Grundprinzipien, die eine REST Applikation wie z.B. die Twitter REST API aufweisen muss. Diese sind unter anderem Skalierbarkeit, Einfachheit, Zuverlässigkeit, Portabilität und andere. Anwendungen und Dienste, die die REST Konzepte und Schnittstellen verwenden, werden auch als RESTful bezeichnet. Die HTTP Standardmethoden, die eine RESTful Anwendung bieten muss, sind z.B. GET, POST, PUT, DELETE. Die Twitter REST API liegt derzeit in der Version 1.1 vor und erlaubt hauptsächlich GET und POST Operationen.

Beispiel:

```
GET https://api.twitter.com/1.1/statuses/
user_timeline.json?screen_name=twitterapi&count=2
```

Hier wird mittels einer GET-Anfrage die Timeline (also die öffentliche Tweetliste) des users *twitterapi* [52] geholt. Rückgabewerte sind im JSON Format und beeinhalten alle Felder, die Twitter für Tweets bereitstellt. Es werden z.B. unter anderem folgende Felder im JSON Format zurückgegeben:

```
 "in_reply_to_user_id_str": null,
    "contributors": null,
    "Text": "Introducing the Twitter Certified
Products Program: https://t.co/MjJ8xAnT",
    "retweet_count": 121,
```

```
    "in_reply_to_status_id_str": null,
    "id": 240859602684612608,
    "geo": null,
    "retweeted": false,
    "possibly_sensitive": false,
    "in_reply_to_user_id": null,
    "place": null,
    "user": {
      "profile_sidebar_fill_color": "DDEEF6",
      "profile_sidebar_border_color": "C0DEED",
      "profile_background_tile": false,
      "name": "Twitter API",
      "profile_image_url": "http://a0.twimg.com/profile_images
/2284174872/7df3h38zabcvjylnyfe3_normal.png",
      "created_at": "Wed May 23 06:01:13 +0000 2007",
      "location": "San Francisco, CA",
      "follow_request_sent": false,
      "profile_link_color": "0084B4",
      "is_translator": false,
      "id_str": "6253282"
```


Die Rückgabe in JSON kann entweder vollständig in einer Datenbank abgelegt werden oder es können bestimmte Felder gespeichert werden. Die wichtigsten Felder sind *Text* für den eigentlich Tweetinhalt, *retweet_count* für die Anzahl an Retweets und das *created_at* Feld mit dem Erstelldatum des Tweets. Dieses letzte Feld ist beispielsweise für Zeitreihenanalysen wertvoll.

Die Antwort der REST-Anfrage wird auch als *REST Ressource* bezeichnet. Es können in jeder Anfrage diverse Parameter mitgegeben werden. Es gibt insgesamt sehr viele Anpassungsmöglichkeiten, so dass Twitter auch programmatisch komplett gesteuert werden kann. Durch die Bereitstellung der REST Schnittstelle können eigene Applikationen entwickelt werden, die diese Schnittstelle verwenden.

Die Twitter Streaming API ist eine weitere Schnittstelle, die Twitter anbietet. Es gibt drei Kategorien von Twitter Streams:

- Public Streams für öffentliche Tweets, Hashtag topics usw.
- User Streams für benutzerspezifische Streams
- Site Streams für mehrfache Anfragen von Benutzern

Die Twitter Streaming API unterscheidet sich grundsätzlich stark von den anderen Twitter APIs, da die Streaming API Echtzeitinformationen abruft. Anstatt einem einzelnen GET-Request eine Antwort zu senden, beantragt der Client eine persistente Stream-Verbindung. Der Twitter Server verarbeitet die Anfrage des Clients und sendet sie als Antwort. Mit dieser API können in Echtzeit Tweets angezeigt oder extrahiert werden. Sie wird insbesondere für datenintensive oder statistische Aufgaben verwendet, wie z.B.

für Data Mining Analysen. Es können auch mit dieser API nach Schlüsselwörtern oder beispielsweise Sprachen gefiltert werden.

Im Vergleich zu den anderen APIs erlaubt die Streaming API ein höheres Datenaufkommen an Tweets. Es ist jedoch vor allem bei wissenschaftlichen Arbeiten darauf zu achten, dass die Streaming API von Twitter aus per Policy gedrosselt wird. Das liegt daran, dass Twitter ein Datenverkäufer ist und nicht möchte, dass es möglich ist, sämtliche Twitterdaten kostenlos zu extrahieren. Standardmäßig kann ein *sample* von maximal 1 % aller weltweit verschickten Tweets gestreamt werden. Diese Obergrenze wird aber nur selten erreicht, falls nach bestimmten Schlüsselworten oder anderen Attributen gefiltert wird. Falls es doch zum Erreichen der 1 % kommt, gibt die API eine Drosselungsnachricht. Der ungedrosselte Twitterstream wird von Twitter auch als *fire hose* bezeichnet. Für akademische Arbeiten kann ggf. per email[13] an Twitter eine Erlaubnis erfragt werden, damit der *garden hose* Zugang mit einer Obergrenze von 10 % verwendet werden kann. Twitter prüft dann diese Anfrage und vergibt eventuell den Zugang. Im Rahmen der vorliegenden Diplomarbeit wurde ein akademischer Zugang in Betracht gezogen aber nicht angefragt, da es bis auf wenige Ausnahmefälle nicht vorkam, dass die 1 % Grenze erreicht wurde. Derzeit werden täglich mehr als 500.000.000 Tweets gesendet. 1 % aller Tweets würde also einer täglichen Obergrenze von 5 Millionen Tweets entsprechen.

Die Twitter Streaming API wurde in dieser Diplomarbeit für die Erfassung und Extraktion der Datenbasis verwendet. Dafür wurden vom 29.06.2014 bis zum 05.08.2014 insgesamt mehr als 1.500.000 Tweets aus dem Twitter *fire hose* gefiltert extrahiert. Der Stream in Twitter wird in der API Dokumentation auch als *Statusstream* bezeichnet. Nähere Details zu dem Extraktionsverfahren sind im Kapitel *Extraktion der Datenbasis* 4 zu entnehmen.

3.8 Twitter4j

Twitter4j steht für *Twitter for java* und ist eine inoffizielle Javabibliothek für die Twitter API [55]. Mit Twitter4j wird der Umgang mit der Twitter API erleichtert, da die Programmierbibliothek als Wrapper für die Twitter API fungiert und viele Funktionen zugänglicher gemacht wurden. Es genügt die Jar-Datei von Twitter4j herunterzuladen und zum CLASSPATH hinzuzufügen. Twitter4j liegt aktuell in der Version 4.0.2 vor und unterstützt den gesamten Twitter API 1.1 Umfang[14]. Der gesamte Umfang von Twitter4j kann in den javadocs eingesehen werden: `http://twitter4j.org/javadoc/twitter4j/`. Einige Code Beispiele aus der Praxis (Auszug von der Streamextraktion mittels Twitter4j Bibliothek und Twitter API 1.1.):

[13] api-research@twitter.com

[14] Während dieser Diplomarbeit wurde die Version der Programmierbibliothek im Oktober 2014 von der Version 3.0.5 auf 4.0.2 erhöht. Durch umfangreiche Änderungen in den Kernfunktionen der Twitter API durch Twitter und Änderungen von twitter4j kam es dazu, dass einige bereits implementierte Streamingkomponenten in dieser Diplomarbeit nicht mehr funktionierten. Es wurde daher auf die aktuelle Version migriert und einige Funktionen umgeschrieben.

```
  FilterQuery fq = new FilterQuery();
        String keywords[] = {movie1, movie1b, movie2,
movie3, movie4, movie4b};
        //String language[] = {"German"};

        fq.track(keywords);
        //fq.follow(users);    //collects tweets by these users
        twitterStream.addListener(listener);
        twitterStream.filter(fq);

....

 try {
       String tweet_trim =
  content.replace("\n", "").replace("\r","");
       String str_id =
 String.valueOf(status.getId());
       String str_follower =
 String.valueOf(status.getUser().getFollowersCount());
       String str_friends =
 String.valueOf(status.getUser().getFriendsCount());
       String str_retweets =
 String.valueOf(status.getRetweetCount()); }

...
```

In diesem Codeauszug wird im ersten Teil eine *Filterquery* definiert. Dieser dient zur parametrisierten Filterung des Gesamtstreams des Twitter *fire hose*. Wenn die Filterquery mit leeren Schlüsselwörtern ohne jegliche Filterung aufgerufen würde, würde zur Laufzeit des Programms binnen weniger Minuten der Zugang zum fire hose stream von Twitter gesperrt wegen, da in 10 Minuten ca. 4.000.000 Tweets im Stream durchlaufen und das die 1 % Grenze für User überschreiten würde (>500 Mio. Tweets pro tag / 24h x 60m). Der *Filterquery* Konstruktor hat verschiedene Signaturen. Mit der längsten Parameterliste kann sie folgendermaßen aufgerufen werden:

```
public FilterQuery(int count,
                   long[] follow,
                   java.lang.String[] track,
                   double[][] locations,
                   java.lang.String[] language)e
```

Im Code der Diplomarbeit wird der Parameter *follow* nicht gesetzt, da dadurch nur der Tweetstream eines bestimmten Nutzers extrahiert würde. Die restlichen Parameter wurden voll genutzt. Es wurde ein Stringarray von Filmen erzeugt und diese dem Pa-

rameter *track* übergeben, um die Streams auf dieses Array zu begrenzen. Die restlichen Methodenaufrufe und Verfahren zur Extraktion werden im Kapitel 4 erklärt.

4 Extraktion der Datenbasis

Es gibt verschiedene Möglichkeiten Daten zu extrahieren. Selbst im Bereich der sozialen Medien gibt es bereits viele unterschiedliche APIs und Programmierbibliotheken. Die großen Social Media Plattformen wie YouTube und Facebook bieten eigene Programmierschnittstellen an. Mittels der YouTube API beispielweise lassen sich hunderte Kommentare mit einigen wenigen Zeilen Javacode extrahieren.

Auch Twitter bietet Schnittstellen zur Datenextraktion an. Da allerdings Twitter auch offiziell Twitter Daten zum Data Mining an einige Unternehmen verkauft, sind diese Schnittstellen aus kommerziellen Gründen teilweise volumenbegrenzt (siehe dazu auch *Twitter und die Twitter API* in Kapitel 3.7). Im Rahmen dieser Diplomarbeit hatte diese Volumenbegrenzung aber nur minimalen Einfluss auf die Datenextraktion, da hauptsächlich die Streaming API mit *Filterquery* verwendet wurde, und die von Twitter vorgegebene Obergrenze des *Fire Hose* dadurch selten erreicht wurde.

Anfangs war es über die Twitter API möglich *since-until* Anfragen mit einer beliebigen Zeitspanne bis hin zum Startdatum von Twitter zu senden. Der Server gab dann sämtliche Tweets zu der angegebenen Zeitspanne zurück. Die Serverantwort war beschränkt auf 1000 Tweets. Allerdings konnte diese Beschränkung mit ein wenig Programmierarbeit umgangen werden. Twitter hat inzwischen die *since-until* Anfragen stark beschnitten, so dass nur maximal 7 Tage vom gegenwärtigen Datum zurückliegende Tweets in der Serverantwort berücksichtigt werden. Anfragen über diese 7 Tage hinaus führen zu einer leeren Antwort vom Server. Dadurch ist es nicht mehr möglich historische Tweets zu extrahieren. Dies ist eine Hürde bei der Erstellung eines Korpus, da zur Erstellung einer großen Datenbasis dann nur noch die Streaming API in Frage kommt, die für einen längeren Zeitraum laufen muss. Dies setzt einen ständigen Online-Betrieb voraus. Wegen dieser Einschränkung war es in dieser Arbeit teilweise nicht möglich, für einige der Filmdatensätze Tweets über einen längeren Zeitraum zu beschaffen.

4.1 Datenkollektion

In dieser Diplomarbeit wurden zwei verschiedene Ansätze bei der Extraktion getestet. Als erste Variante wurde statt der Streaming API von Twitter die Twitter Search API mit Searchqueries verwendet. Die Ergebnisse dieser Searchqueries wurden dann direkt in einer MS SQL Server Datenbank abgelegt. Wegen der Beschränkungen der programmierbaren Searchqueries wurde dann die Streaming API angewandt. Allgemein wurde der gesamte gefilterte Twitterstream in einem CSV Format abgespeichert. Der Vorteil des CSV Formats lag hier in der Einfachheit der Verwaltung und weiteren Verwendung. Auch

ist es innerhalb einer CSV Datei egal, welchen Datentyp das Feld beeinhaltet. Die Felder müssen nicht vorher bezüglich ihrer Datenformate definiert werden. Es können beliebige Formate extrahiert werden. CSV Dateien können außerdem sowohl in WEKA, als auch Rapidminer und anderen Tools ohne weitere Schnittstellen weiterbearbeitet werden. Zur Nutzung der Streaming API werden die bekannten Twitter4j Wrapper-Bibliotheken aus Kapitel 3.8 verwendet.

Die Erstellung einer Zieldatei für die Abspeicherung eines Streams erfolgt folgendermaßen:

```
public static void generateCsvFile(String sFileName)
   {
 try
{
   writer = new FileWriter(sFileName);
    writer.append("User");
    writer.append(";");
    writer.append("UserID");
    writer.append(";");
    writer.append("Date");
    writer.append(";");
    writer.append("Location");
    writer.append(";");
    writer.append("# Followers");
    writer.append(";");
    writer.append("# Friends");
    writer.append(";");
    writer.append("Retweet Count");
    writer.append(";");
    writer.append("lang");
    writer.append(";");
    writer.append("Tweettext");

    writer.append(System.getProperty("line.separator"));
//needed to force new line

   }catch(IOException e)
{
     e.printStackTrace();
}
   }
```

Hier werden die Spalten und Spaltentitel festgelegt. Die oben im *Filewriter* erstellten Felder sind die wichtigsten Datenfelder eines Tweets. Sie beinhalten unter Anderem das

Tweetdatum und die Tweetnachricht. Die Spalte *lang* erfordert im weiteren Verlauf der Arbeit noch eine Überprüfung mittels einer Sprachdetektion in Java, da es nach unseren Erkenntnissen und einigen Forenberichten zufolge Bugs in der Sprachfilterung der Twitter API gibt. Zur Sprachdetektion der Tweets wird *jlangdetect* verwendet. Diese Bibliothek hat sich in der Vergangenheit mit einer hohen Genauigkeit bewährt[1]. In der jetzigen Form erlaubt die Twitter API zwar eine Sprachfilterung (wenn auch fehlerbehaftet), doch auch dies reicht nicht aus um alle fremdsprachigen Tweets zu filtern. Während der Korpuserstellung ist aufgefallen, dass manche Twitteruser mit einem „en“ in ihrem JSON-Sprachfeld dennoch in einer fremden und oft nicht dekodierbaren Sprache geschrieben haben. Die Sprachfilterung über das *Lang* Feld durch Twitter ist also unzuverlässig. Es muss eine Detektion über das *status* Feld (also die Tweetnachricht) erfolgen um sicher zu gehen.

Als Trennzeichen der Spalten wird hier „;“ verwendet. Der Grund hierfür ist, dass bei dem Standardtrennzeichen „ ,“ die Datei korrumpiert wurde, weil viele Tweets Kommata enthalten. Ohne eine zusätzliche Java Bibliothek führte dies dann zu einer fehlerhaften Spaltentrennung in der Datei. Dadurch konnte die Datei weder exportiert, noch weiter bearbeitet werden. Es traten unerklärbare Fehler bei der Stimmungsanalyse auf, die auf dieses Problem zurückgeführt werden konnten. Nach einigen Probeläufen konnte festgestellt werden, dass das Zeichen „;“ fast nie von den Twitter Nutzern verwendet wurde und sich daher besser als Trennzeichen eignet[2].

Nachdem nun die einzelnen Datenfelder definiert wurden, kann die CSV nun mit zu extrahierenden Daten gefüllt werden. Dazu werden im Voraus diverse Twitter4j Klassen importiert. Zu den wichtigen Klassen gehören folgende:

- *twitter4j.FilterQuery*
- *twitter4j.Status*
- *twitter4j.StatusListener*
- *twitter4j.TwitterStreamFactory*
- *twitter4j.conf.ConfigurationBuilder*

Die Filterquery dient der gezielten Suche nach bestimmten Topics und Inhalten im Stream. Der Status ist im Grunde die Twitternachricht eines Nutzers. Sie wird im Twitterjargon und auch in der Twitter4j Bibliothek als Status bezeichnet. In der Klasse *twitter4j.Status* befinden sich auch die Methoden zum Zugriff auf die JSON-Felder eines Tweets, welche vom Server übergeben werden. Der *twitter4j.conf.Configurationbuilder* ist die Builderklasse zur Authentifizierung und Verbindung zum Twitteraccount und Twitterstream über Java[3]:

[1] (https://github.com/melix/jlangdetect)

[2] Diese Problematik kann in einer zukünftigen Arbeit eventuell dadurch vermieden werden, dass die Daten in MongoDB als komplette JSON Daten abgelegt und abgerufen werden oder eine CSV Bibliothek verwendet wird statt den Standardwerkzeugen von Java.

[3] Die Strings, die für den Consumerkey, ConsumerSecret und AccessToken, AccessTokenSecret verwendet werden, wurden hier mit Platzhaltern ersetzt, da es sich um sensible Zugangsdaten handelt.

```
ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true);
        cb.setOAuthConsumerKey("84nhjlokdfgv08426tnbh");
        cb.setOAuthConsumerSecret("09842nlogsdfgu902mfsß2mts");
        cb.setOAuthAccessToken("98235263467-m.mxöcpbjpqü03öl");
        cb.setOAuthAccessTokenSecret("öoljadgpoadsegüpowegtä+üp");
```

Für den Zugang zum Twitterstream ist der Prozess dieser Authentifizierung zwingend erforderlich und dient der Zugangssicherheit des Accounts. Die Tokens und Keys müssen auf der Twitter-Developer Webseite erstellt und angemeldet werden. Dieser Prozess ist nach einer Anmeldung auch programmatisch zur Laufzeit möglich.

Bevor nun die JSON Datenfelder der Tweets in die CSV-Datei kopiert werden können, muss der fire hose des Twitter Streams mit den Schlüsselwörtern limitiert werden. In diesem Fall handelt es sich bei den Schlüsselwörtern um eine Reihe von Kinofilmtiteln. Eine FilterQuery kann folgendermaßen erstellt werden:

```
       FilterQuery fq = new FilterQuery();
        String keywords[] =
{movie1, movie1b, movie2, movie3, movie4, movie4b};
        //String language[] = {"English"};

        fq.track(keywords);
        //fq.follow(users);    //Zur Eingrenzung auf User
        twitterStream.addListener(listener);
        twitterStream.filter(fq);
```

Die Filterquery ist das Kernstück des Extraktors. Mit ihr kann der Stream allgemein an die Bedürfnisse des Experiments angepasst werden. Die textuellen Inhalte von Tweets können hier aber noch nicht abgegriffen werden. Die Query kann mit diversen Konstruktoren überladen werden. In der umfangreichsten Form sieht der Konstruktor generell folgendermaßen aus:

```
public FilterQuery(int count,
                   long[] follow,
                   java.lang.String[] track,
                   double[][] locations,
                   java.lang.String[] language)
```

Wichtige Angaben sind hier *follow*, *track* und *language*. Mit *FilterQuery.follow* kann der Stream eines bestimmten Users extrahiert werden. *Track* erwartet als Argumente ein String-Array mit Schlüsselworten (hier: die Kinofilmtitel). *Locations* und *Language* kann den Stream auf eine bestimmte Sprache oder bestimmte Geo-Koordinaten eingrenzen, so dass nur Tweets einer bestimmten Sprache und eines bestimmten Ortes sichtbar sind.

Genutzt werden in dieser Arbeit hauptsächlich die *track* und *language* Argumente. Track erhält im obigen Beispiel mit *fq.track(keywords)* die Titelliste. Die Bedeutung von *movie4* und *movie4b* beziehungsweise *movie1* und *movie1b* wird im Unterkapitel *Problematiken der Twitter Extraktion* 4.2 erklärt.

Ein Keyword - für die Variable *movie1* beispielsweise - ist ein String von der Form:

```
private static  String movie1 = "the expendables 3";
```

Dadurch wird der *Filterquery.track* Methode mitgeteilt, dass exakt diese Stringfolge in dem Tweet auftauchen muss. Da es sich zu dem Zeitpunkt um einen Filmtitel handelt, der noch nicht erschienen ist, kann davon ausgegangen werden, dass der Tweet von einem Trailer, einer Preview oder einer Usermeinung oder Antizipation handelt. Mittels *twitterStream.filter(fq)* wird dann die erstellte Filterquery dem Twitterstream übergeben. Ab dem Zeitpunkt werden dann nur noch zutreffende Tweets angezeigt und gestreamt.

Es ist zu beachten, dass wegen der beschränkten technischen Möglichkeiten nicht mehrere Computer multiple Streams starten konnten. Dies wäre nötig gewesen um pro Filmtitel eine einzige Titelfilterung durchzuführen, so dass der Twitterstream dann nur Tweets beeinhaltet, welche zu einem einzigen Film gehören. Twitter erlaubt aber keine multiplen Streaminstanzen zu einem Account und einer IP-Adresse. Das heißt, es wären mehrere Rechner mit mehreren Twitter-Accounts und unterschiedlichen IP-Adressen nötig gewesen. Das ist aber in dieser Diplomarbeit technisch und finanziell nicht möglich gewesen. Für eine programmatische Unterscheidung oder aber Abspeicherung in verschiedene Dateien, muss im späteren Verlauf also eine erneute Überprüfung jedes einzelnen Tweets stattfinden. Dadurch genügt es eine einzige Streaminstanz zu initialisieren und die Tweets danach zur Laufzeit zu sortieren.

Nachdem nun die Vorbereitungen für die Extraktion der Datenbasis getroffen sind, können mittels der importierten Javaklassen auf die benötigten Methoden und JSON-Felder der Tweets zugegriffen werden. Die Status-Klasse bietet Schnittstellen, die eine Extraktion einzelner Datenfelder erlauben. Dies kann folgendermaßen erfolgen:

```
String tweet_trim =
content.replace("\n", "").replace("\r","");
String str_id = String.valueOf(status.getId());
String str_follower = String.valueOf(status.getUser().getFollowersCount());
String str_friends = String.valueOf(status.getUser().getFriendsCount());
String str_retweets = String.valueOf(status.getRetweetCount());

if (content.toLowerCase().contains(movie2)) {

writer.append(username);
writer.append(";;");
writer.append(str_id);
writer.append(";;");
```

Tabelle 4.1: Film-Datensätze

Datum	Filmtitel	Records
02.07.14	Deliver Us From Evil	99796
11.07.14	Dawn of the Planet of the Apes	175982
11.07.14	A Long Way Down	1877
11.07.14	Boyhood	40104
18.07.14	The Purge: Anarchy	78219
18.07.14	I Origins	2882
25.07.14	Hercules	253762
25.07.14	Lucy	479929
25.07.14	A Most Wanted Man	8560
11.07.14	Calvary	1500

```
writer.append((status.getCreatedAt()).toString());
writer.append(";;");
writer.append(status.getUser().getLocation()); //or user.getlocation()?
writer.append(";;");
writer.append(str_follower);
writer.append(";;");
writer.append(str_friends);
writer.append(";;");
writer.append(str_retweets);
writer.append(";;");
writer.append(status.getUser().getLang());
writer.append(";;");
writer.append(counter +" "+  tweet_trim +";;");
writer.append(System.getProperty("line.separator")); //adds new line
```

Ursprünglich wurden in diesem Codeabschnitt wesentlich weniger Zeilen Code benötigt. Allerdings hat es zu Problemen geführt, falls der Filewriter auf die Statusmethoden zugreifen sollte. Näheres dazu in Kapitel 4.2.

4.1.1 Datensätze in dieser Arbeit

Datensätze zu folgenden Kinofilmen wurden entsprechend der oben beschriebenen Methoden mittels Twitter Streaming API extrahiert und abgespeichert: 4.1.

Das Datum gibt jeweils das Veröffentlichungsdatum in den USA an[4]. Bei den meisten Filmen handelt es sich um größere Produktionen, die *Wide* ausgestrahlt werden. Das bedeutet, dass der Releasetermin global an einem fixen Tag stattfindet.

[4]In den USA sind die Releasetermine aller Kinofilme generell Freitag bis Sonntag. Das Extrahieren der Tweets musste also immer vor Freitag bereits stattgefunden haben.

4.2 Problematiken der Twitter Extraktion

Eines der Probleme bei der Datenextraktion von Twitter tritt auf, falls selbst mit der Filterquery immer noch zu viele Tweets in den Stream gelangen und der fire hose stream überlastet wird. Dadurch wird automatisch die Tweetrate von dem Twitterserver aus gedrosselt. Falls die Applikation weiterhin zu viele Tweets aus dem fire hose abgreift, wird die Verbindung gekappt. Dies kam in dieser Diplomarbeit nur sehr selten vor, da die Filterquery entsprechend gestaltet war. Trotzdem sollten die Statuslistener *onTrackLimitationNotice* und *onStallWarning* implementiert werden, da letzterer z.B. vor der Überlastung eine Warnung übergibt. Dadurch kann auch während der Laufzeit der Stream fallbedingt reguliert werden.

Ein anderes Problem der Extraktion ist die Sprache. In der Twitter API sind zwar Methoden zur Unterscheidung der Sprache vorgesehen, diese scheinen aber teilweise willkürlich zu greifen. Beispiel:

```
        FilterQuery fq = new FilterQuery();
        String keywords[] =
{movie1, movie1b, movie2, movie3, movie4, movie4b};
        String language_filter[] = {"English"};
        fq.language(language_filter);
```

Eigentlich dürften jetzt nur noch englische Tweets in den Statusstream gelangen, nämlich solche, welche die keywords enthalten (alle zutreffenden) und gleichzeitig in englischer Sprache verfasst sein müssen. In der Praxis hat es aber Fälle gegeben, wo der Filterquery Sprachfilter nicht funktionierte, da in der extrahierten Datenbasis auch Tweets mit dem Eintrag *ES* für Spanisch oder *TR* für Türkisch in dem Datenfeld *language* erschienen sind. Theoretisch ist zwar auch eine nachträgliche Filterung in der Datendatei möglich, führte allerdings zu unnötigen Aufblähungen der Dateigröße und Extraaufwand, denn eine extrahierte CSV mit über 500 MB ist nicht unüblich. Daher wurde das Sprachdetektionspaket in Java verwendet, jlangdetect.

Ein weiteres Problem in der Extraktion von einer Datenbasis von den sozialen Medien sind die Sonderzeichen und Emoticons. Da es von Twitter aus keine Begrenzung in Form und Inhalt von Tweets gibt, können theoretisch alle möglich Sonderzeichen und auch sogenannte Escape-Sequenzen[5] auftauchen.

Ein anderes Problem ist die fehlende Möglichkeit multiple Streaminstanzen über den selben Twitteraccount laufen zu lassen. Aber auch das Erstellen zusätzlicher Accounts ist keine Lösung, da von Twitter aus zusätzlich eine IP-Adressüberprüfung stattfindet. Dadurch kann der Twitter-Server feststellen, ob die selbe Person mehrere Streams startet. In beiden Fällen wird die Verbindung zum Stream unterbrochen.

[5] Es handelt sich hierbei um Sonderzeichen, die von Twitter oder dem Betriebssystem erkannt werden und eine bestimmte Sonderfunktion ausführen. Eine oft in Twitter vom User unbewusst verwendete Escape-Sequenz ist beispielsweise \n für eine Neuzeile.

5 Preprocessing der rohen Datenbasis

Das Preprocessing von den Rohdaten ist eine der fundamentalsten Aufgaben beim Opinion Mining. Wie bereits im Kapitel **Verwandte Arbeiten** 2 aufgezeigt, findet das Thema des Preprocessing auch in diversen wissenschaftlichen Arbeiten Beachtung, unter anderem in [10] und in [24]. In der deutschen Übersetzung verwenden wir die Bezeichnung *Vorverarbeitung*. Da die englische Bezeichnung aber gängiger ist, wird hier im Folgenden die englische Bezeichnung synonym verwendet werden.

Das Preprocessing im Allgemeinen bedeutet, dass eine Datenmenge vor der eigentlichen Datenanalyse aufbereitet beziehungsweise vorverarbeitet wird. Es ist also eine nötige Vorstufe zum Data Mining im Allgemeinen und Opinion Mining in diesem Fall. Durch das Preprocessing wird nicht nur dafür gesorgt, dass die Datenbasis zur weiteren Bearbeitung verwendet werden kann, es ist auch ein wichtiger Faktor der Lastreduzierung. Auf Grund der Bereinigung, Ersetzung und Löschung irrelevanter Zeichen oder ganzer Datenmengen, kann eine höhere Effizienz der Laufzeit des Algorithmus und eine Reduzierung des Rechenaufwands erzielt werden. Beispielsweise kann in einer größeren Datenbasis die Anzahl der Attribute sehr hoch werden. Man spricht dann von einer hohen Dimensionalität des Vektorraums, einem generellen Problem des Data Mining bei großen Datenmengen. Die Laufzeit von Klassifikationen wird dadurch ebenfalls erhöht, da eine sehr viel größere Menge an Attributen betrachtet werden muss.

Das Preprocessing ist im Data Mining ein sehr zeitaufwendiger Prozess. Es ist keine Seltenheit, dass der Anteil des Preprocessing im Data Mining über die Hälfte, teilweise gar über 80% der Gesamtprojektzeit beansprucht. Auch in dieser Diplomarbeit konnte diese Statistik bestätigt werden[1]. Das Preprocessing hat einen großen Teil der Arbeitszeit beansprucht.

5.1 Preprocessing von Twitter Daten

Preprocessing von Twitterdaten unterliegt einigen Eigenheiten, die speziell bei Tweets auftauchen. Einige der Verarbeitungsmethoden werden in dieser Arbeit an die Methoden von [10] angelehnt, da dieser anhand von Tweetdaten die Relevanz von Preprocessing aufzeigt. Einige Schritte der Verarbeitung werden in dieser Diplomarbeit mit selbst geschriebenen Methoden in Java implementiert. Es gibt aber auch Schritte, die unter Verwendung von Hilfstools und APIs durchgeführt werden, um sich von bereits bestehenden Möglichkeiten zu bedienen und nicht den Rahmen zu sprengen. Vor allem RapidMiner

[1] `http://www.kdnuggets.com/polls/2003/data_preparation.htm`

bietet eine Reihe von bereits implementierten Algorithmen an, welche als Operatoren das Preprocessing und die Dimensionalitätsreduktion unterstützen.

Das Preprocessing wird in zwei Phasen durchgeführt. Die erste Phase ist textuell. Die zweite Phase ist analytisch. Mit der textuellen Phase wird vor allem der Tweettext bereinigt und von irrelevantem Inhalt befreit. Die analytische Phase umfasst unter anderem das Stemming und die Normalisierung von Buchstabenwiederholungen, die die Erkennung von Sentiment Worten beeinflussen können. Die Einzelschritte der ersten Phase sehen wie folgt aus:

1. Beseitigung aller URLs und Hyperlinks. Replatzierung durch Placeholder „URL“
2. Beseitigung sämtlicher Zahlen im Tweettext.
3. Entfernung der „RT“ tags.
4. Entfernung aller Hashtags (mit Beibehaltung der Hashtagworte): #
5. Bereinigung von auftauchenden Usernamen im Tweettext. Ersetzung durch „USER“
6. Verarbeitung und Umwandlung des datetime Feldes im Twitter JSON Format.

Schritt 1 bietet Platz zur Diskussion. Es könnte zum Beispiel vorkommen, dass Namen und Begriffe in der URL auftauchen, die das Ergebnis beinflussen. In unserem Fall könnte das zwar nur selten passieren, da viele User in Twitter Verkürzungsdienste für URL-Links verwenden[2], trotzdem muss dies beachtet werden. In der Arbeit von Yanwei Bao et al. [10] wird auch untersucht, ob die Löschung von URLs einen positiven oder negativen Einfluss auf die Ergebnisgenauigkeit hat. Insgesamt kommen die Autoren zum Ergebnis, dass das Entfernen von URLs und Hyperlinks zu einer Verbesserung der accuracy führt. Daher wird auch in dieser Arbeit basierend auf diesem Ergebnis auf URLs in den Records verzichtet und mit Platzhaltern ausgetauscht.

Schritt 2 ist ebenfalls kritisch zu betrachten. Ein „RT“ kennzeichnet den Tweet als *Retweet*, d.h. bei diesem Tweet handelt es sich eine Weiterleitung und Wiedergabe von einem anderen Nutzer. Diese Information könnte von Bedeutung sein, da dies bedeutet, dass vielleicht viele Benutzer die Filmbewertung im Tweet unterstützen und die Ansicht des Versenders des Originaltweets teilen. Tatsächlich sind während der Extraktion der Datenbasis sehr viele Retweets aufgefallen. Vor allem vor der Veröffentlichung des Films traten solche Retweets auf.

Schritt 5 ist ein wichtiger Schritt, da in Twitter sehr viele unterschiedliche Usernamen auftauchen. Diese können auch in den Tweettexten selbst erscheinen, um einen Tweet an einen oder mehrere Benutzer zu schicken. Da das JSON Datenformat hier keine Unterscheidung zwischen Username und eigentlichem Tweettext macht, muss dies beachtet werden. Benutzernamen können frei gewählt werden. Das bedeutet, es könnten auch Worte mit emotionalem Inhalt - also sentiment - gewählt werden. Damit dies zu keiner Verfälschung der Analyse führt, werden diese ebenfalls mit Platzhaltern gefüllt.

[2] wie z.B. http://tinyurl.com/

Es gibt noch andere Preprocessing Methoden, die angewandt werden. In dieser Arbeit wurden die vorherhigen Vorverarbeitungsschritte bisher in Java implementiert. Dies ist auch für die weiteren Schritte möglich. Allerdings bietet RapidMiner eine text mining extension an, welche die Modellierung der nächsten Preprocessing Schritte ermöglicht[3]. Daher wird nun darauf zurückgegriffen.

In RapidMiner wird das Tokenizing, Stemming und Filtering angewandt. Wichtig ist auch das n-gram beziehungsweise 2-gram Tokenizing, das hier noch erläutert wird. Tokenizing ist das Verfahren einen Textstring in seine einzelnen Textbausteine aufzuteilen. Üblich sind hier wortweise Trennungen oder aber auch Symbole und andere. Das ist keine triviale Aufgabe für einen tokenizer, da in einem Text verschiedene Zeichen, Symbole und Punkte auftauchen können. Der tokenizer muss entscheiden, wonach getrennt werden soll. Das Ziel ist es, den Text in eine weiter verarbeitbare Form zu bringen und zu bereinigen. Außerdem kann analysiert werden, ob und wie oft bestimmte Sentiment Worte enthalten sind. Es muss beachtet werden, dass Abkürzungen eventuell vorher aufgelöst werden müssen, da sie ansonsten nicht erkannt werden könnten.

Der nächste Schritt ist das sogenannte Stemming. Mittels stemming werden Worte auf ihren Wortstamm reduziert. Es werden also konjugierte und deklinierte Worte wieder in die Ursprungsform zurückgeführt und der Wortstamm gebildet. In der deutschen Sprache wird synonym auch das Wort *Stammformreduktion* verwendet. Ziel des stemmings in dieser Arbeit ist es die Erkennbarkeit von emotionalen Worten zu erhöhen. Dies ist auch für das lexikale Analyseverfahren notwendig, da SentiStrength keinen Anspruch auf Vollständigkeit aller englischen Worte erhebt.

Als dritter Schritt müssen die *Stopwords* entfernt werden. In diesem Schritt werden all die Worte eliminiert, welche kein sentiment Gewicht haben oder den Satz zu einer negativen oder positiven Bewertung hinführen. Die Stopwords sind sehr zahlreich und befinden sich in einer eigenen Liste. Durch die Entfernung derer werden unnötige Merkmale im Merkmalsvektor der SVM Experimente vorab bereinigt. Das sind beispielsweise Artikel wie „the“ oder Pronomen und ähnliche.

Der vorletzte Schritt ist eines der wichtigsten in dieser Phase des Preprocessing, da ein Parameter dieses Schritts das Gesamtergebnis der sentiment analyse beeinflussen kann. Es handelt sich um das n-gram Verfahren.

Einen gesonderten Platz nimmt hier der letzte Schritt ein. Twitter speichert die Uhrzeiten und das Datum eines Tweets in einer besonderen Form. Es gibt dazu eine eigenes Feld im JSON Datenformat, welches mittels „*created_at*“ angesprochen werden kann. Das Format beeinhaltet auch die Zeitzoneninformation, was in dieser Arbeit zu Problemen beim Konvertieren, Parsen und abspeichern geführt hat. Das created_at Feld wird vom Twitterstream abhängig vom aktuellen Systemstandort einer bestimmten Zeitzone zugeordnet. Die Zeiten sind also keine Lokalzeiten, sondern umgerechnete Zeiten der aktuellen Systemzeituhr des Systems, worauf die Streamimplementierung läuft. Wegen dieser Besonderheit wird dieser Problematik und der Lösung dieser Problematik in 5.1.1 ein eigenes Unterkapitel gewidmet.

[3] http://marketplace.rapid-i.com/UpdateServer/

5.1.1 Das Verarbeitungsproblem des Datumformats bei Twitterdaten

Twitter verwendet in der Speicherung des Datums im JSON Feld eines Tweetrecords ein eigenes US Format. Dieses Datumsformat kann nicht direkt zur weiteren Verarbeitung verwendet werden, da auch Wochentage und die Zeitzoneninformation enthalten sind. Beispielsweise ist es nicht ohne Weiteres möglich, Statistiken über den zeitlichen Verlauf von Tweets zu erstellen, falls das Datumsformat nicht umgewandelt wird. Ein Datumseintrag vom Twitterstream sieht folgendermaßen aus:

```
Mon Jul 28 22:37:49 CEST 2014
```

Es wird zuerst der Wochentag abgekürzt auf drei Buchstaben vorangeführt. Darauf folgt der Monatsname in Buchstaben, der ebenfalls auf die ersten drei Buchstaben abgekürzt wird. An dritter Stelle steht der Monatstag als Ziffer. Danach folgen Uhrzeit mit Minuten und Sekunden, die Zeitzoneninformation CEST[4] und das Jahr.

Das Problem dieses Formats ist es, dass der Wochentag und der Monatsname in ein Ziffernsystem geparsed und konvertiert werden müssen. Das Datum dieses Tweets müsste also eigentlich in der Form TT.MM.JJJJ sein, d.h. 28.07.2014. Erst dadurch kann eine vernünftige Zeitreihe entstehen, die dann auch geplottet werden kann. Die Uhrzeit könnte separat in ein eigenes Feld geschrieben werden. Es wird daher in dieser Arbeit als Teilaufgabe des Preprocessing das Datum geparsed und in zwei neue Datenfelder umgeschrieben. Zu diesem Zweck wird die abstrakte Javaklasse *DateFormat* in der Unterklasse *SimpleDateFormat* implementiert. Mit dieser Klasse können in Java Formatierungen von Daten in verschiedenen Datumsformaten vorgenommen werden. Dafür muss dem Konstruktor als Argument das Datumsformat in abstrakter Form übergeben werden. Das Kernstück der Lösung des Datumsproblems ist also das parsen der Twitterzeit mittels Konstruktor des SimpleDateFormat:

```
final DateFormat df =
new SimpleDateFormat("EEE MMM dd HH:mm:ss ZZZZ yyyy", Locale.US);
         final Calendar c = Calendar.getInstance();
         String datetest_diplom;
         df.setLenient(true);
```

Es wird zunächst eine neue Instanz (*df*) von der Klasse erzeugt, welche im Konstruktor das zu parsende Datumsformat beschreibt. Das Argument „*EEE MMM dd HH:mm:ss ZZZZ yyyy*“ ist das sogenannte *pattern* und zeigt der Klasse mit Platzhaltern an, wo die einzelnen Informationen in dem Rohdatum stehen. Jeder Buchstabe hat hier eine Bedeutung. Genauere Informationen können von der Java API Dokumentation[5] entnommen werden. Insbesondere die Zeitzone muss beachtet werden und wird hier mit „*ZZZZ*“ angedeutet. Ein optionales Argument ist die Lokalinformation *Locale*. Sie gibt

[4] Central European Summer Time
[5] https://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html

die Landessprache des zu formatierenden Strings an. Dort darf also nicht das deutsche Sprachformat stehen, sondern *Locale.US*, obwohl Twitter das Datum mit CEST als europäisch angibt.

Im nächsten Schritt des Datumspreprocessing wird die Inputdatei eingelesen. Hierbei handelt es sich um die Tweetdatensätze aus dem Kapitel 4. Wenn die Daten eingelesen wurden, wird das Datumsfeld umgeschrieben. Zeilenweise wird das Datum eingelesen. Die Datumsinformationen müssen einzeln auseinandergetrennt und neu geschrieben werden. Durch die Datumsklasse und das Pattern können die Stunden und Minuten einzeln entnommen werden. Diese müssen aber in ein als Datum lesbares Format geschrieben werden. Dies geschieht folgendermaßen:

```
  writer.append(c.get(Calendar.DAY_OF_MONTH) + "."
 +( c.get(Calendar.MONTH)+1) + "." +c.get(Calendar.YEAR));
writer.append(";");
writer.append("um " + ((c.get(Calendar.HOUR_OF_DAY)) + ":"
   + c.get(Calendar.MINUTE) + ":" + c.get(Calendar.SECOND)));
writer.append(";");
writer.append(processed_tweet);
writer.append(System.getProperty("line.separator")); //adds new line
```

Es kann mittels *c.get* auf die Informationen im String zugegriffen werden. Allerdings müssen diese wie gesagt in der richtigen Reihenfolge wieder in den writer geschoben werden, da das Resultat sonst nicht als Datum und Uhrzeit begriffen werden kann. Das Datum wird hierzu mittels *DAY OF MONTH, MONTH, YEAR* und deutschen Trennpunkten zusammengesetzt. Durch die Punkttrennung statt der Strichtrennung „/“ weiß das System, dass es sich hier nicht um das US Format, sondern das deutsche Format handelt.

Die Uhrzeit wird in einem neuen Feld erzeugt und ist etwas komplizierter. Im Idealfall würde sie in ähnlicher Art und Weise mittels *HOUR OF DAY, MINUTE, SECOND* erzeugt werden. Allerdings ergibt sich dann das Problem des *leading zero*. Java stellt nämlich bei Stunden, Minuten und Sekunden keine Nullziffer voran, falls die Zahl kleiner als 10 ist. Bei dem Datum kann das ignoriert werden, da das Datum ohne vorangehende Null immer noch als Datum erkannt wird. Bei der Uhrzeit passiert das allerdings nicht. Wenn das neu erstellte Zeitfeld gelesen würde ohne neu zu parsen, würde das Zeitformat nicht erkannt werden.

Um das Problem besser in den Griff zu bekommen, wird nach dem Parsen im obigen Schritt *(c.get)* der gesamte Ausdruck des Datums und der Ausdruck der Uhrzeit in zwei verschiedene Variablen geschrieben. Danach werden die Variablen zu einem Datumobjekt konvertiert. Nun kann mit SimpleDateFormat ein neues Datum -und Uhrzeitformat definiert werden. Hierbei handelt es sich um das gängige deutsche System. Diese beiden neuen Formate können nun den beiden Datumobjekten zugewiesen werden, so dass das Datum und die Uhrzeit im SQL Server als auch in .CSV korrekt erkannt werden.

5.1.2 Retweets: Warum sie nicht entfernt werden

In vielen verwandten Arbeiten werden während der Vorverarbeitung auch Duplikate entfernt. Das sind hauptsächlich identische Kopien bereits vorhandener records. Auch in dieser Arbeit wurde die Durchführung dieses Schrittes für die Vorverarbeitung in Betracht gezogen und teilweise in Kapitel 7 angewendet. Abgesehen von der Notwendigkeit in diesem Sonderfall, wurde auf die Entfernung der Duplikate in dieser Arbeit verzichtet. Technisch ist das Auffinden und Entfernen der Duplikate ein simpler Prozess. Die Entfernung von ca. 12.000 Duplikaten im Hercules-Datensatz etwa benötigt ca. 5 Sekunden Bearbeitungszeit. Trotzdem wird prinzipiell von der Entfernung von Duplikaten abgesehen, da es technisch gesehen keine Duplikate geben kann. Twitter erlaubt nicht das Senden zweier exakt gleicher Tweets. Duplikate können also nur durch vorverarbeitete Retweets entstehen.

Ob Retweets als Duplikate zu betrachten sind, ist eine Design-Entscheidung. Es wäre zwar möglich, alle Experimente jeweils mit und ohne Retweets durchzuführen, wäre aber zu zeitaufwendig und verfälschend, denn Retweets entstehen von Usern, die den Tweet eines anderen Users übernehmen und senden. Auch dies kann nicht mehrfach gemacht werden. Im rohen Datensatz steht aber immer noch zusätzlich der Username des Senders. Daher handelt es sich nicht um eine exakte Kopie, bis das Preprocessing alle Benutzernamen mit dem Platzhaler „USER“ ersetzt. Aber auch dann würde das Entfernen zur Verfälschung führen. Dazu muss die Überlegung gemacht werden, wann ein Retweet stattfindet. Es kann davon ausgegangen werden, dass ein User im Allgemeinen immer dann den Tweet eines Anderen retweetet, wenn er die selbe Meinung mit der somit selben Polarität vertritt. Das heißt das Entfernen dieses Retweets und des zugehörigen Scores bzw. Polarität, würde die bewertende Ansicht dieses Users entfernen und ignorieren. Deswegen wird entschieden, Retweets nicht zu entfernen, solange es sich nicht um eine massive Score-Manipulation des Gesamtdatensatzes handelt[6].

5.2 Dimensionality Reduction

Dimensionsreduktion (DR) steht für Methoden und Algorithmen zur Reduktion der (Hoch-)Dimensionalität der Vektorraumrepräsentation von Daten bzw. Datensätzen. Es wird mit DR versucht „*diejenigen Richtungen in einem hochdimensionalem Raum zu bestimmen, in denen die wesentlichen Strukturen in den Daten deutlich werden*“[7]. Falls der Begriff der Dimensionsreduktion als allgemeiner Oberbegriff für die Verwendung einer Teilmenge aller Features angesehen wird, könnte die *Feature Selection* 5.3 als eine Auslegung von DR und *Feature Extraction* als andere Auslegung angesehen werden [53]. In der Literatur wird die Dimensionsreduktion aber auch bei manchen Autoren als Alternative zur Feature Selection beschrieben. Die Dimensionsreduktion ist in dieser Arbeit

[6] Beim Hercules Datensatz führte ein einziger Tweet des Hauptdarstellers zu insgesamt über 12.000 Retweets innerhalb von 22 Stunden und einer Verdopplung der Anzahl an positiven Tweets im Datensatz.

[7] Wissensentdeckung in Datenbanken, LV SS13, Prof. Morik & Dr. Ligges, S.438

unterschiedlich zur Feature Selection 5.3, welche aus Filtering und Wrapper besteht. Mit Dimensionsreduktion ist in diesem Unterkapitel also z.b. PCA gemeint.

Ziel der Dimensionsreduktion ist im Allgemeinen die Projektion eines p-dimensionalen Datensatzes in einen repräsentativen k-dimensionalen Datensatz:

$$(n \times p) \rightarrow (n \times k) \tag{5.1}$$

mit n Beobachtungen für p Variablen im $\mathbb{R}^p$ *und es gilt* $k < p$

Eine der bekanntesten Methoden zur Dimensionsreduktion ist die Hauptkomponentenanalyse (HKA), im Englischen Principal Component Analysis (PCA). PCA stammt aus der multivariaten Statistik und ist eine korrelationsbasierte Methode. PCA wählt eine Menge an repräsentativen Dimensionen (genannt *Principal Components* bzw. *Hauptkomponenten*) basierend auf dem Grad der Variation der Originaldaten. Die Hauptkomponenten sind folgendermaßen definiert[8]: Gegeben sei $X = (x_1, ..., x_p)$ als Datenmatrix mit n Beobachtungen von p Merkmalen. Jede Spalte

$$x_j = \begin{pmatrix} x_{1j} - \overline{x_j} \\ x_{2j} - \overline{x_j} \\ \vdots \\ x_{nj} - \overline{x_j} \end{pmatrix}$$

stellt die *zentrierten* Beobachtungswerte des Merkmals X_j mit $j = 1, ..., p$ dar. Zentriert werden die Daten im Nullpunkt, indem der arithmetische Mittelwert

$$\overline{x_j} = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \tag{5.2}$$

aller Beobachtungen des Merkmals X_j subtrahiert werden. Dieser Vorgang verschiebt den Schwerpunkt der Datenpunkte in den Nullpunkt des Raums. Das Bild 5.1 veranschaulicht schematisch diesen Vorgang in einem zweidimensionalen Raum ohne Berücksichtigung der anschließenden Dimensionsreduktion. Eine Punktwolke von Beobachtungswerten wird mit den Mittelwerten zentriert.

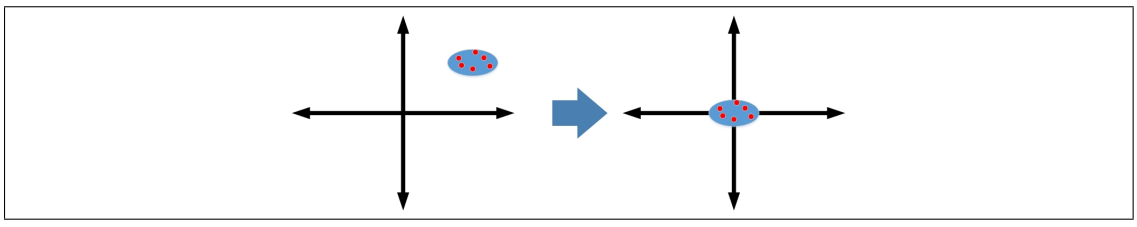


Abbildung 5.1: Schwerpunkt der Daten wird verschoben (ohne Berücksichtigung der Dimensionsreduktion)

Die k Hauptkomponenten werden nacheinander so gewählt, dass sie maximale Varianz von allen gewichteten Summen haben. Dabei muss die jeweils nächste Hauptkomponente k+1 die maximale Varianz von allen Merkmalen haben, die unkorreliert zur zuvor

[8] vgl. Wissensentdeckung in Datenbanken, LV SS13, Prof. Morik & Dr. Ligges, S.440ff

gewählten Hauptkomponente k sind. Die Hauptkomponenten sind also folgendermaßen definiert:

Definition 4. (Hauptkomponentenanalyse):

- *Die Hauptkomponenten sind gewichtete Summen der Originalmerkmale*
- *Die „Beobachtungen" der HKs sind definiert als die entsprechend gewichteten Summen der Beobachtungen der Originalmerkmale.*
- *Die erste HK hat maximale empirische Varianz von allen gewichteten Summen der „Länge" 1 der Originalmerkmale.*
- *Die (k + 1)-te HK hat die maximale empirische Varianz von allen gewichteten Summen der „Länge" 1 der Originalmerkmale, die unkorreliert sind mit jeder der ersten k HKs*

5.3 Feature Subset Selection (FSS)

Feature Subset Selection (kurz: Feature Selection) ist der Begriff für Verfahren und Methoden zur Auswahl von bestimmten Merkmalen aus einer größeren Obermenge von Merkmalen. Die Feature Selection bzw. Merkmalsauswahl ist vor Allem im Text Mining und Opinion Mining interessant, da in diesen Bereichen üblicherweise Zehntausende Wortmerkmale vorkommen. Durch das FSS ist es möglich, die ansonsten nicht behandelbaren größeren Textdatensätze zu verarbeiten und zu analysieren. Mit der Bezeichnung FSS sind nicht Preprocessing Verfahren gemeint, die ebenfalls vorab schon zur Reduktion von Merkmalen führen. So sorgt beispielsweise das Stemming auch für eine allgemeine Reduzierung der Gesamtzahl der Merkmale, indem alle Wortmerkmale mit dem selben Wortstamm zusammengeführt werden und Merkmalsredundanz vermieden wird. Stemming wurde in dieser Arbeit aber schon im textuellen Preprocessing behandelt. Die Merkmalsauswahl bei FSS erfolgt zeitlich nach dem Stemming.

Die Wahl der optimalen Featuremenge ist in der Literatur in mehreren Fällen als NP-hart gezeigt worden (u.a. in [14], Amaldi und Kann 1998 in [6], Blum und Rivest 1992 in [12], GH John et al. in [29]). Das Optimierungsproblem der FSS kann also nur approximativ gelöst werden, indem Heuristiken und Greedy-Algorithmen verwendet werden. Dazu gibt es viele bestehende Ansätze, von denen einige hier vorgestellt werden.

In der Feature Selection werden typischerweise *Gewichte* oder *Scores* für die Merkmale berechnet. Mittels dieser Gewichte w_i zu jedem Merkmal i wird dann ein heuristisches Ranking der Merkmale erstellt. Schließlich werden dann die Top k gewichteten Merkmale als Subset ausgesucht.

Die Feature Selection kann in drei verschiedene Typen unterteilt werden. Diese sind *Filter*, *Wrapper* und *Embedded* Methoden. Es werden für diese Arbeit einige Filter und Wrapper in Betracht gezogen und im Folgenden vorgestellt.

Filter unterscheiden sich von Wrapper Methoden dadurch, dass die Filter FSS Methoden ein Subset suchen und nach einer Bewertung die Subsetmenge dem Learnalgorithmus übergeben. Das heißt, die Erstellung einer reduzierten Menge kann relativ schnell erfolgen. Bei Wrapper Methoden wird der abschließende Lernalgorithmus in die FSS Suche mit eingebunden. Es wird ein Feature Set ausgewählt, das Subset wird evaluiert (bspsw. mittels *Cross Validation*) und dieses wird dann an den Lernalgorithmus als Input weitergegeben. Nach der Erstellung des Klassifikators (Hypothesenfunktion), wird die Accuracy für dieses Subset evaluiert und ein erneuter Durchlauf wird gestartet. Das heißt bei jedem potenziellen Subset wird der Lernalgorithmus angewendet. Dies geschieht so lange, bis eine akzeptable Feature Menge gefunden wird. Durch die Iterationen mit dem Lernalgorithmus sind Wrapper Methoden prinzipiell langsamer als Filter, können dafür aber wegen der iterativen Tests und der Interaktion mit dem abschließenden Lernalgorithmus eine passendere Untermenge für diesen finden. Außerdem werden auch Featurekombinationen mit berücksichtigt.

5.3.1 Feature Selection: Filtering

Es gibt eine Vielzahl von möglichen Filtermethoden für die Auswahl von Merkmalen. Ein Filter ist beispielsweise die *Significance Analysis for Microarrays* (SAM), die als *Weight by SAM* in RapidMiner existiert[9]. Die Formel hinter dem Operator lautet:

$$d(x) = \frac{\overline{x}_+ - \overline{x}_-}{s_0 + \sqrt{\frac{\frac{1}{n_x} + \frac{1}{n_-}}{n_+ + n_- - 2} \sum_{i+} (x_i - \overline{x}_+)^2 + \sum_{i-} (x_i - \overline{x}_-)^2}} \tag{5.3}$$

Die $i_{+/-}$ sind hier die Indizes der Merkmale, $\overline{x}_{+/-}$ das Mittel der Beispiele.

Ein anderer im Text Mining oft verwendeter Filter ist die *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF erzeugt einen Merkmalsvektor, der aus den gewichteten Vorkommen von Merkmalen besteht. Die Gewichtungen werden durch Vorkommenshäufigkeiten bestimmt. Das TF-IDF Maß besteht aus den beiden Komponenten TF und IDF. Die Termfrequenz TF gibt die Merkmalshäufigkeit innerhalb eines Dokumentes an. Sie ist der Quotient aus dem jeweiligen Merkmalsvorkommen und der Gesamtanzahl an Termen im Dokument. Der Quotient der TF gibt somit eine Gewichtung des Merkmals innerhalb eines einzelnen Dokumentes an. In diesem Fall entspricht ein Dokument einem einzelnen Tweet. Je höher der Quotient, desto öfter taucht das jeweilige Wort im Dokument auf. Die Annahme ist, dass Merkmale mit höherer Frequenz eine höhere Relevanz in diesem Dokument haben.

Die IDF Komponente dieses Filters betrachtet nicht ein einzelnes Dokument, sondern die Gesamtmenge aller Dokumente im Korpus. Die IDF ist also die Dokumentenfrequenz eines Merkmals gemessen an allen Dokumenten. Formal kann TF folgendermaßen definiert werden:

Definition 5. (Term Frequency):

[9] Schowe 2011, `http://www-ai.cs.uni-dortmund.de/PublicPublicationFiles/schowe_2011a.pdf`

Für ein Merkmalswort w_i im Dokument d gibt $TF(w_i, d)$ die Vorkommenshäufigkeit an.

Für die Dokumentenvorkommen des Wortes w_i für einen Dokumentenkorpus D gilt:

Definition 6. (Inverse Document Frequency):

$$IDF(D, w_i) = \log \frac{|D|}{DF(w_i)}$$

Die IDF Gewichtung eines Wortes, das in allen Dokumenten sehr oft auftaucht, erhält durch das *log* einen niedrigeren Wert. Falls beispielsweise das englische Wort „*and*“ in allen Tweets eines Datensatzes auftaucht, würde der Quotient 1 ergeben. Durch das *log* würde daraus eine 0 resultieren. Dieses Wort würde also als irrelevant eingestuft werden. Empirisch gesehen kann es sich bei Worten mit $IDF = 0$ häufig um Stoppwörter ohne Sentiment handeln. Falls ein Wort in wenigen Dokumenten auftaucht, erhält es umgekehrt ein höheres Gewicht.

Durch die Kombination beider Verfahren ergibt sich die TF-IDF, die als Produkt von $TF(w_i, d)$ und $IDF(D, w_i)$ berechnet werden kann:

Definition 7. (Term Frequency Inverse Document Frequency):

$$W_i = TF(w_i, d) * IDF(D, w_i)$$

W_i gibt in diesem Zusammenhang das Gewicht des Merkmals an.

Die TF-IDF ist in RapidMiner ein Parameter des Text Mining Operators *Process Documents from Data* und kann bei der Wortvektorerstellung ausgewählt werden.

Eine andere mögliche Funktion zur Gewichtung und anschließenden Filterung von Merkmalen ist die lineare Korrelation nach Pearson. Mit dem Pearsonschen Korrelationskoeffizient können lineare Abhängigkeiten zweier Merkmale gewichtet werden:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{5.4}$$

Es gibt diverse andere mögliche Gewichtungsfunktionen, die für die Feature Selection via Filter angewendet werden können. Einige dieser Filter wie z.B. auch I*nformation Gain* und χ^2 sind in RapidMiner vorhanden.

5.3.2 Feature Selection: Wrapper

Die Feature Selection via Wrapper it vor allem bekannt durch die Greedy-Algorithmen *Forward Selection* und *Backward Elimination*. Beide Methoden verwenden iterative Aufrufe des zugrunde liegenden Lernalgorithmus und der Evaluierungsmethode wie der Cross-Validation. Dadurch kann es zu längeren Laufzeiten als bei den Filtermethoden kommen.

Die Forward Selection wird mit einer leeren Merkmalsmenge initialisiert. Es werden dann iterativ nach und nach Merkmale in die Menge aufgenommen. Bei jeder Änderung der Menge wird die neue Menge hinsichtlich der Performanz evaluiert. Das Abbruchkriterium kann das Erreichen aller Features oder eines vorgegebenen Performanzwertes sein.

Die Backward Elimination durchläuft die Suche nach der bestmöglichen Merkmalskombination in umgekehrter Reihenfolge ab. Sie wird mit der Gesamtmenge aller Merkmale initialisiert. Iterativ werden dann aus der Gesamtmenge einzeln die Merkmale eliminiert. Bei jeder Eliminierung wird die Performanz der neu entstandenen Menge gemessen. Dies wird solange wiederholt, bis die Performanz anfängt stark zu fallen oder ein bestimmter vorgegebener unterer Performanzwert erreicht wurde.

Algorithm 1 Forward Selection

```
1: procedure FSS_FORWARD_SELECTION
2:     Initialize Feature Set S = ∅
3: loop:
4:     for given Set S, create S_i = S ∪ {i} with i ∉ S
5:     Evaluate and choose best S_i
6:     if termination_criterion = true then
7:         close;
```

Die Wrapper werden in dieser Arbeit bei der Merkmalsauswahl der Prädiktorvariablen im Regressionsmodell der multiplen linearen Regression verwendet. Dazu werden sequentiell beide Auswahlmethoden (Forward Selection und Backward Elimination) ausgeführt, um die für das Modell geeigneten Variablen zu wählen. Insbesondere bei Filmen mit sehr wenigen Tweets (I-Origins z.B.) führte die kombinierte Anwendung der Wrapper zu einer passenderen Auswahl der Prädiktorvariablen. Dies resultierte im anschließenden t-Test in einer Erhöhung der Signifikanz (Niveau $\alpha = 0.05$).

5.3.3 Ausgewählte Variante für SVM

Für die vorliegende Arbeit wurden mehrere Varianten der Dimensionreduktion und Feature Subset Selection getestet. Dies war besonders wichtig, da die Datensätze sehr groß waren und mehrere Zehntausend Features erzeugt wurden. Dadurch wurde die Berechenbarkeit mit den limitierten technischen Mitteln stark eingegrenzt. Ein Auswahlkriterium war daher, dass eine starke Dezimierung der Anzahl der Merkmalsvektoren stattfinden soll, ohne dass zu starke Einbußen der Accuracy zustande kommen.

Getestet wurde unter Anderem die PCA und beide hier vorgestellten FSS Wrapper. Außerdem wurde der TF-IDF Parameter in der Wortvektorgenerierung verwendet. Die genauen Ergebnisse können im Kapitel *SVM Experimente* 7.2 entnommen werden.

6 Prediction Modell und Extended Lexicon

Dieses Kapitel stellt die verwendeten Modelle und die Lexicon Expansion vor, um mit diesen Mitteln die zur Vorhersage benötigten Prädiktoren und Variablen zu generieren. Zur Expansion und Domänenanpassung des Lexicons wurde operativ eine zusätzliche Wortliste verwendet, die semi-automatisiert erstellt wurde. Dabei wurden unter anderem englische Online-Wörterbücher bezüglich Slangausdrücken manuell gesichtet und mit einem simplen Abgleich mit mehreren Datensatzsamples hinzugefügt. Zusätzlich zu dieser Methode wird in dieser Arbeit prototypisch eine vollautomatische Methode zur Lexicon Expansion vorgeschlagen. Diese verwendet WordNet und Synsets, um mittels eines minimalen initialen Seed-Lexicons Synonyme und Antonyme durch die WordNet Topologie zu finden. Zu diesem Zweck werden auch Similarity-Algorithmen verwendet, die eine Gewichtung und automatische Sentiment-Score Berechnung der neuen Einträge durchführen.

Zu jedem Filmdatensatz gibt es jeweils 3 verschiedene Statistiktabellen mit Ergebnissen, welche zeilenweise die täglichen Twitter-Statistiken, Sentimentscores, offizielle Verkaufszahlen u.a. Variablen gegenüberstellen. Bei diesen 3 Tabellen handelt es sich um Varianten der Stimmungsanalyse. Die 1. Variante ist die Lexiconanalyse mit der Originalwortliste von SentiStrength. Die 2. Variante verwendet SentiStrength mit der Lexiconexpansion ohne WordNet. Die dritte Variante dient zum Vergleich der Ergebnisse und stellt die Ergebnisse der Support Vector Machine Experimente dar.

6.1 Extended Lexicon

Eine der Innovationen dieser Arbeit in der Sentimentanalyse im Bereich der Vorhersage von Filmerfolg ist die Einführung eines zusätzlichen lexikalen Wortschatzes für den *Dictionary based* Ansatz. Im Rahmen der Diplomarbeit trat das Problem auf, dass es bei den dictionary Ansätzen praktisch noch keine Vorarbeit bezüglich der Detektion von Sentiments gab, die in Texten mit Film-Terminologien auftritt. Es gibt zwar Arbeiten über Sentimentanalyse von Daten über Filme, diese verwenden aber keine Lexicons oder überarbeiteten Lexicons. Es betrifft zum Einen allgemeine Worte und Satzkonstruktionen und zum Anderen auch Umgangssprache, die im Kontext von Kinofilmen eine andere Bedeutung beeinhaltet als kontextunabhängige Sprache. Beispielsweise ist das Adjektiv-Wort *unerwartet* ein von der Stimmung her negativ behaftetes Wort. Im Kontext der Filme ist dies aber ein sehr positives Wort, vielleicht sogar eines der erstrebenswertesten Adjektive für einen Film. Denn ein Film mit erwartetem Ablauf und Abschluss wird generell als langweilig und negativ bewertet werden. Vor allem in den sozialen Medien

wie Twitter können keine langen Bewertungen verfasst werden. Die User bewerten daher sehr oft in kurzen und informellen Texten.

6.1.1 Extended Lexicon Hintergrund

In dieser Arbeit wird ein zusätzliches erweitertes Lexikon mit Akronymen, Idiomen und Abkürzungen eingeführt. Diese Erweiterung wird in das Java Framework SentiStrength eingebaut. Die Wahl von SentiStrength bietet sich hier an, da SentiStrength bereits sehr viele englische umgangssprachliche Redewendungen und Bausteine enthält, diese aber noch mit Filmterminologie erweitert werden müssen.

SentiStrength verwendet in der Originalversion folgende *Lookup-Tables*:

- BoosterWordList: Diese Worte erhöhen oder verringern Scores von Sentiments
- EmoticonLookupTable: Smiley und andere Icons mit Stimmungsinhalt
- EmotionLookupTable: Kernstück der Lookups. Worte und Sentiment Scores.
- EnglishWordList: Alle Worte der englischen Sprache ohne Scores.
- IdiomLookupTable: Idiome, Redewedungen.
- NegatingWordList: Verneinungen zur Umkehrung des Score Vorzeichens
- QuestionWords: Fragewörter
- SlangLookupTable: Abkürzungen wie „lol“

Es gibt also eine Liste mit Idiomen in SentiStrength (*IdiomLookupTable*). Diese ist aber sehr beschränkt und für die Filmdomäne fast nutzlos. Sie enthält lediglich folgende umgangssprachliche Ausdrücke: *how are you, it hanging, wat up, what's good, what's up, whats good, whats up, wuts good.* Diese Idiome in SentiStrength werden teilweise mit Polaritäten wie 2 bewertet, da sie eine positive Stimmung suggerieren. Leider sind die Hälfte dieser Ausdrücke Fragesätze, welche in Tweets zu Filmen nicht vorkommen. Daher ist diese Liste für die Zwecke dieser Arbeit nicht verwendbar, was auch in der fehlerhaften Bewertung von antizipativen Worten sichtbar wird.

Falls eine erfolgreiche Erweiterung des Wortschatzes durchgeführt werden kann, könnte dies dazu genutzt werden, um mittels lexikalem Ansatz eine Trainingsmenge von Tweets zu labeln und diese dann als Trainingsinput für einen lernbasierten Support Vector Machine (SVM) Ansatz zu verwenden. Da es eines der großen Herausforderungen der lernbasierten Ansätze ist, eine ausreichend große Traininsgmenge zu labeln (oft manuell), würde die erfolgreiche Adaption eines solchen Verfahrens das Labelproblem beheben. Durch das Voranstellen der Lexikalanalyse und dem Labeling könnten kostengünstig große Traininsdatensätze generiert werden. Derzeit ist es nicht unüblich auf Services wie *Amazon Mechanical Turk* zuzugreifen, um mit Menschenhand viele Datensätze manuell zu labeln [7]. Der Service von Amazon ist allerdings mit Kosten verbunden. Der An-

satz der Kombination von lernbasierten und lexikonbasierten Algorithmen könnte diesen Aufwand ersetzen.

6.1.2 Fehlerhafte Klassifizierungen in der Domäne ohne extended lexicon

Grundlage für diese Extension des SentiStrength Lexikons sind - sofern vorhanden - Sprachquellen mit Auflistungen von Idiomen der englischen Sprache. Diese sind aber meistens kontextunabhängig und zum Teil bereits in SentiStrength übernommen worden. Daher werden diese mit Idiomen, Redewendungen und Abkürzungen erweitert, welche während der Verfassung dieser vorliegenden Diplomarbeit entdeckt wurden. Es handelt sich hierbei unter Anderem um englische Redewenungen oder unübliche Satzkonstruktionen, welche oftmals in Tweets auftauchten und mittels SentiStrength Algorithmus nicht einzuordnen waren. Bei diesen Satzkonstruktionen handelt es sich oft um Gebilde, die ein Mensch direkt einem Sentiment zuordnen könnte. Maschinell werden diese Gebilde aber kaum erkannt und dadurch direkt als *neutral sentiment* klassifiziert. Das führt zu einem Verlust der Prognosegenauigkeit, da dieser Text von menschlicher Sicht her eindeutig zuzuordnen ist. Ein bei Tweets zu Kinofilmen nicht selten auftretendes Beispiel ist hier das Idiom **can't wait**. Als Mensch interpretieren wir eingedenk des Kontextzusammenhangs das Idiom „*can't wait to watch expendables 3!!*“ als eine positive Stimmungslage bezüglich der Filmthematik *Expendables 3*. Eine automatisierte Analyse von SentiStrength teilt uns aber dies mit:

The text „can't wait to watch expendables 3!“ has positive strength 0 and negative strength 0.

Der Algorithmus kann den Text nicht gegen ein Sentiment klassifizieren obwohl das positive Sentiment aus unserer Sicht wahrscheinlich bei mindestens 3 liegt und das negative Sentiment eindeutig bei -1.[1] Auch alternative Suchen auf den bekannten Sentimentanalyse-Lexikons wie SentiWordNet[2] gaben keine Ergebnisse für diese Redewendung. Die Redewendung ist im Englischen aber sprachlich erfasst. Das Cambridge Dictionary teilt uns für das Idiom *can't wait* folgendes mit:

can't wait *idiom*
(also can hardly wait) to be very excited about something and eager to do or experience it: I can't wait to see you.

Es ist also ein bekanntes Idiom in der englischen Sprache, wird aber im Algorithmus der Sentimentanalyse und auch in den Sentimentanalyse Lexikons wie SentiWordnet ignoriert, da der Wortschatz die Redewendung nicht erfasst. Vor allem aber in den sozialen Medien, insbesondere Twitter, sind Redewendungen und Abkürzungen wegen der limitierten Zeichenzahl oft anzutreffen. Zusätzlich zu der Erweiterung des vorhandenen Wortschatzes gibt es ein weiteres Ziel bezüglich der Sentimentanalyse. In dem in dieser

[1] Es handelt sich hierbei um eine Dualskala von SentiStrength. Der Algorithmus liefert in dem Fall für jede Analyse immer 2 Werte mit [1-5] für positive Sentiments und [(-1)-(-5)] für negative. 1 steht dabei für nicht positiv, 5 für extrem positiv. Analog -1 für nicht negativ und -5 für extrem negativ.
[2] `http://sentiwordnet.isti.cnr.it/`, Sentiment Mapping von WordNet.

Arbeit verwendetem SentiStrength Algorithmus gibt es diverse Ausdrücke und Satzbausteine, die erfasst, erkannt und klassifiziert werden. Allerdings haben diese eine gänzlich andere Wertigkeit in Bezug des Sentiments in der Filmsparte als im allgemeinen Sinn. Dieser Fall soll anhand des in der Einführung erwähnten Wortes *unexpected* beispielhaft im Detail gezeigt werden:

- unexpected
- unexpectedly +2
- unexpectedness

Das Wort hat den Wortstamm *unexpected* und bedeutet im Deutschen *unerwartet* oder *urplötzlich*. Alleinstehend könnte das Wort eventuell wertfrei (d.h. ohne jegliches Sentiment) verstanden und betrachtet werden. Im generellen Kontext aber, werden alle Worte mit dem Wortstamm unexpected eher negativ behaftet angesehen. Der SentiStrength Algorithmus ignoriert das Wort weitestgehend in allen Formen bis auf die Variante *unexpectedly*. Der Algorithmus addiert bei jedem Vorkommen +2 zu dem positiven Score-Anteil. Der Grund dieser Unterscheidung konnte in der Diplomarbeit nicht erschlossen werden. Für einen exemplarischen Tweet mit *unexpected* gibt SentiStrength folgende Ausgabe:

```
Approximate classification:
the movie was terrible[-4] in the beginning
but totally unexpected in the end !
[-1 punctuation emphasis] [sentence: 1,-5]
[result: max + and - of any sentence]
[overall result = -1 as pos<-neg] (Detect Sentiment)
```

Das Adjektiv *terrible* wird mit -4 korrekt erkannt aber der gesamte restliche Satz bis auf das Ausrufezeichen[3] werden ignoriert. Selbst das Boosterwort *totally* führt hier zu keiner Polaritätsanpassung des Sentiments. Dies ist ein Indiz dafür, dass hier einige Ergänzungen zum Lexikon durchgeführt werden müssen. Insgesamt erhält der Satz den schlechtesten möglichen Score von $1 - 5 = -4$. Der Satz sollte tatsächlich aber deutlich positiver sein, da das Wort *unexpected* mit dem Boosterwort *totally* einen sehr hohen positiven Sentiment innehalten. Das bedeutet der Algorithmus hat hier einen Tweet fälschlicherweise als negativ klassifiziert.

Im Vergleich zum SentiStrength Wert soll hier auch die Ausgabe von SentiWordNet angegeben werden 6.1.

SentiWordNet hat nur einen Vorschlag für das Wort *unexpected* und dieser ist eher negativ: 6.1. P ist hier der positive Polaritätsscore, O steht für den neutralen beziehungsweise objektiven Polaritätsscore und N zeigt den Score für Negativität. P wird hier nur mit 0.125 und O mit 0.5 angegeben. Der Negativscore ist mit 0.375 dreimal höher als der positive Wert. Das Wort wird also insgesamt als neutral mit starker Tendenz zu negativ

[3] Ausrufezeichen erhöhen oder erniedrigen den letzten bekannten und erfassten Sentimentscore. In diesem Fall wird das negative Emotionswort terrible in der Negativität aufgewertet.

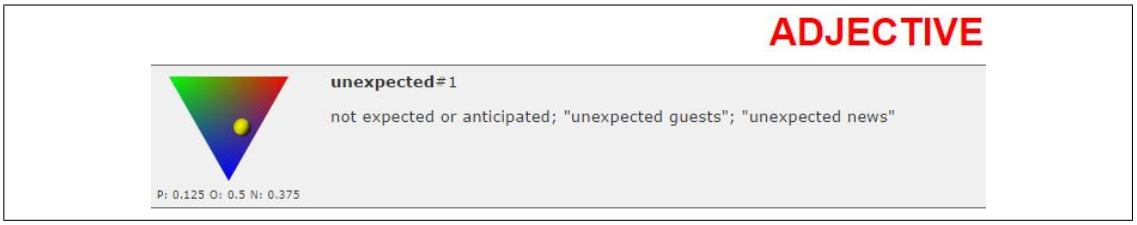


Abbildung 6.1: SentiWordNet: Sentiment für „unexpected“

bewertet, falls der Wortschatz von SentiWordNet bei der Sentimentanalyse herangezogen würde.

Es ist also gezeigt worden, dass mittels SentiStrength, SentiWordNet und WordNet[4] der Beispieltweet mit hoher Wahrscheinlichkeit mit einem negativen label versehen würde. Das wäre in der Domäne der Kinofilme aber völlig falsch und muss daher angepasst werden.

Der Fall mit dem Adjektiv stellt exemplarisch die Problematik bei der Sentimentanalyse ohne Kontext dar. Zielvorgabe ist es daher, solche Ausdrücke beziehungsweise Idiome in die Lexikonerweiterung aufzunehmen, bereits vorhandene Wortschätze zu erweitern und gegebenenfalls die Polaritätsscores zu ändern. Dadurch könnte eine lexiconbasierte Stimmungsanalyse über die Filmdomäne realisiert werden.

6.1.3 Manuelle domänenspezifische Erweiterung des sentiment lexicons

Ziel dieses Unterkapitels ist die Erweiterung des Lexicons, um genauere Klassifikationen zu erhalten. Zum Zeitpunkt dieser Arbeit wurden keine vergleichbaren Domänen-Erweiterungen von Sentiment Lexicons in einer anderen Arbeit gefunden. Auch diverse Recherchen in großen englischen Dictionaries wie Oxford oder Cambridge bezüglich spezieller Idiome in der englischen Sprache waren erfolglos[5]. Daher gibt es keine Richtlinie oder einen Vergleich. Es muss folglich eine eigene Methode gefunden werden, wie möglichst viele Redewendungen und Umgangssprache erfasst und in das Wortlexicon eingetragen werden können. Problematisch ist hier, dass in den sozialen Medien teilweise Redewendungen bzw. Idiome verwendet werden, die in keinem bekannten Dictionary aufgelistet sind. Es ist also eine Art neuer *social media slang* vorhanden. Hinzu kommt, dass Twitter im Vergleich zu den anderen sozialen Medien noch seltenere Idiome und Abkürzungen beeinhalten kann, da die User sich möglichst kurz mit 140 Zeichen ausdrücken müssen. Das bedeutet, dass selbst vorhandene allgemeine Listen von englischen Idiomen nicht unbedingt in den Twitterkontext zu Filmbewertungen passen würden. Die vorhandenen Twitter Datensätze müssen daher als Grundlage genommen werden.

Es gibt einige grundsätzlich unterschliedliche Möglichkeiten das Lexicon einer lexiconbasierten Sentiment Analysis zu erweitern. Im Falle dieser Arbeit handelt es sich um das Lexicon von SentiStrength. Ein Ansatz zur Lexiconerweiterung ist das Durchforsten

[4] SentiWordNet beeinhaltet den gesamten Wortschatz von WordNet und beschriftet diese mit Scores.
[5] http://www.oxforddictionaries.com/de/ und http://dictionary.cambridge.org/

aller vorhandenen Datensätze und manuelles Detektieren und Erfassen von englischen Ausdrücken. Das ist zwar theoretisch möglich, doch wegen der rund 1.500.000 Tweets in dieser Arbeit viel zu zeitaufwendig und nicht mehr im Rahmen dieser Arbeit. Das Problem hierbei ist nicht das manuelle Erweitern der Wortbilbibliothek, sondern das Durchsuchen von über einer Million Kurztexten.

Ein anderer Ansatz ist automatisch eine Lexiconexpansion durchzuführen. Dabei werden ausgehend von einem vorhandenen *Seed-Lexicon* Methoden angewendet, um neue unbekannte Worte zu finden und diese automatisch dem Lexicon hinzuzufügen. Es gibt in einigen Arbeiten Ansätze durch sogenannte *Seed Adjektive* die Expansion zu starten und Polaritäten dahingehend für neue Worte zu vergeben, dass sie einen Bezug zu den Seed Adjektiven haben. Das bedeutet es wird angenommen, dass beispielsweise bestimmte positive Adjektive immer im Zusammenhang mit anderen positiven Worten stehen müssen. Falls diese Worte entdeckt werden und bisher nicht in dem Seed Lexicon enthalten sind, werden diese aufgenommen und mit der selben oder ähnlichen Polarität des Seed Adjektives versehen. Dabei wird die Polarität des neuen Wortes in Abhängigkeit zu der Frequenz der Proximität des Wortes zum seed Adjektiv berechnet[6] [48]. Durch diesen automatischen Ablauf kann ein Lexicon vollautomatisch oder semi-automatisch unter Einbezieung vorhandener Lexicons wie WordNet aufgebaut werden. Ein ähnlicher automatischer Ansatz mit etwas anderen Ähnlichkeitsmaßen wird prototypisch in Unterkapitel *Automatische Lexicon-Expansion* 6.1.4 vorgeschlagen.

Diese Arbeit befasst sich mit der Idee, die Menge an Tweets mit potenziellen Idiomen und Worten, die der Filmdomäne angehören, im ersten Schritt zu minimieren. Falls die Teilmenge am Ende klein genug ist, wird eine manuelle Erfassung und Erweiterung des Lexicons in Betracht gezogen.

Es wird zunächst ein bekannter Datensatz zu einem Film genommen: *Hercules*. Der Datensatz Hercules beeinhaltet alle Tweets zum Film *Hercules* vom 20.07.2014 bis zum 28.07.2014. Erscheinungsdatum war der 25.07.2014. Insgesamt wurden in dieser Zeit 253.763 Daten mittels ununterbrochenem Twitter Stream extrahiert. Der rohe Datensatz wird gemäß der Schritte des Preprocessing in Kapitel 5 verarbeitet. Es wird ein neuer minimierter und vom noise[7] gereinigter Datensatz erstellt. Dieser Datensatz beeinhaltet nun alle englischen Tweets ohne möglicherweise verfälschende Benutzernamen, Zahlen und URLs. Duplikate werden bisher nicht entfernt[8].

Im nächsten Schritt werden die Sentiment Scores jedes einzelnen Tweets berechnet. Dazu werden einige Schritte wie im Kapitel Experimente 7 ausgeführt. Der SentiStrength Algorithmus wird im Anschluss der Datensatzminimierung auf den neuen Hercules Datensatz angewandt. Für SentiStrength wird als Parameter der Scoreberechnung der Wert *Scale* genommen. Dieser Schritt dient dazu eine *Scale Polarity* zu berechnen, mit der der Algorithmus drei verschiedene Polaritäten berechnet, wobei der erste Wert als Defaultwert für die Positivität die 1 ist. Der zweite Wert ist standardmäßig -1 und stellt den

[6] „frequency of the proximity of that adjective with respect to one or more seed words“. Taoboada 2011, S.270ff

[7] Unbekannte Zeichenketten und nicht interpretierbare Sprachen.

[8] Mit Duplikaten sind hier identische Retweets vom selben user gemeint.

Grad der Negativität dar. Beide Defaultwerte geben an, dass keinerlei positive oder negative Sentiments enthalten sind. Der dritte und letzte Wert der Score Berechnung ist das Gesamtergebnis des Satzes. Dieser letzte Score wird beim Scale Parameter als Summe der beiden vorangegegangenen Werte berechnet[9].

Wie in Kapitel **SentiStrength** 3.5.1 erläutert, ist diese Unterscheidung aus linguistischen Gründen notwendig, da in einem Text simultan negative und positive sentiments auftauchen können. Die Höhe der Zahl gibt die Stärke der jeweiligen Emotion an und ist im Betrag höchstens 4.

Für das Realbeispiel aus dem Datensatz Hercules sieht das Ergebnis bei Tweet Nummer 47 folgendermaßen aus:

```
Originaltweet:
Can't wait to see Hercules this Friday
Score:
1 -1 0
Can't wait to see Hercules this Friday [sentence: 1,-1]
[result: max + and - of any sentence]
[scale result = sum of pos and neg scores]
```

Der Tweet aus dem aktuellen Datensatz erhält die Standardbewertung 1 für positives Sentiment und -1 für negatives Sentiment. Damit teilt SentiStrength mit, dass kein Sentiment festgestellt wurde und daher die neutrale Polarität vergeben wird.

Exakt an dieser Stelle kann nun eine weitere Minimierung des Datensatzes erfolgen: Zur Auffindung von neuen Ausdrücken können nämlich jene Tweets extrahiert werden, welche nach dem Preprocessing und der Scoreberechnung eine neutrale Bewertung erhalten haben. Diese Tweets könnten entweder tatsächlich keinerlei Sentiments enthalten und es würde sich in diesem Fall um ein true positive[10] handeln. Es wäre aber auch möglich, dass die neutrale Polarität als false positive[11] angegeben wurde und es sich eigentlich um einen Ausdruck im Tweet handelt, welcher nur dem Lexicon unbekannt ist. Bei Betrachtung des Kurztextes durch einen Menschen wird nämlich deutlich, dass der Tweet eine starke positive Emotion enthält (*Can't wait*).

Es kann beobachtet werden, dass es Worte, Ausdrücke und Idiome gibt, die in den bekannten Lexica nicht vorhanden sind. Diese möglicherweise falschen neutralen Tweets können nun von dem übrigen Hercules-Datensatz getrennt werden, um sie weiter für die Erweiterung des Lexicons zu nutzen. Ziel ist es, diese neue Teilmenge von vermeintlich neutralen Tweets weiter zu minimieren, indem tatsächlich neutrale Tweets von den *false positives* getrennt werden.

[9] Alternativ kann der trinary Paremeter verwendet werden, um als Bestimmungsmaß das Maximum beider Polaritäten zum Betrag zu verwenden.
[10] Korrekterweise neutral klassifizierter Tweet in diesem Fall.
[11] Fälschlicherweise neutral klassifizierter Tweet in diesem Fall.

Es gibt diverse wissenschaftliche Arbeiten wie [39] und [48][12], die darauf hindeuten, dass die Ausfilterung von neutralen Texten das Gesamtergebnis (accuracy) verbessert. Insbesondere in der Arbeit von Taboada et al. in [48] wird beschrieben, dass neutrale Texte *noisy data* generieren können und dabei wegen der Neutralität keinen Beitrag zum Gesamtergebnis der Sentiments liefern. Daher wird in Anlehnung an diese vorangegangenen Arbeiten im Weiteren auf die Einbeziehung neutraler Tweets in der Sentiment Analysis via SVM verzichtet[13]. Diese werden aber im Folgenden weiterhin zur Erweiterung des SentiStrength Lexicons verwendet.

Nachdem nun der reduzierte Datensatz nur noch neutrale Tweets enthält, kann dieser Datensatz zur Auffindung von neuen Ausdrücken verwendet werden. Zu diesem Zweck werden in einer Schleife alle neutralen Tweets durchlaufen. In dem Durchlauf wird in jedem Tweet ein Vergleich mit einem manuell erstelltem *slang lexicon* durchgeführt. Falls es eine Übereinstimmung in einem Tweet mit neutraler Polarität und einem slang Ausdruck gibt, wird dieser Tweet wiederum extrahiert und in einen neuen Datensatz namens *herc_pp_lex1* geschrieben. Herc steht hier für den Originaldatensatz Hercules, pp gibt an, dass der Datensatz das Preprocessing durchlaufen hat und lex1 gibt an, dass dieser Datensatz bereits mit dem slang lexicon iterativ abgeglichen und gefiltert wurde.

Die Auflistung 6.1 zeigt einen Auszug aus der neuen Wortliste. Zu beachten ist hierbei, dass der Ausdruck 12 beispielsweise bereits in der EmotionLookuptable von SentiStrength auftaucht. Allerdings wird der Begriff dort als wildcard[14] mit einer Negativwertung von -2 verwendet. Damit das dopppelte Auftauchen von bestimmten Worten in anderen Listen keinen Konflikt erzeugt, werden einige Ausdrücke in die IdiomList von SentiStrength eingetragen, wodurch sie eine höhere Priorität erhalten. Das bedeutet, dass alle anderen Vorkommen des Wortes in den anderen Listen (siehe dazu 6.1.1) durch die Bewertung in der neuen Idiomlist überschrieben werden .

Dier minimierte Datensatz *herc_pp_lex1*, der mit der Slang Liste abgeglichen wurde, muss nun stichprobenartig geprüft werden. Falls ein neutraler Ausdruck gefunden wird, der nach menschlichem Befinden ein Sentiment enthält, wird dieser der neuen lookuptable *Idiomstable* hinzugefügt. Es muss drauf geachtet werden, dass alle Variationen des Ausdrucks hinzugefügt werden. Oft schreiben User auf Twitter abgekürzte Verneinungen im Englischen ohne Trennzeichen[15]. Die Ausdrücke, die gefunden werden, werden manuell mit Labels bzw. Scores versehen. Im Rahmen dieser Arbeit wird der zusätzliche Wortschatz auf maximal 100 neue Ausdrücke limitiert. Eine größere Liste könnte in einer Dissertation oder computerlinguistischen Arbeit erstellt werden. Dafür könnte die in dieser Diplomarbeit erstellte Idiomliste als seed lexicon verwendet werden. Ähnliche seed lexicons werden beispielsweise in der Arbeit von Qiu et al. verwendet [43].

[12] S. 296ff

[13] Dies dient auch der Einfachheit, da klassische SVMs nur 2 Klassen kennen, die nur mit einer Anpassung Multiklassen behandeln können.

[14] * Notation am Wortstamm als regulärer Ausdruck

[15] Can't, Cant, Cannot

Tabelle 6.1: Neue Idiomliste

No.	Idiom	Score
1	can't wait	4
2	cant wait	4
3	cannot wait	4
4	thumbs up	3
5	gotta go see	3
6	have to see	3
7	unreal	-1
8	wanna watch	2
9	wanna see	2
10	must watch movie	4
11	blockbuster	2
12	bomb	3
13	attraction	2
14	mind blowing	4
15	mind-blowing	4
16	intimidat*	3
17	unexpected	3
18	gripping	3

6.1.4 Automatische Lexicon-Expansion

In diesem Unterkapitel wird prototypisch ein Schema skizziert, wie mittels eines initialen Seed-Lexicons eine Expansion realisiert werden kann. Das Verfahren wird in Pseudo-Code gezeigt und macht von der hierarchischen Struktur von WordNet Gebrauch.

Für dieses Verfahren muss initial eine Wortliste mit Sentiment Scores erstellt werden. Die initialen Scores können entweder durch SentiWordNet oder SentiStrength erstellt werden. Nach dieser Erstellung kann ein Algorithmus über diese Wortliste iterieren und nach Synsets suchen, die dieses Wort enthalten. Sobald ein Synset gefunden wurde, kann mittels eines Similarity-Measure die Ähnlichkeit bzw. semantische Distanz des Wortes zu den Worten im Synset berechnet werden. Bekannte Maße für diese Berechnung sind beispielsweise LIN (Lin, 1998)[16] oder RES (Resnik,1995)[17]. Sobald das Synset fertig ist, werden entsprechend der semantischen Distanzen Scores an diese neuen Worte vergeben. Die Scores sind die Scores des Initialworts gewichtet mit der semantischen Wortdistanz. Danach kann iterativ mit diesen Worten WordNet weiter durchsucht werden.

Diese Methode wird dann auf die selbe Art und Weise für Antonyme gestartet. Antonyme sind Worte, die die gegensätzliche Meinung eines Wortes darstellen. Es ist möglich diese Antonyme mittels einer Java API in WordNet zu finden[18]. Der Vorteil dieses Schrittes ist

[16] Lin, D. (1998, July). An information-theoretic definition of similarity. In ICML (Vol. 98, pp. 296-304).
[17] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007.
[18] `https://code.google.com/p/ws4j/`

es, dass gefundene Antonyme auf die selbe Art und Weise wie die Synsets zur Erweiterung des Lexicons verwendet werden können. Dazu muss lediglich der Score des Initialwortes umgedreht und mit der entsprechenden Wortdistanz gewichtet werden.

Die Scores können auch alternativ direkt mit SentiWordNet berechnet werden, allerdings haben in dieser Arbeit Tests aufgedeckt, dass SentiWordNet keine Umgangssprache oder Slang erkennt. Daher bietet es sich an, eine eigene Initialwortliste zu erstellen (das Seed-Lexicon), diese mit SentiStrength zu klassifizieren, und dann mit der beschriebenen Methode iterativ zu expandieren.

Diese Methode könnte in einer zukünftigen Arbeit implementiert und getestet werden. In dieser Diplomarbeit wurde prototypisch ein solches Tool implementiert. Allerdings wurde der Algorithmus nicht verwendet und fließt auch nicht in die Arbeit ein. Dazu müssten umfangreichere Tests und Probeläufe gemacht werden.

6.2 Modellierung

In dieser Arbeit werden zwei grundlegend verschiedene Ansätze verfolgt. Zum Einen wird mit einer lexikon-basierten Herangehensweise mittels SentiStrength Algorithmus eine Analyse durchgeführt. Dieser Algorithmus entnimmt den textuellen und emotionalen Inhalt und gibt eine Bewertung für jeden einzelnen Tweet aus.

Der zweite Ansatz ist der Machine-Learning Ansatz mittels Support Vector Machine (SVM). Für die SVM wird die Implementierung in Rapid Miner verwendet. Der SVM Algorithmus erwartet einen völlig anderen Typ von Inputformat für das Training und das Testen. Das heißt das Preprocessing aus Kapitel 5 muss erweitert werden, denn der SVM Algorithmus in Rapid Miner erwartet einen Wort-Vektor.

Im letzten Schritt wird gezeigt, wie mit einer Kombination beider Herangehensweisen Ergebnisse erzielt werden können. Dazu werden die Trainingsdatensätze mit SentiStrength mit Labels versehen. Diese Trainingsdatensätze werden dann dem SVM Algorithmus zum Trainieren übergeben, woraufhin der SVM Algorithmus das Testen starten kann. Durch diesen hybriden Ansatz kann ein maschinelles Lernverfahren ohne das Labeling-Problem gelöst werden. Es müssen nicht umständlich und manuell oder kostenintensiv Labels gesetzt werden, was einen großen Vorteil bietet.

6.2.1 SVM Modellierung in RapidMiner

In RapidMiner gibt es verschiedene SVM-Implementierungen. Unter Anderem gibt es einen linearen SVM Operator, der auf der in Java geschriebenen *JMySVM* basiert. Eine andere SVM Variante ist LibSVM. LibSVM wurde von Chih-Chung Chang and Chih-Jen Lin entwickelt. LibSVM bietet diverse Parameter und auch Möglichkeiten eine Regressionsanalyse durchzuführen. Für diese Arbeit wurden beide Variante auf einer Datenbasis geprüft. Die Wahl fiel auf LibSVM. Die Entscheidungsgrundlage war nicht die Performanz der Algorithmen. Für dieser Arbeit wird eine Multi-Class fähige Support Vector

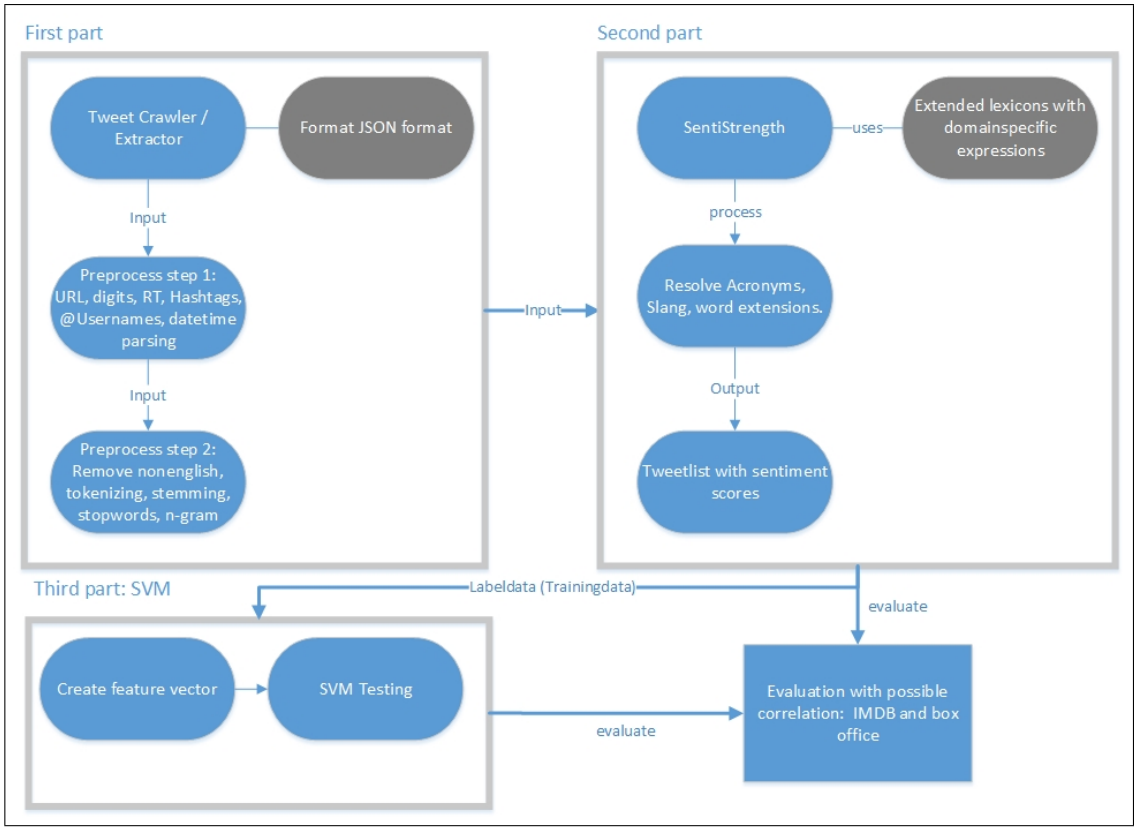


Abbildung 6.2: Herangehensweise

Machine benötigt. Es wäre zwar möglich und bei wissenschaftlichen Arbeiten in der Stimmungsanalyse nicht unüblich nur positive und negative Beispiele anzunehmen aber diese Ansicht wird in dieser Arbeit nicht vertreten. Das Fehlen einer neutralen Klasse würde all jene Tweets falsch klassifizieren, die weder positiv noch negativ sind. Das SVM Modell müsste also solchen Tweets eine positive oder negative Klasse „aufzwingen". In *The Importance of Neutral Examples for Learning Sentiment* [31] von Koppel et al. wird zur Relevanz der neutralen Klasse folgende Feststellung aufgeführt:

We have seen that in learning polarity, neutral examples cannot be ignored. Learning from negative and positive examples alone will not permit accurate classification of neutral examples. Moreover, the use of neutral training examples in learning facilitates better distinction between positive and negative examples.

Prinzipiell ist eine SVM für die Trennung von 2 Klassen mittels einer optimalen Hyperebene gedacht (siehe Kapitel SVM 3.6) und SVMs werden in der Stimmungsanalyse oft angewendet. Allerdings kann mittels Kombination mehrerer Lerner oder Kaskadierung auch ein Mehrklassenproblem gelöst werden. So könnten bei einem 3-Klassen Problem in der Sentiment Analysis zunächst die positiven und nicht positiven Tweets klassifiziert werden. Im nächsten Schritt würden alle nicht-positiven in 2 andere Klassen wie negative und neutrale klassifiziert werden. Am Ende würde man alle Ergebnisse zu einer Gesamtansicht zusammenfassen.

LibSVM erlaubt bereits in der Implementierung die Klassifizierung von Multi-Class Problemen. Der Operator kann für das Training am Inputport *polynominale* Labelwerte annehmen. Für das Testen können polynominale Kategorien klassifiziert werden. Daher eignet sich dieser SVM-Lerner besonders gut für die Stimmungsanalyse. Auch bezüglich der Performanz des Trainingsdatensatzes konnten keine größeren Unterschiede im Vergleich zur 2-Klassenvariante mit linearem SVM festgestellt werden[19]. In RapidMiner wird LibSVM mit den Standardparametern mit Typ C-SVC verwendet. Als Kerneltyp wird eine Polynomfunktion 2. Grades gewählt, da in einigen Tests in dieser Arbeit festgestellt wurde, dass der lineare Kerneltyp eine sehr geringe Performanz aufweiste[20]. Eine Erhöhung des Grades der Polynomkernelfunktion führte zu einer Verschlechterung der Accuracy. Der Grad 2 konnte daher heuristisch als beste Wahl ausgemacht werden. Alle anderen Parameter werden mit Standardwerten belassen, da dort keine Verbesserungen gefunden werden konnten[21].

Die Modellierung in RapidMiner ist unterteilt in Training und Testing. Zunächst werden die Trainingsdaten mit dem Retrieve Operator eingelesen. Dann müssen die Rollen der Spalten gesetzt werden (*Set Role Operator*). Bei den Trainingsdaten sind das die Rollen Tweettext und Sentiment. Der Output des Selects führt zum Operator *Process Documents from Data*. Dies ist ein sehr wichtiger Operator in der Textverarbeitung und ist ein

[19] Es gab in der Praxis einige Ergebnisse mit bis zu 10 % höherer Accuracy bei 2 Klassen SVM im Vergleich zu LibSVM mit 3 Klassen. Allerdings ist dieser Unterschied mit einigen Parameteranpassungen wie z.b. Poly-Kernel ausgeglichen worden.

[20] Die Accuracy lag zwischen 44 bis 51 %.

[21] Der C und gamma Parameter können bei LibSVM zur Vermeidung von Overfitting verwendet werden. Das Finden der einer optimalen Kombination der beiden Werte in der Multi-Class Sentiment Analysis SVM sollte in einer zukünftigen Arbeit detailliert betrachtet werden.

Container für mehrere Subprozesse. Viele Schritte wie die Entfernung der Stoppwörter aus dem Preprocessing 5 können in diesem Operator verwendet werden. Der Operator erstellt am Outputport einen sogenannten Wort-Vektor. Dabei handelt es sich um die Umwandlung des Dokuments (ein einzelner Tweet) in eine Vektordarstellung im Vektorraum. Dies ist im Text Mining ein notwendiger Schritt. Die Erstellung des Wort-Vektors kann nach verschiedenen Prinzipien wie dem einfachen Vorkommen eines *Tokens* oder dem TF-IDF 5.3 Verfahren geschehen. In dieser Arbeit wird TF-IDF verwendet.

Zunächst wird jeder Tweet in Tokens unterteilt. Die Tokens sind in diesem Fall Worte, getrennt durch Leerzeichen. Dann wird der Text in Kleinbuchstaben vereinheitlicht. Nachfolgend werden alle englischen Stoppwörter entfernt, die keinen Beitrag zur Stimmungserkennung leisten. Daraufhin folgt die Stammformreduktion mittels *Porter Stemming*. Bevor am Ende alle Tokens mit einer Länge von weniger als 2 Zeichen entfernt werden, wird mittels n-gram Operator ein 1-gram bzw. unigram erzeugt. Nach diesem Schritt ist die Vektordarstellung des Tweets fertiggestellt. Näheres zur Vektorisierung wird in 6.2.1 erklärt. Der Wort-Vektor wird dem Container Operator[22] *Validation* übergeben. Validation ist die eigentliche SVM Ausführung für den Trainingsteil und wird im Machine Learning zur Schätzung der Fehlklassifikation verwendet.

Mit Validation wird i.A. die sogenannte Kreuzvalidierung (k-fold X-Validation) durchgeführt, in der auf den zufällig ausgewählten Subsets $k-1$ ein Modell trainiert wird, um dieses dann auf einem anderen Subset k zu testen. Dieser Vorgang wird k-mal wiederholt, um jedes k einmal als Test und die restliche Menge als Training zu verwenden. Am Ende wird eine Schätzung mittels z.B. Durchschnitt der k Ergebnisse ausgegeben. Es ist anzunehmen, dass bei der Verwendung von $k = 10$ bei der k-fachen Kreuzvalidierung, die Fehlerrate hinreichend geschätzt werden kann[23]. In dieser Arbeit konnte festgestellt werden, dass die Fehlerrate sich bei $k = 5$ eingependelt hatte und sich danach nicht mehr stark änderte. Trotzdem stellte sich beim Experiment heraus, dass die tatsächliche Fehlerrate deutlich über der angegebenen Fehlerrate der Kreuzvalidierung liegen muss[24]. Der Grund liegt wahrscheinlich an dem Restbestand des Rauschens und einem allgemeinen Overfitting an das Trainingsmodell. Dies wird im Kapitel Experimente 7 vor allem durch einen Vergleich der SVM Methode mit der Lexicon Methode deutlich. Im Allgemeinen ist die Kreuzvalidierung eine bewährte Methode, um bei nicht vorhandenen Validierungsdaten einen Teil der Daten zum Trainieren und den Rest zum Testen zu verwenden.

Formal kann die K-fache Kreuzvalidierung [25][25] folgendermaßen definiert werden:

Definition 8. (K-fache Kreuzvalidierung):

Der vorliegende Datensatz wird in K etwa gleich große Partitionen aufgeteilt. Die Partitionen K-1 werden zum Trainieren und die k-te Partition zum Testen und Berechnen

[22] In RapidMiner als Nested Operator bezeichnet.
[23] Wissensentdeckung in Datenbanken SS2013, Prof. Morik u. Uwe Ligges, S.390.
[24] Diese Erkenntnis konnte durch einen Vergleich der Ergebnisse vom SVM Modell und Lexicon Modell gewonnen werden. Zusätzlich war die Vorhersagekraft des SVM Modells mittels Regression unter der des Lexiconansatzes verglichen worden.
[25] S.242f

Tabelle 6.2: Konfusionsmatrix

		Predicted: Positive	Predicted: Negative	Total
Correct	Positive	TP	TN	$TP + TN$
	Negative	FP	FN	$FP + FN$
	Total	$TP + FP$	$TN + FN$	N

des Prediction Error für die k-te Partition verwendet. Dieser Vorgang wird für alle k = 1,...,K *wiederholt und die K Schätzungen werden für die Prediction Error kombiniert.*

$\kappa : \{1, ..., N\} \rightarrow \{1, ..., K\}$ *sei eine Indexierungsfunktion zu einer Partition, zu der eine Beobachtung i zufällig allokiert wurde.* $\hat{f}^{-k}(x)$ *ist die Schätzfunktion für die entnommene k-te Partition. Die Prediction Error mittels Kreuzvalidierung ist dann:*

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i))$$

Zur Überprüfung des Modells wird der Operator *Performance Classification* am Ende der Validierung angehängt. Dieser kann diverse Performanzwerte in Abhängigkeit der Testergebnisse des Trainingsmodells liefern. Insbesondere die *Class Accuracy* ist eine der wichtigsten Kennzahlen bei der Klassifizierung. Sie gibt wie beschrieben an, wie genau die Schätzung der Klassenzugehörigkeit ist.

RapidMiner gibt die Performanzwerte in einer sogenannten *Konfusionsmatrix* wieder (Tabelle 6.2). In dieser Matrix wird die Gesamtperformanz des Modells visuell dargestellt. Die Konfusionsmatrix wird im Data Mining zur Beurteilung des Klassifikationsmodells verwendet und gibt Auskunft über die Menge an korrekt und falsch klassifizierten Klassen. Durch die Konfusionsmatrix werden die Gütekennzahlen *Precision* und *Recall* abgeleitet. Das sogenannte F-Maß ist eine Kombination von Precision und Recall. Es wird durch das harmonische Mittel[26] der beiden Werte berechnet.

TP steht in der Konfusionsmatrix für *True Positive* und gibt die korrekt klassifizierten positiven Beispiele an. FP steht für *False Positive* und gibt die Anzahl der Fehlklassifikationen der positiven Beispiele an. Analog gilt dies für TN und FN. Dies ist ein 2-Klassenbeispiel. In dieser Diplomarbeit gibt es noch zusätzlich eine 3. Spalte, welche analog den Fall der korrekten und fehlerhaften neutralen Beispiele darstellt. Die Kennzahlen zum Evaluieren des Modells ergeben sich aus den Spalten und Zeilen.

$$Precision = \frac{TP}{TP + FP} \tag{6.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

[26] Das harmonische Mittel ist das arithmetische Mittel der Kehrwerte.

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6.3}$$

Die Precision gibt den Anteil der positiv klassifizierten Beispiele an den tatsächlichen positiven Fällen an. Recall hingegen gibt den Anteil an korrekt positiv erkannten Beispielen an. Die *Accuracy* insgesamt, welche auch von RapidMiner bei den Ergebnissen ausgegeben wird, gibt an, wie gut das Modell gearbeitet hat. Sie wird berechnet durch

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{6.4}$$

Diese Kennzahlen können alle analog für den 3-Klassen Klassifikator und der neutralen Klasse verwendet werden.

Das gesamte Modell in RapidMiner kann in Abbildung 6.3 betrachtet werden.

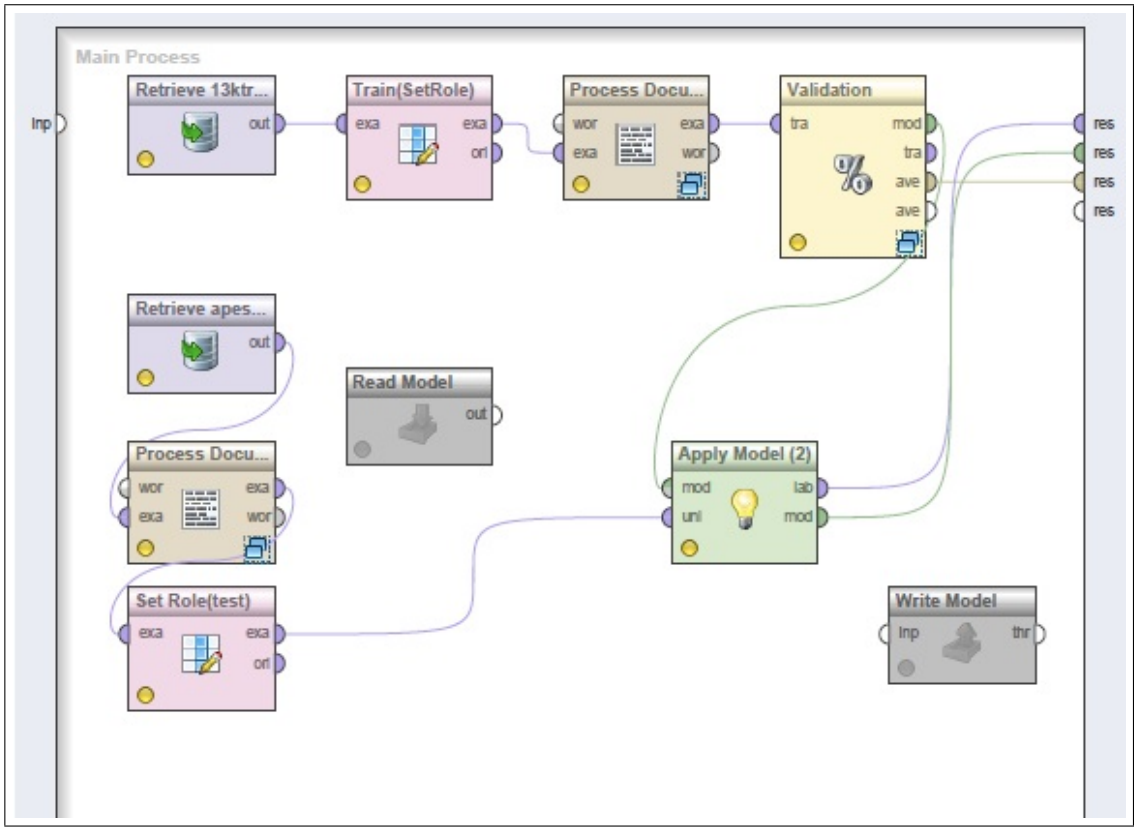


Abbildung 6.3: Modell in RapidMiner

Es ist wichtig zu wissen, dass die Accuracy, welche von dem Performanztest geliefert wird, **keine** Garantie der Zuverlässigkeit des Modells bezüglich der tatsächlichen Klassifizierungsgenauigkeit liefert. Sie ist lediglich ein Indiz und eine Hilfestellung, um verschiedene Parameter und Modellierungsvarianten zu testen. Es kann nicht mit absoluter Gewissheit gesagt werden, dass eine zufriedenstellende Performanz mit Kreuzvalidierung auch

mit exakt der selben Performanz auf einem realen Testdatensatz ohne Labels funktioniert. Selbst bei einer unendlich großen Trainingsmenge kann nicht davon ausgegangen werden, dass der tatsächliche Klassifikationsfehler ausreichend approximiert wurde.

Die Performanz einer SVM hängt von einigen Faktoren ab. Unter anderem muss ein ausreichend großer Trainingsdatensatz vorhanden sein[27], um die Zielfunktion gut genug approximieren zu können. *Übergeneralisierung* und *Overfitting* müssen vermieden werden. Beim Overfitting passt sich das Modell zu stark an die Trainingsbeispiele an und interpretiert unabhängige Testdaten dadurch falsch. Das Modell muss also weiter *generalisiert* werden. Overfitting kann aber auch bei unzureichender Trainingsbeispielmenge vorkommen. Es können Modelle entwickelt werden, die das Risiko der Fehlklassifikation minimieren. Dazu gibt es beispielsweise die *Risikoschranke nach Vapnik*.

Wort-Vektor Generierung: Bag-of-Words

Die Support Vector Machine Grundlagen wurden im Kapitel *SVM* (3.6) bereits erläutert. Wie dort beschrieben, erwarten SVM Algorithmen das Wort bzw. den Satz in einem Vektorformat. Ein Dokument muss für die Verwendung von Support Vector Maschinen vollständig vektorisiert sein, wobei ein Dokument in diesem Zusammenhang eine unbestimmte große Menge an Sätzen beinhalten kann oder aber ein einzelner Satz sein kann. Im Rahmen dieser Arbeit handelt es sich um einzelne Sätze, das heißt Tweets. Üblicherweise bestehen Tweets aus 1 oder 2 Sätzen. Ein Dokument könnte also beispielsweise ein Datensatz mit mehreren Tausenden Tweets zu einem Film sein.

Bei Support Vector Maschinen spielt der Wort-Vektor bzw. Feature Vector eine zentrale Rolle. Es gibt diverse Arbeiten, die sich mit dem Thema der Wahl von verschiedenen Features und des zugehörigen Feature Vektors beschäftigen. In [4] beispielweise werden mehrere „*feature families*“ verwendet. Zur Überprüfung der effektivsten Methoden werden dann in der Testphase nacheinander die Feature Sets entfernt. Am Ende stellt sich dann die ideale Menge an Feature Sets dar, welche die höchste accuracy erzielt.

In dieser Arbeit wird der Feature Vector mittels Bag-of-Words Modell generiert. Das Bag-of-Words Modell wird in Klassifikationsalgorithmen und NLP verwendet, um ein Vektormodell der Tweets zu erstellen. Dazu werden aus den vorhandenen Text-Dokumenten (einzelne Tweets) ein Vokabular bzw. Dictionary[28] aufgebaut. Dieses Vokabular besteht aus der binären unigram Vorkommens-Repräsentation der Gesamtmenge der Tweetdokumente nach der Vorverarbeitungsphase. Das bedeutet, es werden alle die aus der Vorverarbeitungsphase übriggebliebenen relevanten Einzel-Worte aller Tweets eines Datensatzes in einer Liste gespeichert. Zur Minimierung der Features werden also vorab nicht benötigte Worte eliminiert. Dies geschieht bereits im Preprocessing.

Diese Wort-Liste dient als Referenzvektor zur Vektordarstellung der einzelnen Tweets. Ein Tweet wird dann durch die Anzahl der Worte dargestellt, die der Tweet aus dem

[27] Fluch der hohen Dimension: Höhere Dimensionalität des Vektorraums erfordert mehr Trainingsbeispiele.

[28] Dictionary im Bag-of-Words Kontext ist nicht zu verwechseln mit dem Lexicon oder Dictionary der lexikalen Stimmungsanalyse.

Vokabular als Teilmenge enthält. Üblicherweise kann bei Bag-of-Words jeder Wert $n \in \mathbb{N}$ mit 0 angenommen werden. Die Anzahl des Vorkommens kann dann als Gewichtung weiterverwendet werden. Beispielsweise kann das häufige Auftreten eines positiven Emotionswortes im Gegensatz zu negativen Worten in einem Bericht über ein Fussballspiel ein Indiz dafür sein, dass die Mannschaft das Spiel gewonnen hat. Da es sich aber bei den einzelnen Wort-Vektoren um Tweets mit einer maximalen Zeichenanzahl von 140 handelt, wird aber in dieser in dieser Arbeit von einer einfachen Gewichtung ausgegangen. Dafür wird das Modell mit binären Werten verwendet, sodass die Gewichtung aus dem Vorhandensein oder Nicht-Vorhandensein eines Wortes besteht. Das heißt, ungeachtet der Anzahl der Wortvorkommen, erhält der Vektor die Werte aus der Menge $\{0, 1\}$.

Die Reihenfolge in der Repräsentation eines Tweets ist wichtig. Eine 1 an einer Stelle bedeutet, dass das Wort an der Stelle des Vokabulars in dem betrachtetem Tweet vorhanden ist. Eine 0 würde das Gegenteil bedeuten. Das Bag-of-Words Modell soll nun anhand eines Tweet-Beispiels gezeigt werden:

Ferner erlaubt Rapid Miner mit Operatoren einige weitere Minimierungen des Vektorraums, die auch in den SVM-Experimenten dieser Arbeit verwendet werden. Es können beispielsweise *Forward Selection* oder *Backward Elimination* auf die Features angewendet werden. Forward Selection initialisiert mit einer leeren Menge und fügt sukzessiv Features zur Menge hinzu und überprüft, ob eine Verbesserung der accuracy mit diesem Feature erzielt werden konnte. Dies wird in einer Schleife ausgeführt, solange Verbesserungen möglich sind. Backward Elimination erfolgt analog auf folgende Weise:

```
Initialisiere Featuremenge X = {1,...,n}
Entferne Feature x in X, so dass accuracy erhöht wird.
Refit model.
Wiederhole, solange Verbesserung möglich.
```

Hier wird also anders als bei der Forward Elimination bereits mit der Maximalmenge der Tweet-Features initialisiert. Auf Grund dieser Eigenschaft kann diese Methode rechenintensiver sein, aber möglicherweise auch besser[29].

N-gram Auswahl

Eine wichtige Rolle in der Modellierung bei der Sentimentanalyse spielt die Auswahl der sogenannten *n-grams*. Im obigen Beispiel wurde das unigram Verfahren angewendet. Es wird immer dann von unigram gesprochen, wenn für das n eines n-gram die 1 gewählt wurde. Dadurch bestehen die Wortvektoren aus einzelnen Worten, statt aus mehreren. Die Auswahl zwischen unigram, bigram etc. kann das Endergebnis der SVM Sentiment Analysis gravierend beeinflussen.

Ein n-gram Wort besteht im Falle von 1-gram, d.h. unigrams, aus genau einem Wort. Im Fall von 2-gram - auch bigram genannt, bestünde das n-gram Wort aus 2 Worten.

[29] `http://rapid-i.com/wiki/index.php?title=Feature_selection`

In der Anwendung einer der Verfahren bei der Feature Selection können große semantische Unterschiede entstehen. Dies soll anhand eines Tweet-Auszugs aus dem Herc_base Datensatz verdeutlicht werden:

*Hercules at the IMAX was f***ing unreal*

Die unigrams dazu sind: $\{\{Hercules\}, \{IMAX\}, \{f***ing\}, \{unreal\}\}$.

In diesem Fall würden insbesondere die lezten beiden Features für ein Ergebnis mit negativem sentiment sorgen, wenn sie einzeln betrachtet werden mit unigrams, denn das *f***ing* und *unreal* sind einzeln betrachtet negative Worte, die sowohl im SVM Trainingsdatensatz gelernt werden, als auch durch SentiStrength erkannt werden. Die Bigrams zu dem obigen Beispiel würden folgendermaßen aussehen:

$\{\{HerculesIMAX\}, \{IMAXf***ing\}, \{f***ingunreal\}\}$.

Die Auswertung dieser Featureliste würde positiver ausfallen, mindestens aber neutral, da der zusammenhängende bigram Ausdruck *f***ing unreal* ein positiv behaftetes Wort ist. Er könnte auch schon im manuel gelabeltem Trainingsdatensatz vorliegen oder zumindest in der Extended Lexicon (Idiom List) von SentiStrength 6.1.1.

6.2.2 Regressionsmodell zur Vorhersage

Die vorliegende Arbeit verwendet die multiple lineare Regression um die Vorhersagbarkeit des Erfolges eines Kinofilms anhand von a priori Datensätzen(Twitter) aufzuzeigen (Hypothese). Damit dieses Modell verwendet werden kann, wurden jeweils Tweets über ca. 7 Tage extrahiert. Nach einem Preprocessing wurden die Daten mittels Sentiment Analysis klassifiziert. Basierend auf dieser Klassifikation wurden *Prädiktoren* von diesen Daten abgeleitet. Ein Prädiktor ist hier eine Inputvariable der Regression. Beispielsweise ist die Gesamtzahl der positiven Tweets für einen Tag ein einzelner Prädiktor. Synonym für Prädiktor ist auch die Bezeichnung *unabhängige Variable*. In der multiplen linearen Regression gibt es im Gegensatz zur einfachen Regression mehrere Prädiktoren und genau eine[30] abhängige Variable bzw. Output, welche durch diese Prädiktoren beschrieben werden.

date	sales	#tweets	tweetrate	#pos	#neg	#neutral	sum(tot.score)	sum(pos.score)	sum(neg.score)	Tot.score avg	pos avg	neg avg	PT-NT ratio
-5	-	4237	177	1471	734	2032	1110	2030	-920	0,262	1,380	-1,253	2,00
-4	-	14511	605	4785	2706	7020	2554	6875	-4.320	0,176	1,437	-1,596	1,77
-3	-	17636	735	7787	1926	7923	7900	10736	-2.836	0,448	1,379	-1,472	4,04
-2	-	26713	1113	9783	3396	13534	11385	16279	-4.894	0,426	1,664	-1,441	2,88
-1	-	28073	1170	10835	2856	14382	12891	16775	-3.884	0,459	1,548	-1,360	3,79
0	11.058454	42313	1763	15538	4690	22085	16818	23630	-6.812	0,397	1,521	-1,452	3,31
1	10.226374	35753	1490	14644	4014	17095	17413	23033	-5.620	0,487	1,573	-1,400	3,65
2	8.515435	37246	1552	13925	5978	17343	13509	21163	-7.654	0,363	1,520	-1,280	2,33
3	3.337517	33591	1400	11043	4701	17847	11070	17592	-6.522	0,330	1,593	-1,387	2,35

Abbildung 6.4: Prädiktoren für den Hercules Datensatz

[30] In der multivariaten linearen Regression können auch mehrere abhängige Variablen vorhergesagt werden. Dies ist aber für diese Arbeit irrelevant, da der Erfolg nach Annahme hauptsächlich von dem Umsatzwert abhängt.

Die Abbildung 6.4 zeigt einen Auszug der Prädiktoren für die extended[31] Variante des Hercules Datensatzes. Die abhängige Variable ist der aufgerundete Tageswert der Ticketverkäufe des Films in der Spalte *sales*. Es handelt sich hierbei um Verkäufe in Millionen. Die Spalte *date* ist als einzige keine Variable und zeigt den zeitlichen Abstand der Tageszeile zum Erscheinungstag des Films an. Das heißt, die Zeile mit $date = -5$ gibt in diesem Fall in dieser Zeile die abgeleiteten Prädiktoren 5 Tage vor Erscheinungstag im Kino an. Dementsprechend gibt 0 den Erscheinungstag und alle positiven Werte die Tage **nach** Erscheinen an.

Diese Tabellen und Prädiktoren wurden bei der Datensatzerstellung zu jedem der 10 Filme jeweils **dreifach** erstellt. Es wurden also insgesamt 45 solcher Tabellen mit jeweils 14 Prädiktoren erstellt. Jede Variante steht für einen anderen Satz der Erstellung der Variablen. Die erste Variante ist das Ergebnis der Stimmungsanalyse mittels Lexicon Ansatz. Die zweite Variante ist der Lexicon Ansatz mittels expandiertem Lexicon. Die dritte Variante beeinhaltet Prädiktoren, die sich auf die Supported Vector Machine Experimente beziehen. Wegen der sehr hohen Berechnungszeit wird eine kleinere Auswahl der 10 Filme für SVM bereitgestellt. Insgesamt wurden mittels SVM 6 Filme klassifiziert.

Die allgemeine lineare Regression kann folgendermaßen definiert werden:

$$\hat{y} = \beta_0 + \beta_1 x \tag{6.5}$$

Das $\hat{y}$ ist der geschätzte Wert durch den einzelnen Prädiktor bzw. die unabhängige Variable x. Der Wert β_0 ist die Regressionskonstante und β_1 ist der Regressionskoeffizient. Bei dieser einfachen Gleichung sind nur lineare Abhängigkeiten von einer Variablen modellierbar. Bei der multiplen linearen Regression wird die Gleichung 6.5 um mehrere Prädiktorvariablen erweitert. Da in dieser Arbeit 14 verschiedene Prädiktoren identifiziert wurden und diese möglicherweise alle die abhängige Variable beeinflussen könnten, wird die Gleichung erweitert und resultiert in 6.6. Die Variable ϵ ist hierbei der Störfaktor. Sie gibt den Fehler an, welcher durch nicht beschriebene zufällige Einflüsse entsteht, die nicht durch die Prädiktoren verursacht werden. Die Störgröße sind die *Residuen* der Regression, d. h. die Differenz zwischen Regressionsgerade und Messwerten.

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \epsilon \tag{6.6}$$

Multikollinearität

Bei der Regressionsanalyse mit mehreren uabhängigen Variablen kann der Fall eintreten, dass die Variablen untereinander korrelieren. Dies kann dazu führen, dass die Regressionsgleichung nur wenig Aufschluss darüber gibt, welche Prädiktorvariable Einfluss auf die Zielvariable hat. Falls dies der Fall ist, spricht man von der *Multikollinearität*

[31] Prädiktoren wurden hier mittels Lexiconverfahren und domänenspezifischer Vokabularerweiterung durchgeführt.

der Variablen. Das heißt die unabhängigen Variablen sind untereinander abhängig. Es muss dann diagnostisch festgestellt werden, wie stark die Multikollinearität ist. Bei einer starken Multikollinearität müssen Maßnahmen ergriffen werden, um dieses Problem zu korrigieren oder zu mindest abzuschwächen. Allgemein wird die Multikollinearität als Bedrohung für ein statistisches Modell betrachtet, da die zu schätzende Abhängigkeit zur Zielvariablen in der Regressionsgleichung dadurch beeinträchtigt wird [20]. Anders ausgedrückt kann gesagt werden, dass zwei stark untereinander korrelierende unabhängige Variablen mit der jeweils anderen Variablen ausgedrückt werden können, um die abhängige Variable zu erklären. Es gibt also eine redundante Variable, die eliminiert werden könnte.

Es gibt einige Methoden, um den Grad der Multikollinearität von Variablen festzustellen. Diese Methoden sind in der Literatur nicht einheitlich oder standardisiert. In dieser Arbeit wird die Multikollinearität mit *Korrelationsmatrizen* behandelt. Dies ist eine übliche Vorgehensweise. Dazu wird die *Pearson Korrelation* einer jeden Prädiktorvariable in Matrixform zu jeder anderen Variable berechnet und aufgeschrieben. Dadurch werden in einer einzigen Übersicht alle stark korrelierenden Variablen sichtbar. Diese sind durch einen hohen Pearson Korrelationskoeffizienten bemerkbar(siehe Abbildung 6.5). Die allgemeine Pearson Korrelation berechnet sich durch

$$r = \delta(X, Y) = \frac{Cov(X, Y)}{\sigma(X)\sigma(Y)} \tag{6.7}$$

$Cov(X, Y)$ dient der Berechnung der Kovarianz der beiden Variablen X und Y. σ ist die Standardabweichung. In unserem Fall wird die Pearson Korrelation für eine Messreihe gepaarten Messungen verwendet. Dann lautet die Formel umgeformt

$$r_{xy} = \frac{\sum_{i=0}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=0}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}} \tag{6.8}$$

Wobei gilt:

- $-1 \leq r_{xy} \leq 1$
- $r_{xy} = +1 \Leftrightarrow y_i = ax_i + b$ mit $a > 0$ (größte positive Korrelation)
- $r_{xy} = -1 \Leftrightarrow y_i = ax_i + b$ mit $a < 0$ (größte negative Korrelation)
- $r_{xy} = 0$ (kein linearer Zusammenhang)

Diese Formel wird auch *Bravais-Pearson Korrelationskoeffizient* genannt.

Bei einem r ab 0,3 wird dies als mäßige Korrelation betrachtet. Bei Werten mit $r > 0,5$ liegt per Konvention eine starke Korrelation vor. Positive Korrelationen sagen aus, dass große Werte der einen Variablen gleichmäßig in Relation mit großen Werten der anderen Variablen auftreten. Negative Korrelation verläuft derart, dass große Werte mit kleinen Werten der anderen Variable auftreten.

In diesem Auszug der Korrelationsmatrix 6.5 ist abzulesen, dass einige Variablen stark korrelieren. Das liegt vor allem daran, dass die Variablen einige Gemeinsamkeiten haben. Beispielsweise ist es ein erwartetes Symptom, dass die Anzahl der positiven Tweets mit der Gesamtanzahl aller Tweets steigt. Auch die Tweetrate korreliert auf natürliche Weise mit der Gesamtzahl an Tweets, da die Tweetrate eine Kombination aus der Tweetzahl ist. Daher werden bei diesen Variablen starke Korrelationen bei allen Datensätzen erwartet. Dieses Problem wird dadurch behandelt, dass nur eine dieser stark korrelierenden Variablen im Regressionsmodell verwendet wird, da sie redundant sind.

```
> cor(hercdata,use="complete.obs",method="pearson")
                    date      sales sale_per_cin   X.tweets  tweetrate      X.pos      X.neg
date           1.0000000  0.8880168    0.8879607 -0.8669789 -0.8669248 -0.9091415 -0.61075771
sales          0.8880168  1.0000000    1.0000000 -0.9329679 -0.9329504 -0.9388634 -0.78549975
sale_per_cin   0.8879607  1.0000000    1.0000000 -0.9329534 -0.9329359 -0.9388326 -0.78547108
X.tweets      -0.8669789 -0.9329679   -0.9329534  1.0000000  1.0000000  0.9826327  0.90427226
tweetrate     -0.8669248 -0.9329504   -0.9329359  1.0000000  1.0000000  0.9826217  0.90428424
X.pos         -0.9091415 -0.9388634   -0.9388326  0.9826327  0.9826217  1.0000000  0.84890368
X.neg         -0.6107577 -0.7854998   -0.7854711  0.9042723  0.9042842  0.8489037  1.00000000
X.neutral     -0.8618139 -0.9241724   -0.9241739  0.9922938  0.9922988  0.9597884  0.88066146
sum.tot.score. -0.9411381 -0.9625491   -0.9625360  0.9401522  0.9401389  0.9766999  0.73826046
sum.pos.score. -0.9056084 -0.9635303   -0.9635149  0.9848107  0.9848023  0.9956786  0.85172561
sum.neg.score. -0.6298693 -0.7749999   -0.7749814  0.9100916  0.9100982  0.8493980  0.98937542
Tot.score.avg -0.5703166 -0.3603424   -0.3602473  0.1924261  0.1922784  0.3490778 -0.05814679
pos.avg       -0.4520752 -0.3415170   -0.3415412  0.3178800  0.3178783  0.3150896  0.07420225
neg.avg        0.4420806  0.5171844    0.5170386 -0.3988438 -0.3989440 -0.4787219 -0.36064960
PT.NT.ratio   -0.6490080 -0.4388985   -0.4388216  0.3424057  0.3423471  0.4730159 -0.01830222
```

Abbildung 6.5: Auszug aus Korrelationsmatrix des Datensatzes Hercules

Der Vorteil der Korrelationsmatrix ist die einfache Erstellung und schnelle Erkennung von paarweisen linearen Korrelationen. Ein großer Nachteil ist, dass nur paarweise Relationen aufgedeckt werden und Variablenkombinationen nicht berücksichtigt werden. Variablenkombinationen spielen aber besonders bei der multiplen linearen Regression eine Rolle. In dieser Arbeit wird zusätzlich der sogenannte *Variance Inflation Factor* (VIF) und die *Tolerance* (TOL) berechnet. Diese beiden Werte können auch als Indikatoren für etwaige Multikollinearität verwendet werden. Der VIF eignet sich besser zur Aufdeckung von Multikollinearität ([28], S.101). Der VIF nimmt Werte > 1 an und hat keine obere Grenze. Typischerweise werden in der Literatur VIF Werte ab > 10 als problematischer Grad an Multikollinearität betrachtet ([28],[23]). Dieser Wert ist aber eher eine *rule of thumb* als ein festgelegter Grenzwert[32].

Der VIF wird für jede einzelne Prädiktorvariable berechnet und kann mit der Formel

$$VIF_j = \frac{1}{1 - R_j^2} \tag{6.9}$$

berechnet werden. Jeder Prädiktor x_j hat also einen eigenen VIF Wert entsprechend der Gleichung 6.9. Das R_j^2 ist das *Bestimmtheitsmaß*, welcher als Gütemaß der Regression aussagt, in welchem Ausmaß die Varianz der abhängigen Variable durch die Prädiktorvariablen erklärt werden kann. Eine allgemeine Definiton des Bestimmtheitsmaßes ist

[32]Es gibt auch Arbeiten mit VIF threshold von 25.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{6.10}$$

mit den Summen der quadrierten Residuen

$$SS_{res} = \sum_{i}^{n} (y_i - \hat{y_i})^2 \tag{6.11}$$

und den Gesamtsummen der quadrierten Summen der Differenzen von der abhängigen Variable und des arithmetischen Mittels $\bar{y}$ durch die Gleichung

$$SS_{tot} = \sum_{i}^{n} (y_i - \bar{y})^2, \tag{6.12}$$

wobei $\hat{y_i}$ ein vorhergesagter Wert ist und y_i ein beobachteter Wert. In dieser Diplomarbeit wir der *adjusted R-Square* bzw. das *korrigierte* R^2 verwendet. Dieser bezieht auch die Anzahl der Prädiktorvariablen ein und ist generell kleiner als der normale Wert von R^2.

Es kann auch der *Tolerance* (TOL) einer Prädiktorvariablen berechnet werden. Dieser ist der Kehrwert des VIF und nimmt Werte zwischen 0 und 1 an. Der TOL Wert kann also mittels

$$TOL_j = 1 - R_j^2 = \frac{1}{VIF} \tag{6.13}$$

berechnet werden. TOL Werte nahe bei 0 geben an, dass eine starke Multikollinearität vorliegt. Hohe Werte gegen 1 hingegen bedeuten keine Multikollinearität. Auch bei TOL gibt es keinen einheitlichen Konsensus bezüglich der Interpretation oder des Threshold.

Bei der Verwendung des Bestimmtheitsmaßes R^2 ist zu berücksichtigen, dass dieses zwar aussaugt wie gut die Regression die tatsächlichen Datenpunkte approximiert (Wert 1 sagt bspw. aus, dass perfekte Übereinstimmung vorliegt), aber es wird **keine** Aussage darüber gemacht, **ob** ein kausaler Zusammenhang zwischen der abhängigen und den unabhängigen Variablen besteht. Eine hohe Korrelation bzw. ein hoher R^2 Wert beweist weder hinreichend noch notwendig eine Kausalität und kann lediglich als Indiz und Hinweis dienen.

Es ist außerdem wichtig zu erwähnen, dass die Multikollinearität ein wichtiges Thema in der Literatur des statistischen Lernens ist. Die vorhandene Multikollinearität wird in der Literatur generell zwar als potentielles Problem betrachtet, aber es gibt auch Kritiker, die vor einer Überbewertung von vorhandener Multikollinearität oder dem Automatismus der Verwerfung von Prädiktoren mit hohen VIF warnen. [37] kritisiert, dass die Eliminierung von Prädiktorvariablen wegen hoher Multikollinearitäten (berechnet durch hohe VIF) ein Fehler sein kann, da dadurch die theoretische Grundlage sich ändert[33].

[33]The problem with this solution is that dropping X_j from the equation means that the ith regression coefficient no longer represents the relationship between the Y and X_i controlling for X_j and any other

Insbesondere Goldberger et al. wird bei kritischen Betrachtungen von Multikollinearitäten oft erwähnt [22]. Goldberger kritisiert vor allem, dass in vielen Büchern die Multikollinearität und mögliche schwere Folgen diskutiert werden. Aber es würde oft nicht Erwähnung finden, dass die Multikollinearität und die stark korrelierenden Prädiktoren deswegen auftreten könnten, weil eine zu kleine Prüfmenge (Sample) vorliegt. Er schreibt eine Parodie über diese Thematik und führt dazu den neuen Begriff der *Micronumerosity* ein. Dadurch möchte er verdeutlichen, dass das eventuelle Fehlen einer ausreichend großen Sample Datenmenge nicht ignoriert werden sollte, falls eine Multikollinearität geahnt wird.

Tatsächlich wurden in dieser Arbeit teilweise starke Multikollinearitäten in den unabhängigen Variablen festgestellt (siehe nächstes Kapitel). Diese werden auch auf die kleine Fallzahl (Sample Größe) zurückgeführt. Die Twitter Datensätze beinhalten für manche Filme Streamdaten über 6-10 Tage. Es gibt aber auch Filme mit Daten über ca. 3 Wochen. Es sind zwar sehr viele Tweets vorhanden, aber diese mussten aus technischen Gründen als Tagesbasis berechnet werden, da es keine Quelle gibt, um Verkaufszahlen (unabhängige Variable) innerhalb eines Tages zu erfahren. Es ist nur möglich, für jeden Film den Tagesumsatz zu erfahren. Einige weniger bekannte Filme haben außerdem nur Umsätze für Wochenenden bekanntgegeben. Dieser Problematik wurde zu einem Teil entgegengewirkt, indem ein neuer Datensatz während der Erstellung dieser Diplomarbeit extrahiert wurde, welcher über mehrere Wochen an Tagesdaten enthält. Dieser Datensatz gilt als Prüfsatz, um die Glaubhaftigkeit der Ergebnisse für die weniger großen Datensätze zu überprüfen.

Zuvor war es möglich über den Twitterstream historische Daten abzugreifen. Dadurch wäre es möglich gewesen, nachträglich Twitterdaten zu vergangenen Filmen zu extrahieren. Twitter hat aber diese Möglichkeit von der Serverseite aus gesperrt. Es ist technisch nicht mehr möglich ältere Tweets des allgemeinen Streams zu erhalten, die länger als 7 Tage zurückliegen.[34]

6.2.3 PT-NT Metrik

Im Rahmen dieser Arbeit wird im Allgemeinen eine Vorhersage der Verkaufszahlen mittels multipler linearer Regression durchgeführt. Es gibt aber auch zwei Arbeiten, welche den Erfolg von Filmen mittels Twitterdaten über eine sogenannte PT-NT Ratio vorhersagen ([27],[8]). Es handelt sich hierbei um einen diskreten Wert, der nicht direkt aus einem statistischen Modell entnommen wird. Der PT-NT Wert wird aus den Ergebnissen der Stimmungsanalyse abgeleitet. Dabei wird insgesamt für jeden Film ein PT-NT Ratio berechnet. Daraufhin wird ein allgemeiner Threshold bzw. Grenzwert festgelegt.

independent variables in the model. The model being tested has shifted, and this often means that the theory being tested by the model has changed.Simply dropping X_j because it is highly correlated with the X_i or using step-wise regression to select variables (typically the selected variables are not „too highly correlated“ with each other) leads to a model that is not theoretically well motivated. S.683 in Quality & Quantity (2007) 41, DOI 10.1007s1113500690186

[34] Lediglich bezahlte Dienste wie Topsy können historische Daten bereitstellen. Diese Möglichkeit ist für diese Arbeit keine Alternative.

Graduell werden Filme mit einem niedrigem PT-NT als weniger erfolgsversprechend deklariert. Analog werden Filme mit sehr hohem PT-NT als sehr erfolgreich vorhergesagt. Der PT-NT Ratio wird in dieser Arbeit auch als zusätzliche Prädiktorvariable verwendet.

Die vorliegende Arbeit wird zu dem linearen Modell diese PT-NT Metrik berechnen und zum Vergleich heranziehen. Der PT-NT Wert errechnet sich durch

$$PTNTratio = \frac{|positiveTweets|}{|negativeTweets|}. \tag{6.14}$$

7 Experimente: Stimmungsanalyse der Datenbasis

Die Experimente in diesem Kapitel sind unterteilt in SentiStrength und SVM. Es werden also Lexiconanalyse-Ergebnisse und Support Vector Machine Ergebnisse aufgeteilt gezeigt. Die Experimente werden exemplarisch an einigen ausgewählten Datensätzen gezeigt. Die Ergebnisse der Stimmungsanalyse für **alle** Datensätze werden daraufhin aufgelistet. Die Ergebnisse dieser Stimmungsanalyse sind Statistiken und Prädiktorvariablen, welche in ein lineares Regressionsmodell eingegeben werden, um anschließend t-Tests durchzuführen.

Es werden die Experimente und Ergebnisse folgender 9 Titel gezeigt: *A Long Way, A Most Wanted Man, The Purge, Guardiands of the Galaxy, Hercules, I Origins, Dawn of the Planet of the Apes, Boyhood, Calvary.* Die Ursprungsliste umfasst 15 Filme. Jedoch werden die Ergebnisse einiger Filme wie *Child of God* wegen zu geringer Tweetzahlen nicht gezeigt.

7.1 SentiStrength Experimente

Der erste Datensatz ist der Datensatz zur Tweetkollektion des Films *Hercules*. Hercules gehört zu keinem größeren Franchise und ist weder Prequel noch Sequel. Diese Zusatzinformationen werden für die anfängliche deskriptive Analyse nur zwecks Vollständigkeit erwähnt, haben aber ansonsten in dieser Analyse keinen Einfluss. Es gibt einige Arbeiten, die Faktoren wie Franchise, Autor, Schauspieler als Zusatzfaktoren (siehe Kapitel Verwandte Arbeiten 2) verwenden. Die Abbildung 7.1 zeigt die offiziellen Verkaufszahlen des Films[1]. Für diese Arbeit werden anhand des Hercules Datensatzes die Experimente und Ergebnisse gezeigt. Dieser Datensatz dient als Exempel für alle anderen Datensätze. Falls nicht anders erwähnt, unterlaufen alle Filmdatensätze die selben Abläufe wie dieser examplarische Datensatz.

Hercules hat weltweit insgesamt 243 Millionen Dollar eingespielt. Davon sind rund 72 Millionen Dollar aus den heimischen Kinos in den USA erfolgt. Die Produktionskosten belaufen sich auf 100 Millionen Dollar. Das heißt, der Film konnte sich nur durch die ausländichen Kinobesucher finanzieren. Sie brachten mehr als doppelt so viel Geld ein wie die Besucher aus den USA. Die Grafik zeigt den zeitlichen Ablauf der Verkäufe, angefangen mit dem Release am 20.07.2014. Die Rank-Linie sagt aus, dass der Film beim Eröffnungswochenende die zweithöchsten Verkäufe aller Filme erreicht hat. Dieser

[1] Verkaufszahlen: `http://www.boxofficemojo.com/`, `http://www.the-numbers.com/`

Date	Rank	Gross	% Change	Theaters	PerTheater	Total Grossin	Days
25.07.14	2	11.058.454		3595	3076	11.058.454	1
26.07.14	2	10.226.374	-8,00%	3595	2845	21.284.828	2
27.07.14	2	8.515.435	-17,00%	3595	2369	29.800.263	3
28.07.14	2	3.337.517	-61,00%	3595	928	33.137.780	4
29.07.14	2	3.780.503	13,00%	3595	1052	36.918.283	5
30.07.14	2	2.614.620	-31,00%	3595	727	39.532.903	6
31.07.14	2	2.115.145	-19,00%	3595	588	41.648.048	7
01.08.14	4	3.170.032	50,00%	3595	882	44.818.080	8
02.08.14	4	4.376.365	38,00%	3595	1217	49.194.445	9
03.08.14	4	3.463.970	-21,00%	3595	964	52.658.415	10
04.08.14	3	1.421.112	-59,00%	3595	395	54.079.527	11
05.08.14	3	1.631.475	15,00%	3595	454	55.711.002	12
06.08.14	3	1.155.389	-29,00%	3595	321	56.866.391	13
07.08.14	3	894.856	-23,00%	3595	249	57.761.247	14
08.08.14	7	1.766.639	97,00%	2896	610	59.527.886	15
09.08.14	6	2.361.945	34,00%	2896	816	61.889.831	16
10.08.14	6	1.618.307	-31,00%	2896	559	63.508.138	17

Abbildung 7.1: Hercules box office

Rang wurde bis zum 31.07.2014 gehalten. Danach gingen die Verkaufszahlen wie auch der Rang deutlich runter. An der Abbildung sind einige Beobachtungen möglich. So kann die auch für die Tweets wichtige Beobachtung gemacht werden, dass der sehr hohe Verkauf am Releasetag nie wieder erreicht wird. Der erste Tag ist der Peak im Verlauf. Danach gehen die Zahlen in bestimmten Abständen runter. Es ist auch zu sehen, dass die Verkäufe einen Zickzack-Verlauf haben. Es ist zu überprüfen, ob dieser charakteristische Trendverlauf auch bei den Tweets stattfindet. Falls die Tweetrate, die Anzahl der Tweets und die Tweetsentiments den selben Verlauf aufweisen, könnten dies erste Indizien für einen Zusammenhang sein. Falls dieser Zusammenhang mit den vorliegenden Daten **vor** dem Releasetag berechnet werden kann, würde dies unter Umständen den Aufbau eines prädiktiven Modells erlauben.

Periodisch gibt es kleinere Spitzen, die dann in den selben periodischen Intervallen wieder sinken. Diese Spitzen treten an exakt jedem Freitag auf und halten an bis zum Montag. Es ist also offensichtlich, dass viele Menschen an den Wochenenden den Film besuchen. Diese Beobachtung wird nachfolgend mit den Tweets verglichen werden. Eine andere nicht direkt sichtbare Tatsache ist, dass an Samstagen deutlich mehr Besuche stattfinden als an Freitagen oder Sonntagen. Der Samstag ist fast immer der höchste Verkaufstag, wenn man von den Eröffnungs-Freitagen absieht. Deutlich zu sehen ist dies am 04.08.2014 mit 4,3 Millionen Dollar, der sogar die Verkaufszahlen vom 28.07.2014 übertrifft. Dieser sehr deutliche Peak liegt vermutlich an der Eröffnung des Films in einem anderen Land. Diese Information konnte aber in den Recherchen nicht bestätigt werden, da es keine öffentliche Datengrundlage dafür gibt.

Diese Beobachtungen konnten auch bei den anderen Filmen gemacht werden. Es kann verallgemeinert gesagt werden, dass alle Filme ihre höchsten Einnahmen am Veröffentlichungstag machen und dann einen Abwärtstrend beginnen. Zu einigen der wenigen bekannten Filmen gab es keine Möglichkeit mehrere Tage an Verkaufsdaten zu erhalten, da diese oft nach dem ersten Wochenende entweder aus dem Kino genommen wurden, oder von den Webseiten wie boxofficemojo nicht mehr weiterverfolgt wurden.

7.1.1 Experimente mit Originallexicon und expandiertem Lexicon

Nachfolgend werden die Box Office Daten mit den Ergebnissen der lexikalen Sentiment Analysis verglichen. Zunächst folgen die Tagesübersichten ***vor*** dem Veröffentlichungstag des Films. Die Datenextraktion beim Hercules Datensatz beginnt am 20.07.2014. Der Kinostart war am 25.07.2014. Es wurden also täglich die Tweets 6 Tage zuvor extrahiert und analysiert. Die Abbildung 7.2 zeigt den Tagestrend sämtlicher Tweets zum Film an diesem Tag. Insgesamt gab es in dem Zeitraum 4.237 Tweets. Zwei Drittel dieser Tweets sind von leichter bis starker positiver Polarität. Der durchschnittliche Sentiment Score des Tages beträgt $0,31$[2]. Also kann die Aussage gemacht werden, dass die potentiellen Kinogänger fünf Tage vor Erscheinungsdatum eher positive als negative Emotionen gegenüber dem Film haben.

Wichtig ist zu beachten, dass die Abbildung 7.2 mit dem Originallexicon von SentiStrength durchgeführt wurde. SentiStrength beeinhaltet bereits im Gegensatz zu anderen Stimmungsanalyse-Werkzeugen sehr viele umgangssprachliche Ausdrücke, insbesondere aus dem Bereich Social Media. Daher wird eine bereits hohe Accuracy bezüglich der korrekten Klassifizierung der Labels erwartet.

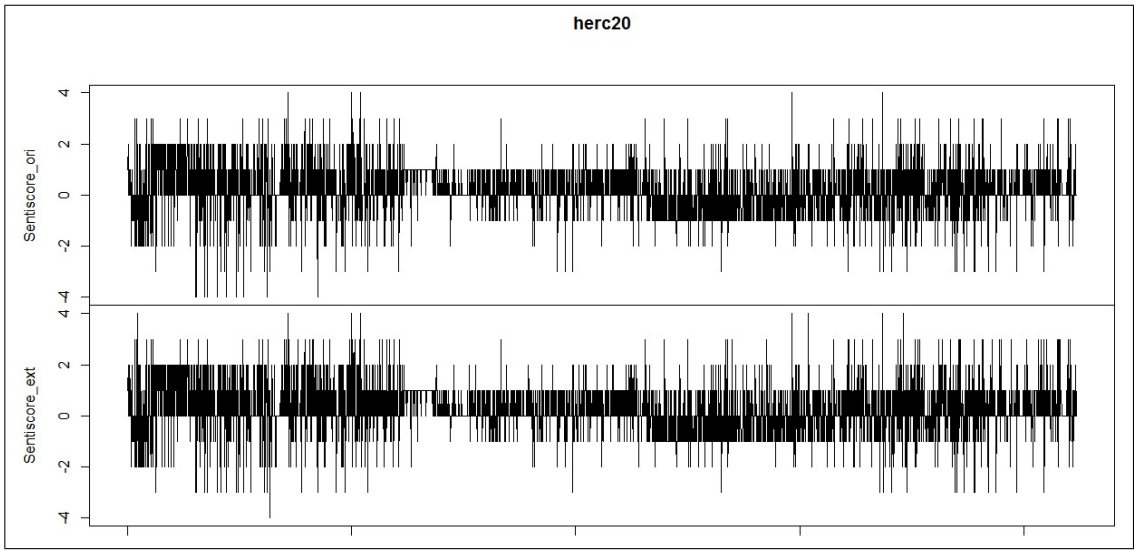


Abbildung 7.2: Hercules Intraday 20.07.2014

Am nächsten Tag wurden deutlich mehr Tweets zum Film erstellt. Am 21.07.2014 waren es bereits 27.265 Tweets (7.4). Besonders auffällig ist die Änderung zum Vortag bezüglich der Anzahl an positiv klassifizierter Tweets: über 60 % aller gesendeten Tweets werden mittels Originallexicon als positiv erkannt. Auch ist der Anteil an negativen Tweets mit 12,5 % deutlich unter dem Vortag. Neutral oder aber potenziell nicht erkannte Polarität haben etwa 26,8 % aller Tweets. Auch der Sentiment Score Average liegt mit 0,60 fast doppelt so hoch wie am Vortag. In der Analyse des Datensatzes konnte ein plausibler

[2]Neutrale Tweets wurden hier exkludiert, da dieser Teil der Analyse noch mit dem Originallexicon durchgeführt wird. Das bedeutet, dass möglicherweise viele der neutralen Tweets mit dem Extended Lexicon noch einer Polarität zugewiesen werden könnten.

Grund für diese plötzliche Erhöhung der Tweets gefunden werden. Die Annahme war, es könnte sich um eine neu gestartete Twitter-Werbekampagne zum Film handeln, die einen positiven Effekt hatte. Dieser Tag hebt sich im Vergleich zu den anderen Tagen deutlich ab. Zur Auffindung des Grundes dieses großen Unterschiedes wurde zunächst der Ansatz verfolgt, User mit besonders vielen Followers zu filtern. Gesucht wurden die 100 höchsten Follower-User, die wegen ihrer Abonennten gegebenenfalls sehr viele Retweets erhalten haben könnten. Diese könnten ein Indiz für die Erhöhung liefern.

Tatsächlich konnte am Ende ein User identifiziert werden, der mit 7.390.889 Followern am 21.07.2014 eine extrem hohe Retweet-Rate erzielt hat (siehe Abbildung 7.3). Insgesamt wurde ein bestimmter Tweet von diesem User um 01:01:40 getwittert und erhielt innerhalb von 22 Stunden 12.122 Retweets. Der Tweet kann auf Twitter eingesehen werden[3]. Der Autor des Tweets ist der Hauptdarsteller des Films und beeinflusste die Tweetrate und die Stimmung an diesem Tag **extrem**.

Abbildung 7.3: Hauptdarsteller; Tweet mit sehr hohem Retweet

Diese Retweets werden daher im modifizierten Datensatz nun als Noise bzw. Duplikate bewertet und fließen nicht mehr in die Gesamtbewertung ein. Insgesamt handelt es sich hierbei um 12.122 Tweets mit einer jeweiligen positiven Bewertung von 1. In dieser Diplomarbeit werden aber generell Retweets ***nicht*** als Duplikate entfernt, da es sich nach dem Preprocessing bei einigen Tweets um scheinbare Duplikate handelt, diese aber eigentlich Tweets von verschiedenen Usern sind. Das bedeutet, textuell handelt es sich nach der Vorverarbeitung um Duplikate. Im Rahmen dieser Sentiment Analysis sind diese scheinbaren Duplikate aber wichtige Indikatoren für die Polaritätsmessung. Ein Retweet

[3] https://twitter.com/therock/status/491004039216119809

findet bei Twitter meistens nur dann statt, wenn der User sich mit dem Originaltweet von der Polarität her in Übereinstimmung befindet. Es kann daher nicht von Noise oder Datenverunreinigung gesprochen werden. Der Tweet des Hauptdarstellers 7.3 ist hier eine Ausnahme, da die Retweets zu eine massiven Fälschung der Stimmung vor diesem Tweet bei Twitter geführt haben. Es wurde hierbei deutlich, wie schnell durch einen User mit hoher Abonenntenzahl die Gesamtstimmung bezüglich eines Themas auf Twitter beeinflusst werden konnte[4]. Betrachtet man den trendhaften Verlauf in Abbildung 7.7, wird schnell deutlich, dass die Entfernung dieser Duplikate für diesen Tag ein logischer Schritt ist.

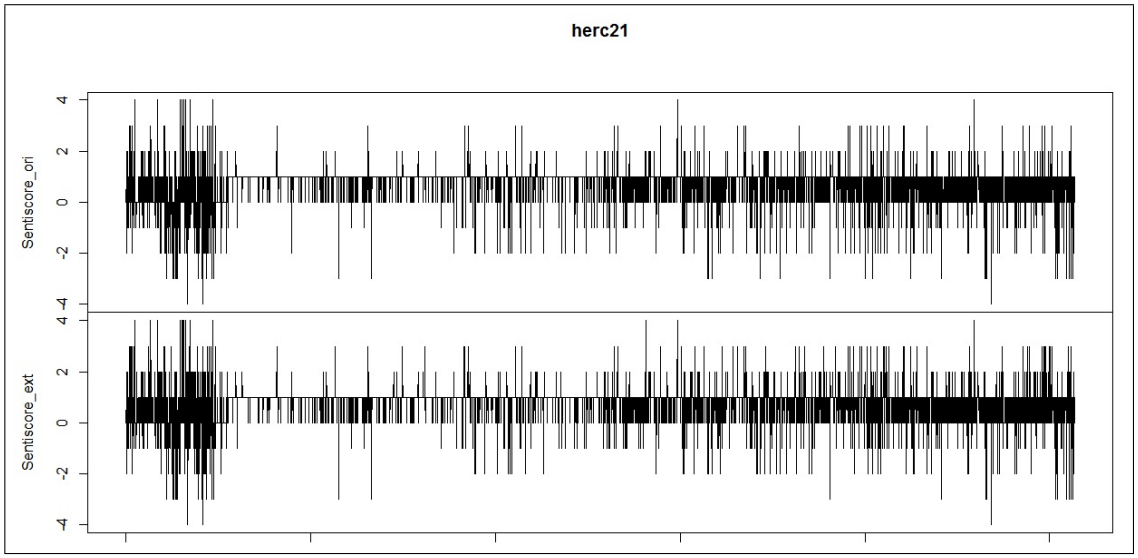


Abbildung 7.4: Hercules Intraday 21.07.2014

Nach Entfernung der Retweets des Darstellers liegt Score-Durschnitt bei 0,26 und ist positiv mit fallender Tendenz im Vergleich zum Vortag. Der Durschnitt ist somit weniger als halb so hoch, falls die Retweets nicht gezählt werden, jedoch immer noch positiv.

Abbildung 7.5 zeigt den Verlauf der Tweets und zugehörigen Scores am Veröffentlichungstag. Es ist eine deutliche Erhöhung aller Tweets zu beobachten. Auch die generelle positive Stimmung hat zugenommen. Der durchschnittliche Score aller Tage bis zum Release betrug 2.89. Am Releasetag gibt es eine Erhöhung auf insgesamt durschnittlich 3.31 und am nächsten Tag auf 3.65. Dies deutet auf eine positive Reaktion der Besucher kurz vor und nach dem Film hin. Tatsächlich erreicht der Folgetag fast die selbe Summe an Besuchern wie am Erscheinungstag (10.2 Mio. zu 11.05 Mio).

Die Abbildung 7.6 zeigt den Verlauf der durschnittlichen Sentiment Scores auf Tagesbasis. Es ist ein Zick-Zack Trend zu erkennen, bei dem abwechselnd an jedem nächsten Tag der Durschnitt der Scores sinkt und am nächsten Tag wieder steigt. Die Verkäufe hingegen nehmen immer mehr ab.

[4]Diese Beobachtung unterstützt die These dieser Diplomarbeit, denn falls ein Zusammenhang zwischen Tweets und Filmerfolg aufgezeigt werden kann, würde dies auch gleichzeitig bedeuten, dass Tweets von solchen Usern wie dem Hauptdarsteller direkten Einfluss auf den Erfolg eines Films haben.

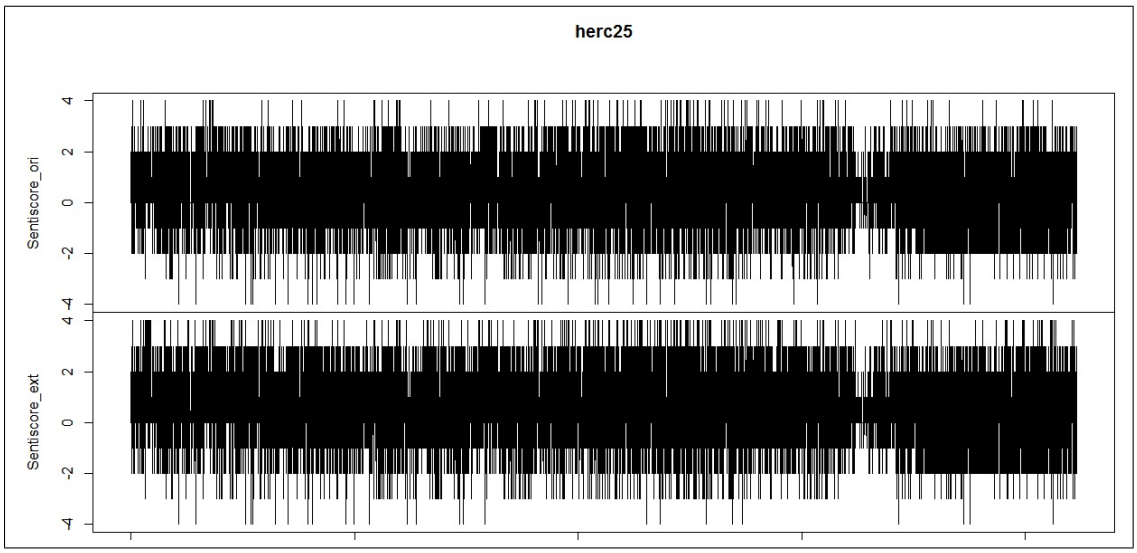


Abbildung 7.5: Hercules Intraday 25.07.2014

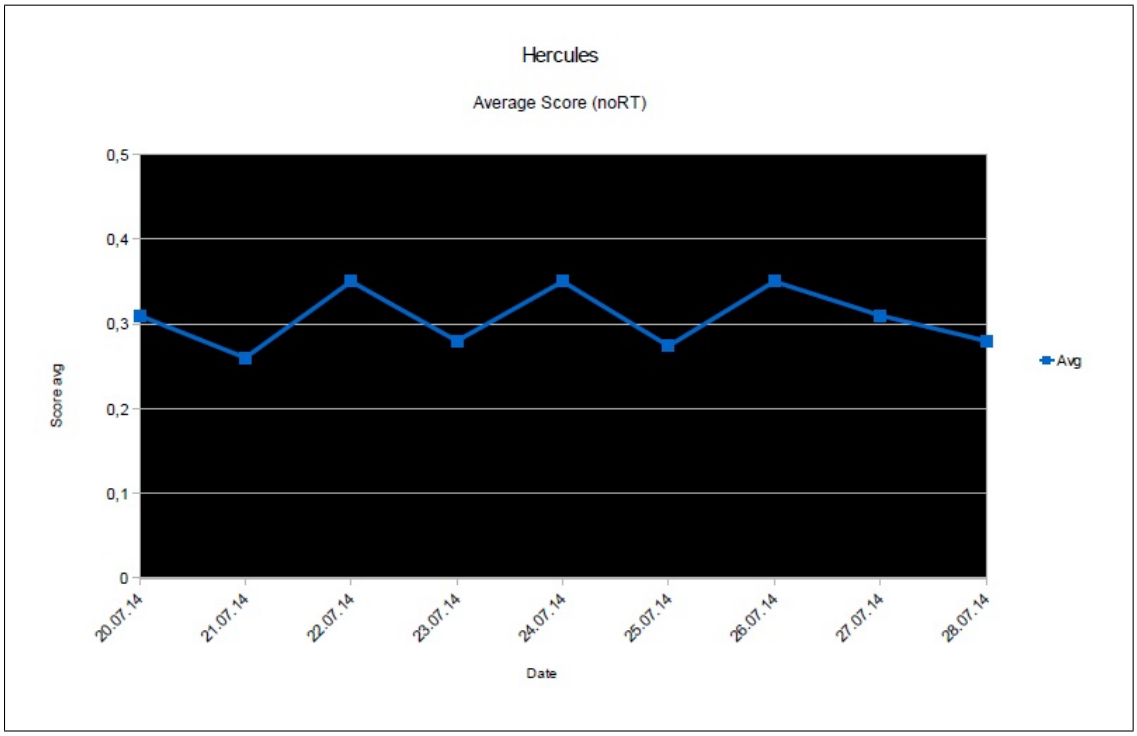


Abbildung 7.6: Hercules score average over time

In Abbildung 7.7 sind die Gesamtstatistiken des Herkules-Datensatzes zu sehen. Die *positive, negative, neutral* Linien beziehen sich auf die rechte zweite y-Achse und sind prozentuale Angaben. Auch hier ist ein Zick-Zack-Verlauf zu erkennen. Ein deutlicher peak in der Anzahl der Tweets ist am 25.07.2014 zu erkennen. Die höchste Anzahl an Tweets werden also am Erscheinungstag erstellt. Diese bestehen zum einen Teil aus Usern, die den Film gesehen haben und bewerten und zum anderen Teil aus Usern, die mitteilen, dass sie den Film sehen möchten. Es kann auch beobachtet werden, dass am 22.07. der Anteil an negativen und neutralen Tweets plötzlich sinkt und die positiven Tweets dominieren. Dies könnte auf die Twitter-Kampagne des Hauptdarstellers am Vortag zurückgeführt werden. Bei den Tagen nach Erscheinungsdatum sinken die Tweetraten genauso rasch, wie sie angestiegen sind. Zum selben Zeitpunkt sinkt auch die Anzahl an positiven Tweets.

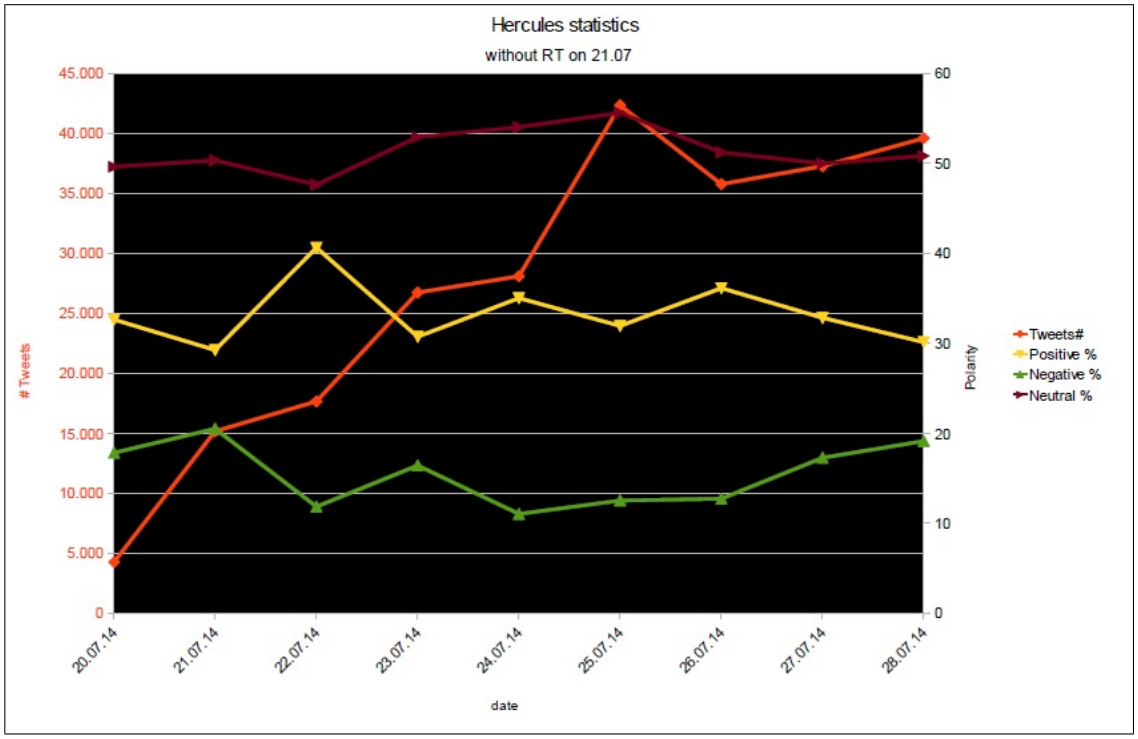


Abbildung 7.7: Hercules score average over time

Der erweiterte Wortschatz wird in 8.1 bewertet. Allgemein ist an allen Datensätzen mit dem extended lexicon zu beobachten, dass es deutlich mehr Tweets gibt, die mit einem Score von 4 versehen wurden und weniger Tweets mit -4 klassifiziert wurden. Diese können an den Aussschlägen bei den y-Werten gesehen werden.

Am 20.07.2014 in Abbildung 7.2 ist des Weiteren ein Ausreißer zu erkennen, der mit einer negativen Polarität von -4 klassifiziert wurde: *Worst movie ever! USER: Watching The legend of Hercules with the gents. The side comments are making my day :'D*

Inhaltlich ist zu sehen, dass es sich bei dieser Bewertung um einen anderen Film handelt, der am 10. Januar 2014 erschien. Die Polarität und Stärke wurde richtig erkannt, aber

in solchen Sonderfällen müsste eine separate Filterung nach ähnlichen Begriffen gemacht werden. Da es sich um einen Einzelfall handelt, wird dies hier ignoriert.

Die Verteilungen der Sentiment Scores können in Abbildung 7.8 und 7.9 entnommen werden. Die Histrogramme sind derart aufgebaut, dass der dunkelgrüne Bereich die Überschneidungen der Stimmungen bei Originallexicon und Extended Lexicon zeigt. Nicht überlappende Bereiche sind entweder rosa (Originallexicon) oder hellgrün (Extended Lexicon). Es zeigt sich, dass bei sämtlichen Filmen die neutralen Tweets gleich stark dominieren. Es fällt bei den neutralen Tweets auf, dass bei allen Filmen das erweiterte Lexicon weniger neutrale Tweets entdeckt hat als das Original. Gleichzeitig haben Filme nach der Stimmungsanalyse mit dem Extended Lexicon rechts von den neutralen Tweets in den meisten Filmen mehr positive Tweets als das Originallexicon. Dies könnte den Schluss zulassen, dass vom Originallexicon möglicherweise viele positive Tweets nicht erkannt und daher neutral klassifiziert wurden. Besonders auffällig ist dies bei dem Film Guardians of the Galaxy. Das Extended Lexicon hat in diesem Film deutlich mehr Tweets mit der Bewertung +3 erkannt (hellgrüner nicht überlappender Bereich rechts der Nullsäule).

Einen großen Unterschied zwischen den Stimmungen (Original und Extended) sieht man bei *The Purge: Anarchy*. Bei der Originalwortliste wurden deutlich mehr Tweets neutral klassifiziert als im Extended Lexicon. Da es sich bei diesem Titel um einen Film handelt, der viele Worte wie *scary, horror* u.a. enthalten kann, wird deutlich, dass die Originalvariante sehr viele Fehlklassifikationen gemacht hat. Die Erweiterung des Lexicons führte hier zu einer großen Verschiebung der Polarität zum Positiven. In der Analyse verdoppelte sich die PT-NT Ratio von ehemals $3,1$ auf $6,46$, was der IMDB Bewertung von 6.5 / 10 sehr nahe kommt. Dieser extreme Unterschied ist vor allem wegen des Genres dieses Films passiert. IMDB listet den Film als Horror/Thriller auf. Der Originalalgorithmus von SentiStrength hat also Probleme bei der Klassifikation von Filmen aus diesem Genre, weil domänenunabhängig als negativ empfundene Worte wie *scary, horror, shock, etc.* in dieser Domäne ein positives Attribut darstellen. Erweiterte Lexicons müssen also insbesondere in diesem Bereich viele Worte abdecken, um positive Tweets nicht fälschlicherweise als negativ zu deklarieren.

Filme wie *Hercules, Guardians of Galaxy* und *Dawn of the Planet of the Apes* weisen starke Ähnlichkeiten in den Verteilungen der Scores auf. Es gibt eine große Menge neutraler Tweets und zur rechten Seite hin (positive Tweets) wird eine Treppenform angenommen. Es gibt dann sehr viele positive Tweets, die trendförmig abnehmen. Bei weiniger großen Filmen wie *Most Wanted Man* gibt es diese Treppenform nicht. Auch ist der Gesamtdurschnitt der Scores niedriger als bei den Blockbuster Filmen (lediglich 1.48 durschnittlicher Score am Erscheinungstag mit Gesamttweets von 8544).

Dass bei allen Filmen auch bei dem erweiterten Lexicon immer noch sehr viele neutrale Tweets (wenn auch weniger) auftauchen, deutet darauf hin, dass es noch viel Potential bei der Erweiterung des Wortschatzes gibt. Es ist aber auch wahrscheinlich, dass trotz der Preprocessing Methoden immer noch ein sehr großer Anteil an irrelevanten Tweets vorhanden (Noise) ist. In einer zukünftigen Arbeit könnten weitere Verbesserungen bezüglich der Erkennung von Spam-Tweets implementiert werden, um die Anzahl der neu-

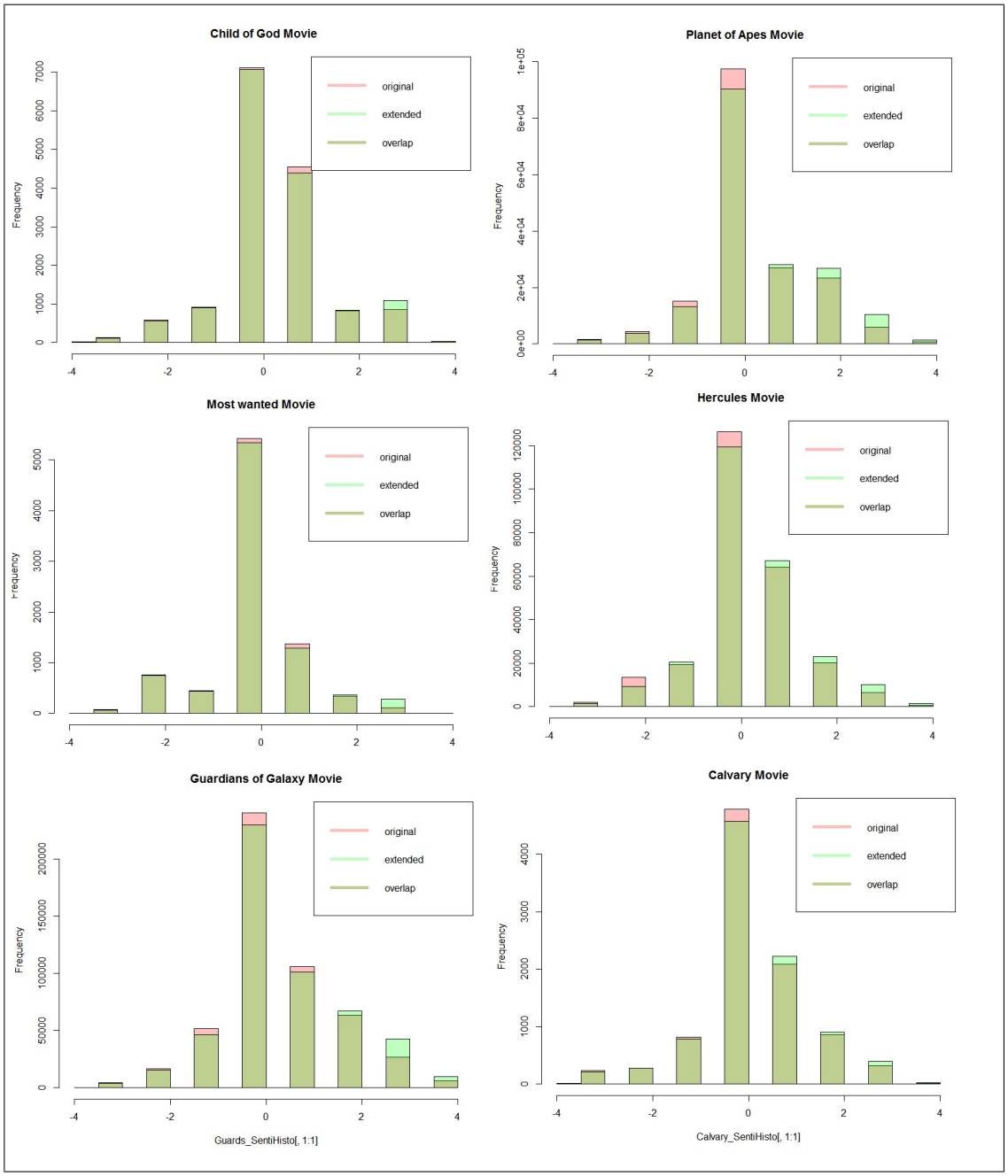


Abbildung 7.8: Verteilungen der Stimmungen mehrerer Filme

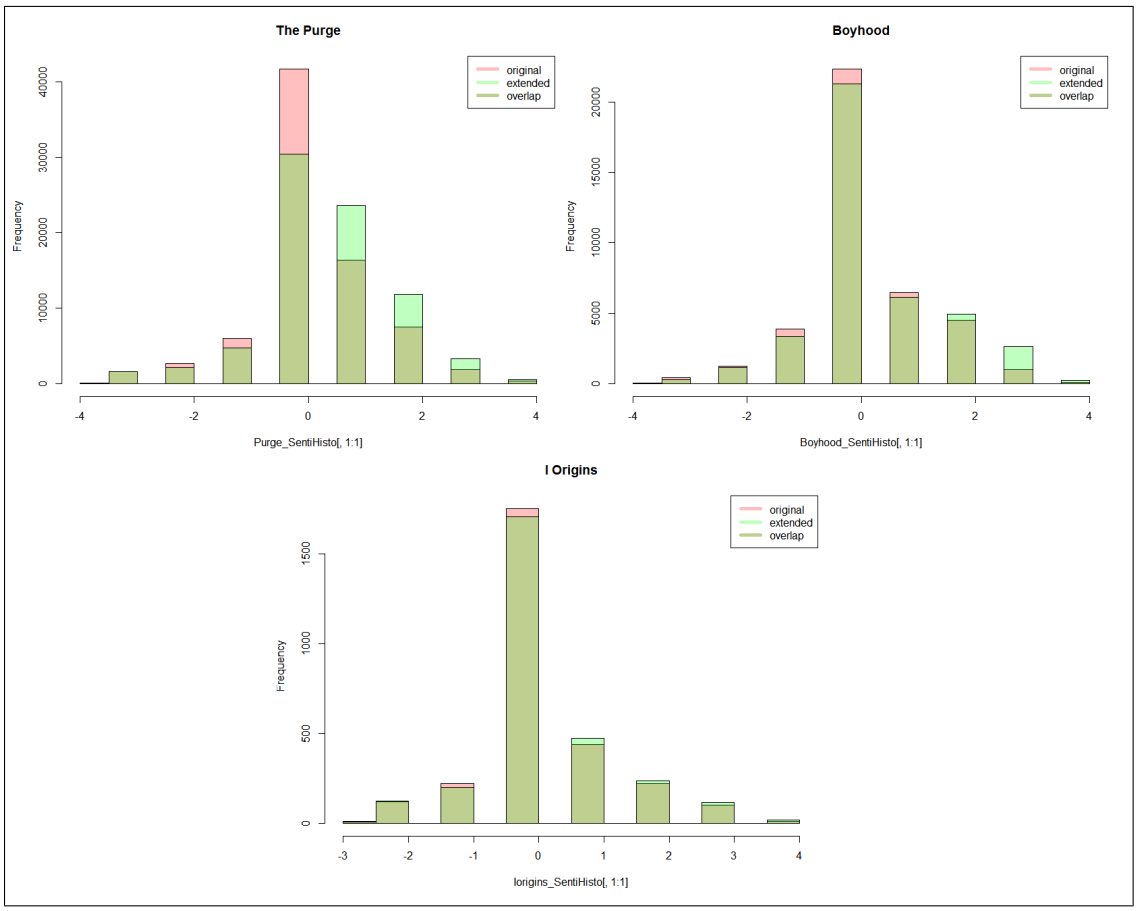


Abbildung 7.9: Verteilungen der Stimmungen mehrerer Filme

tralen Tweets zu minimieren. Insbesondere irrelevante Tweets nach Erscheinungstag, die illegale Streamlinks anbieten, überfluten Twitter bei den bekannteren Filmtiteln.

In der Übersicht 7.10 sind die Ergebnisse der Sentiment Analysis dargestellt. Jede Zeile stellt je einen Film dar, wobei die linken Teilbilder die Ergebnisse mit dem Originallexicon zeigen. Die schwarze Linie zeigt die Anzahl der Gesamttweets im Zeitverlauf. Die grüne Linie stellt die positiven, die rote Linie die negativen und die blaue Linie alle neutralen Tweets dar. Eine allgemein gültige Beobachtung für alle Filme ist, dass in der Stimmungsanalyse immer die neutralen Tweets überwiegen. Da in dieser Diplomarbeit nicht auf Retweets als Duplikate verzichtet wird[5], ist dies ein erwartetes Ergebnis.

Bei dem Film *Guardians of the Galaxy* (ersten beiden Bilder der Abbildung) gab es insgesamt 515.508 Tweets mit einem PT-NT Verhältnis von 3,11 (extended). Dies ist eine der höchsten festgestellten Werte unter allen beobachteten Filmen. Dabei ist der PT-NT Wert mit Originallexicon deutlich geringer und beträgt 2,60. Dies zeigt, wie gravierend die Ergebnisse bereits mit einer kleinen Lexiconerweiterung schwanken können. Es ist zu erwarten, dass bei einer größeren Wortliste der PT-NT Wert der extended Variante noch höher ausfällt. Bereits bei diesem Experiment sind fast die Hälfte aller Tweets zu diesem Film positive Tweets (220.326). Wegen des sehr großen Gefälles von den positiven Tweets zu den negativen (66.139), sind die negativen Tweets dieses Films in der Abbildung 7.10 in dieser Skalierung nicht mehr sichtbar. Die sehr starke positive Polarität spiegelt sich auch in den Verkäufen dieses Films wieder (siehe Kapitel 8). Der Film ist der erfolgreichste unter allen beobachteten Filmen und erzielte insgesamt 771 $ Millionen an Kinoverkäufen und 110 $ Millionen an DVD Verkäufen. Bei dem offiziellen Budget von 170 $ Millionen entspricht das mehr als dem fünffachen Betrag der Kosten. Auch die IMDB Bewertung von 8.1 / 10 ist sehr positiv und bei IMDB als Bewertung selten vorzufinden.

Die anderen beiden Filme zeigen ebenfalls einen Anstieg des PT-NT Wertes nach der Lexiconerweiterung, doch dieser ist nicht so stark wie bei Guardians. Der Film *I Origins*(3. Reihe) beispielsweise ist eine Low-Budget Produktion. In der Abbildung ist zu erkennen, dass das Maximum der Verkäufe schneller erreicht wird als bei den großen Filmen. Auch tritt die Rückgangsphase der Tweets sehr viel schneller ein. In den Experimenten konnte diese Beobachtung bei allen Low-Budget Filmen gemacht werden. Dies führte manchmal dazu, dass der Film die Produktionskosten nicht mit den Verkäufen decken konnte. Im Fall von I Origins jedoch wurden die geringen Produktionskosten von 100.000 $ mit einem Gesamtumsatz von ca. 400.000 $ (mit DVD Verkäufen) vierfach gedeckt. Mit einer hohen positiven Bewertung von 7.3 / 10 belohnten die Zuschauer den Film über IMDB, obwohl insgesamt nur sehr wenige Tweets zu dem Film erstellt wurden. Insgesamt gibt es in 7 Tagen nur 2882 Tweets mit lediglich 354 negativen Tweets. Bei diesem Film handelt es sich um eine Nischenproduktion mit kleiner Kundschaft, die aber über Twitter fast nur Positives berichten konnte. Im selben Zeitraum gab es über 500.000 Tweets für Guardians of the Galaxy. Wegen der sehr geringen Zahl der Tweets (7 Tage vor Release gab es 61 Tweets, davon 0 negativ), fällt der PT-NT Wert mit 4,71 bzw. 5.13 (extended)

[5]Diese Arbeit betrachtet Retweets als westentlichen Bestandteil von Twitter. Die Löschung der Retweets als „Duplikate“ würde zwangsläufig zu einer Verzerrung der tatsächlichen Analysegrundlage führen.

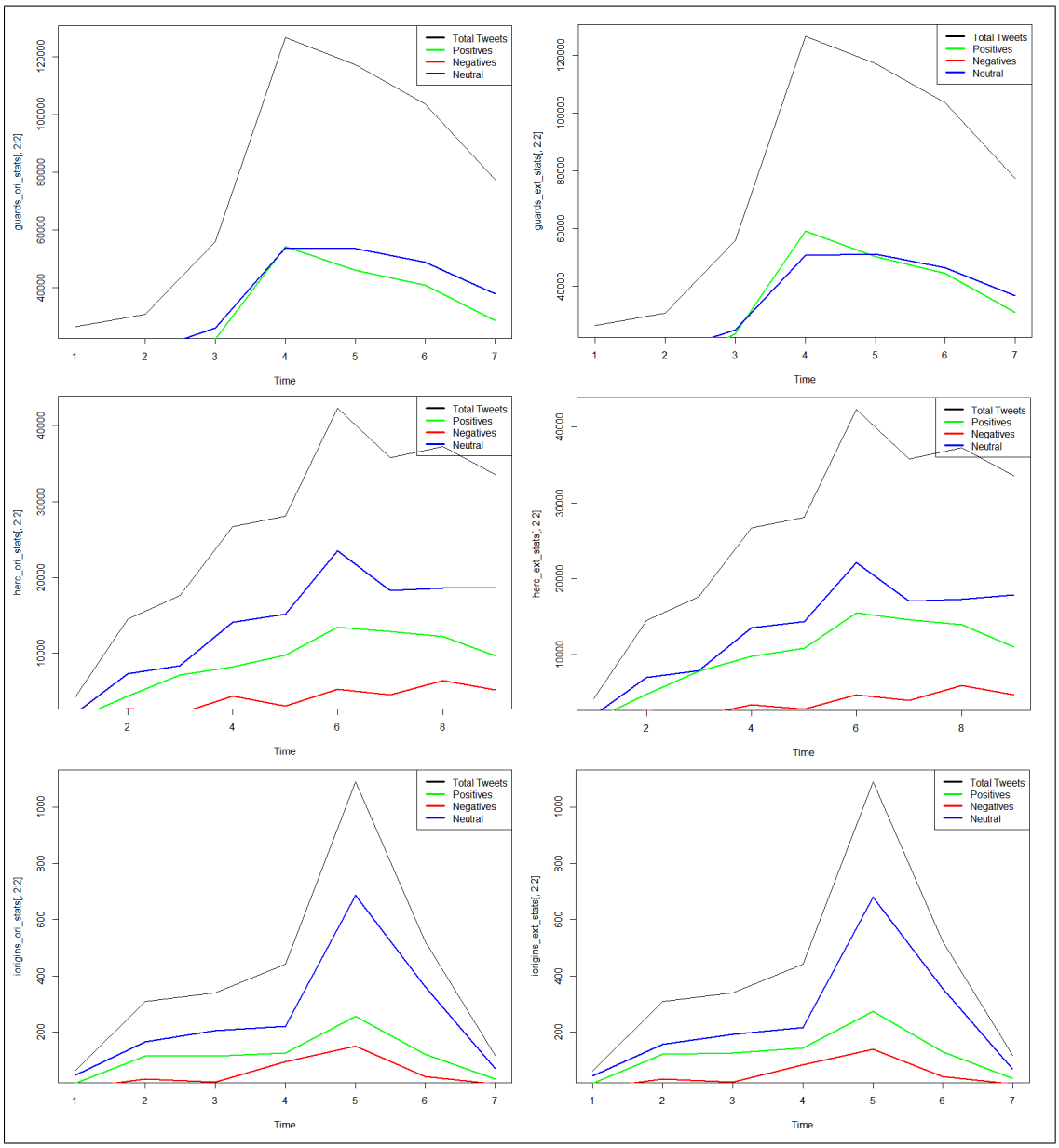


Abbildung 7.10: Linke Seite: Originallexicon Sentiments, Rechte Seite: Extended Lexicon Sentiments

sehr hoch aus. Dieser Film ist im Verhältnis zu der sehr kleinen Budgetsumme also extrem erfolgreich gewesen, was auch die PT-NT Werte der Stimmungsanalyse bestätigen konnten.

Die Abbildungen 7.11 und 7.12 zeigen die Tweetverläufe der restlichen Filme. Auffällig sind hier die Schwankungen der Tweetzahlen bei dem Film *A Most Wanted Man*. Als einziger Film unter der beobachteten Menge machten die neutralen Tweets fast die Gesamtmenge aller Tweets aus. Das Verhältnis der positiven und negativen Tweets ist sehr ausgeglichen (2.1). Der Film machte keine Verluste und kann mit einer IMDB Bewertung von 6.9 / 10 immer noch erfolgreich angesehen werden. Bei einem Budget von ca. 15 $ Millionen erreichte der Film ca. 31 $ Millionen Umsatz und brachte somit den doppelten Betrag der Kosten wieder ein. Ausgehend von den Tweetdaten und dem PT-NT Wert, hätte bei diesem Wert ein schlechterer Umsatz erwartet werden können, denn es gibt nur wenige positive Tweets. Tatsächlich könnte der unerwartete Erfolg diesen Films mit äußeren Umständen zusammenhängen, denn der bekannte Hauptdarsteller *Philip Seymour Hoffman* verstarb 02.02.2014 kurz vor Veröffentlichung des Films am 25.07.2014. Da es sich um eine berühmte Person handelt, kann davon ausgegangen werden, dass dies zu einer deutlichen Erhöhung der Besucherzahlen geführt hat, welche in den Tweets aber nicht sichtbar war. Auch die Analyse mit erweitertem Lexicon macht hier keinen Unterschied. Es muss im Kapitel 8 geprüft werden, inwieweit dieser Sonderfall die prädiktive Analyse mittels Regression und t-Test beeinträchtigt.

Bei den Filmen in Abbildung 7.11 ähneln sich die Verläufe der linken Seite sehr stark der rechten Seite (Extended Lexicon). Das bedeutet, dass die Erweiterung des Lexicons bei diesen 3 Filmen zu keiner großen Veränderung der PT-NT Werte geführt hat.

Es wird in den Experimenten auch auf einen möglichen Zusammenhang des Box Office mit der Anzahl der Tweets mit URL Verlinkungen geachtet. Zu diesem Zwecke wurden bereits im Preprocessing in Kapitel 5 alle Hyperlinks in den Tweets mittels regulären Ausdrücken herausgefiltert und mit Platzhaltern („URL“) ersetzt. Dieser Schritt der Vorverarbeitung kann nun derart verwendet werden, dass eine Aufstellung der täglichen Hyperlinks zu jedem Film auf Twitter erstellt werden kann. Ohne der Vorverarbeitung ist eine Zählung der URLs in den Tweets nicht machbar, da insgesamt ca. 2.000.000 Tweets extrahiert wurden und die Hyperlinks in verschiedenen Formen und Abkürzungsdiensten erscheinen[6]. Die Abbildung 7.13 zeigt den zeitlichen Verlauf der URLs. Die Extrema treten jeweils immer am Erscheinungstag und Folgetag auf.

Die Häufigkeiten der URLs basieren bei jedem Film in der Abbildung 7.13 auf einer anderen Tweetskala und zeigen den generellen Trend. Da die Anzahl der URLs mit der Anzahl der Gesamttweets steigt, weisen die beiden Prädiktorvariablen Tweetrate und #URLs bei allen Titeln eine sehr hohe Korrelation auf und werden daher im Kapitel *Evaluation und Success Prediction* (8) in keiner der Regressionsmodelle im selben Modell verwendet. Beispielsweise beträgt die Korrelation beider Variablen in *Guardians of the Galaxy* (Extended Lexicon) $R^2 = 0.95$ und bei Hercules (Extended Lexicon) $R^2 = 0.76$. Diese Eigenschaften müssen bei der Modellierung beachtet werden, da Multikollinearitäten (wie in Kapitel 6.2.2) erwähnt die Aussagekraft des Modells schmälern können.

[6] Viele URLs werden mit Diensten wie tinyurl abgekürzt.

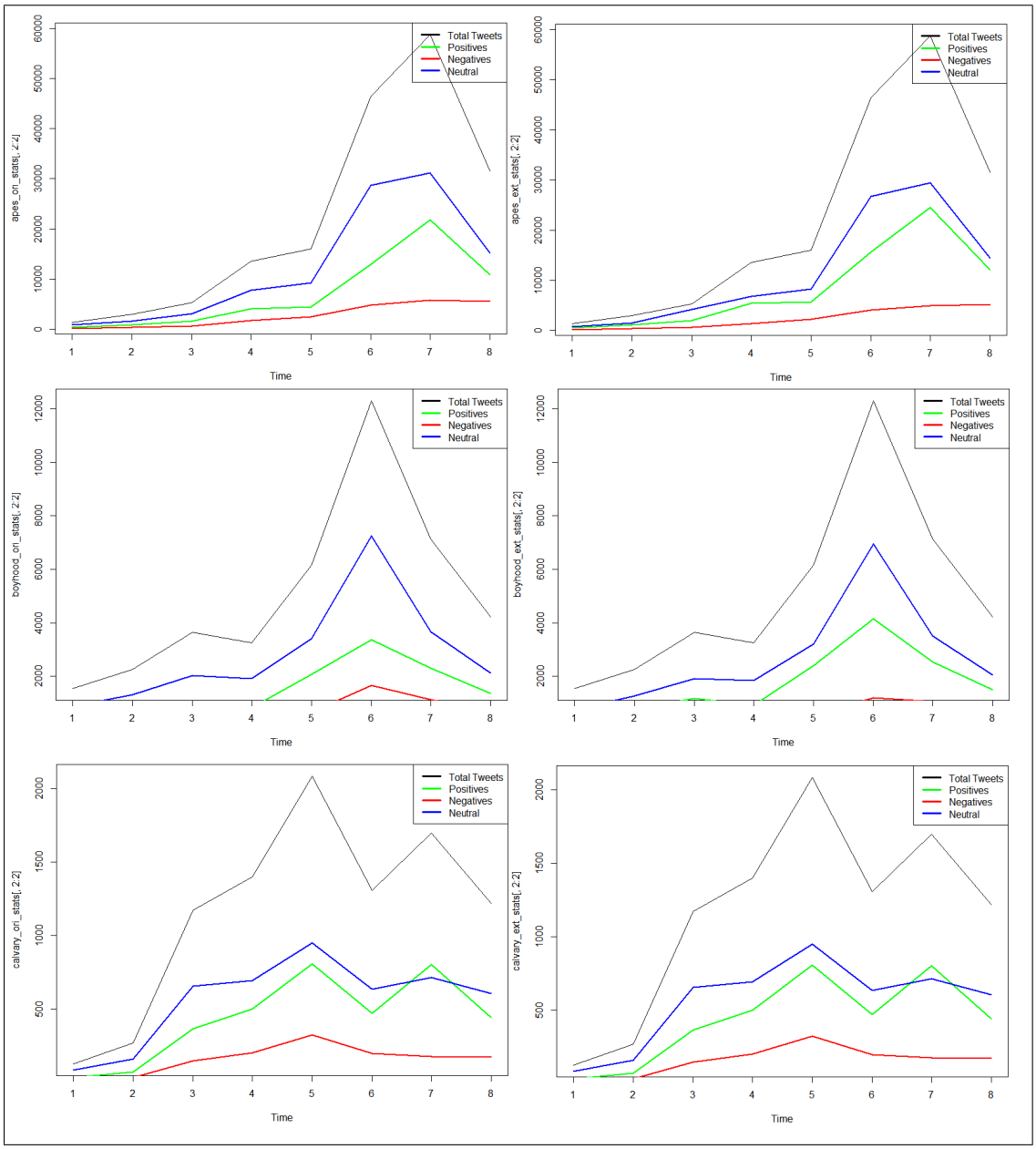


Abbildung 7.11: Linke Seite: Originallexicon Sentiments, Rechte Seite: Extended Lexicon Sentiments

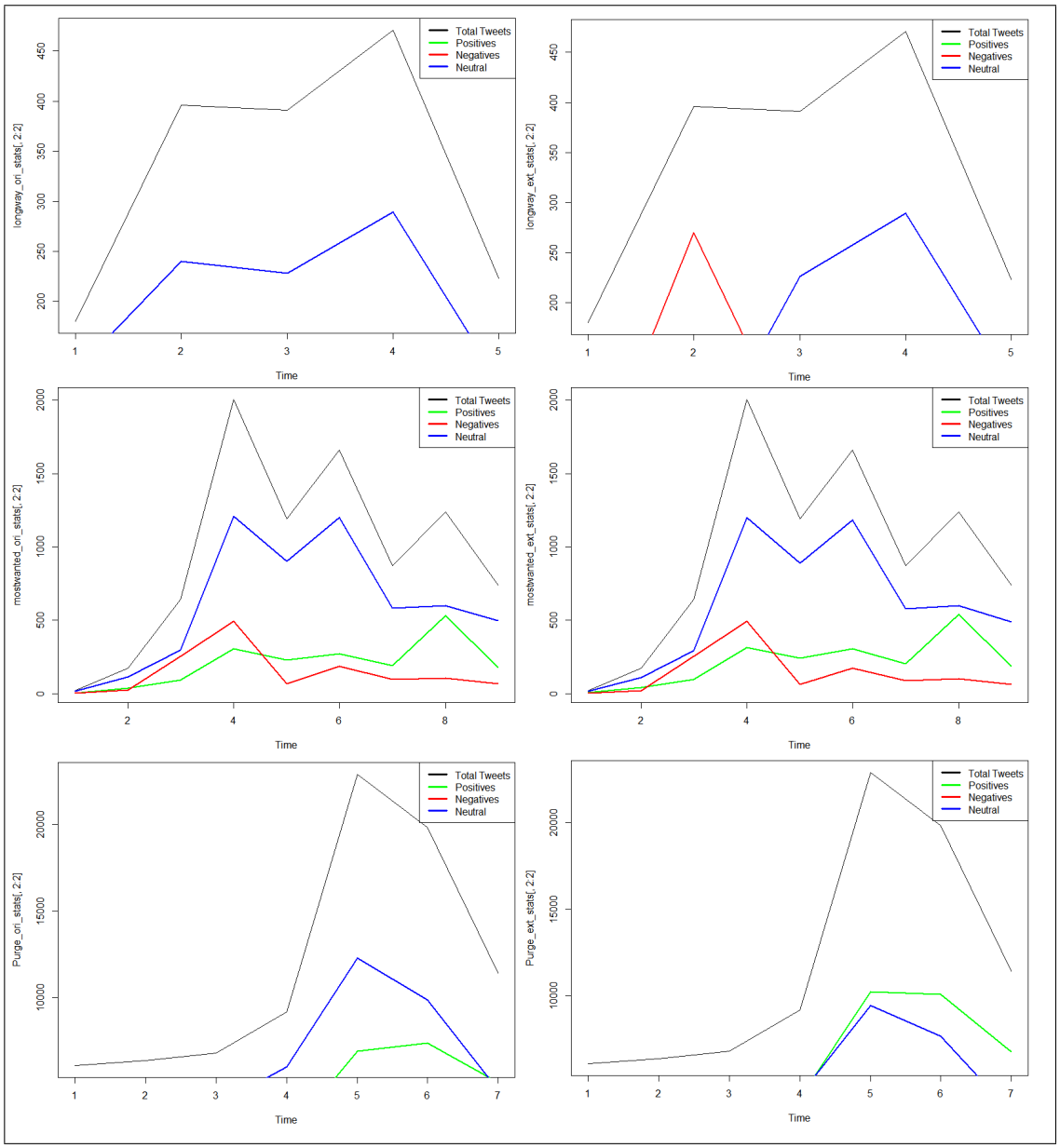


Abbildung 7.12: Linke Seite: Originallexicon Sentiments, Rechte Seite: Extended Lexicon Sentiments

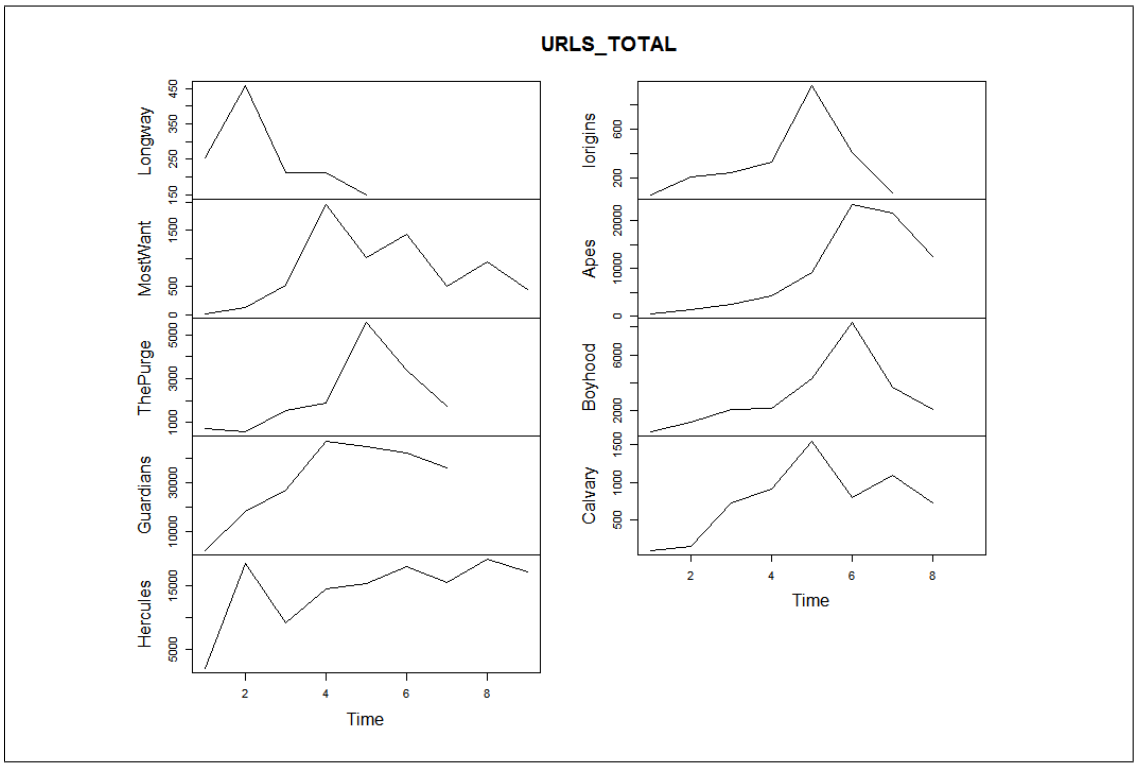


Abbildung 7.13: Zeitlicher Verlauf der URLs

Insgesamt ist wegen der hohen Korrelation zur Tweetrate und der Verläufe in Abbildung 7.13 zu erwarten, dass der Prädiktor URL ein ähnlich starkes Gewicht bei der Vorhersage im Regressionsmodell hat. Diese Annahme konnte in den Vorhersagen im nächsten Kapitel bestätigt werden.

7.2 Hybrides Experiment: SVM und SentiStrength Kombination

Support Vector Maschinen erwarten eine große Menge an Daten für das Training. Ohne genügend Trainingsdaten, können SVMs nicht die geforderten Ergebnisse liefern. SentiStrength kann schneller als maschinelle oder menschliche Verfahren automatisch eine große Menge an Daten klassifizieren. Es stellt sich die Frage, ob ein Trainingsdatensatz mittels eines Lexiconverfahrens wie SentiStrength dazu verwendet werden könnte, um einen günstigen und ausreichend großen Trainingsdatensatz zu erstellen. Falls dies möglich wäre, könnten APIs wie SentiStrength in Wrapper gepackt und zur Erstellung von Trainingsdatensätzen als Operatoren in RapidMiner eingebunden werden. Dadurch wäre es im Data Mining bzw. Text Mining sehr viel leichter an gelabelte Daten zu kommen.

Für das Training der SVM werden zufällig 13.000 Tweets aus mehreren Datensätzen entnommen. Da die SVM wegen der Trainingsphase eine längere Laufzeit hat, müssen

irrelevante oder doppelte Tweets entfernt werden. Vor Allem Spam-Tweets können der SVM Schwierigkeiten bereiten, da völlig sinnfreie Merkmale und Zeichenaneinanderreihungen den Merkmalsvektor verlängern können. Für die Stimmungsanalyse sind Duplikate erlaubt. Bei der SVM würden sie jedoch keinen Beitrag leisten. Allgemein ist das Text Mining mit SVM sehr zeitaufwendig. Methoden zur Dimensionsreduktion werden in dieser Arbeit einfach gehalten, da beispielsweise PCA für einen Datensatz bereits zu einer mehrtägigen Laufzeit geführt hat und dies in dieser Diplomarbeit nicht praktikabel ist.

Der Trainingsdatensatz von 13.000 Tweets führt bei einigen Filmexperimenten zu einer sehr hohen Zahl an neutralen und negativen Tweets. Obwohl der Trainingsdatensatz mittels SentiStrength erstellt wurde, ergeben sehr große Unterschiede in den Ergebnissen. Es ist zu erwähnen, dass hier mit SVM 3 Klassen betrachtet werden, wohingegen SentiStrength eine Skala von -4 bis $+4$ berechnet.

accuracy: 83.75% +/- 0.42% (mikro: 83.75%)

	true positiv	true negativ	true neutral	class precision
pred. positiv	3835	177	383	87.26%
pred. negativ	195	4105	334	88.58%
pred. neutral	671	418	3283	75.09%
class recall	81.58%	87.34%	82.08%	

Abbildung 7.14: SVM Konfusionsmatrix: Kreuzvalidierung

Die Abbildung 7.14 zeigt die Performanzwerte der SVM Experimente mit *unigram*. Es handelt sich hierbei um eine 5-fach Kreuzvalidierung in RapidMiner LibSVM. Das Ergebnis wird in einer Konfusionsmatrix gezeigt. Eine generelle Definition und Beschreibung der Konfusionsmatrix wird im Kapitel 6.2 in Tabelle 6.2 gezeigt. Im Unterschied zu der in der Literatur üblichen 2-Klassen SVM und 2-Klassen Konfusionsmatrix, zeigt Abbildung 7.14 einen Performanzvektor mit 3 Klassen an. In dieser Diplomarbeit wird zusätzlich die neutrale Klasse zur SVM Stimmungsanalyse mit eingebracht und aufgeführt.

accuracy: 82.55% +/- 0.82% (mikro: 82.55%)

	true positiv	true negativ	true neutral	class precision
pred. positiv	3837	185	288	89.03%
pred. negativ	333	4101	588	81.66%
pred. neutral	531	414	3124	76.78%
class recall	81.62%	87.26%	78.10%	

Abbildung 7.15: SVM Konfusionsmatrix: bigram

In Abbildung 7.15 sind die Ergebnisse für das SVM Modell mit bigrams dargestellt. Es fällt auf, dass der Class Recall der neutralen Klasse von 82.08% auf 78.10% gefallen ist. Auch die Accuracy fällt leicht zurück. Es wurden mehrere Durchläufe mit 3-gramm und 4-gramm getestet und festgestellt, dass die Performanz am besten mit unigrams ist. Da dies das Ergebniss von 5-fach Kreuzvalidierung über dem selben Trainingsdatensatz ist, kann aber nicht ausgesagt werden, ob unigrams auch für das Testset auf selbe Weise resultieren. Daher wird mit dem unigram SVM Model fortgefahren.

Die SVM Stimmungsanalyse mit Trainingsdaten von SentiStrength lieferte im Vergleich zur reinen SentiStrength Methode sehr große Unterschiede. Insbesondere die Anzahl der negativen und neutralen Tweets war sehr viel höher, was zu einer niedrigen PT-NT Ratio geführt hat. Da die Kreuzvalidierung und Performanzanalyse des SVM Modells hohe Accuracy-Werte von 83 % lieferte, liegt eventuell ein Overfitting Problem vor. Das Modell passt sich möglicherweise zu stark an das Trainingsmodell an und generalisiert schlecht. Diese Annahme beruht auf den teilweise sehr unterschiedlichen Ergebnisse der SVM Experimente. Die Tabelle 7.1 zeigt die Ergebnisse im Vergleich. Es wurden 6 der 9 Filme mit dem SVM Modell untersucht[7].

	SVM	Lexicon PTNT (orig.)	Lexicon PTNT (ext.)
Boyhood	0,56	2,53	3,36
Dawn of the Planet of Apes	0,61	2,49	3,19
Guardians of Galaxy	1,07	2,6	3,11
Hercules	1,81	2,34	2,9
A Most Wanted Man	1,89	2,15	2,69
The Purge:Anarchy	4,05	3,1	6,46
I Origins	n.a.	4,71	5,13
Calvary	n.a.	2,66	2,99
Child of God	n.a.	4,48	4,68
A Long Way	n.a.	2,72	2,92

Tabelle 7.1: Vergleich: SVM und Lexicon

Insgesamt fallen die SVM Klassifikationen deutlich negativer aus als die Lexicon-Varianten. Selbst sehr erfolgreiche Filme wie *Boyhood* erzielen nur einen PT-NT Wert von 0,56 im Vergleich zu 3,36 beim Extended Lexicon. Da der Film auf dem IMDB Portal mit 8,1 sehr hoch bewertet wurde und auch insgesamt sehr hohe Einnahmen erzielen konnte, kann die Schlussfolgerung gemacht werden, dass das SVM Modell verbessert werden muss.

#tweets	tweetrate	#pos	#neg	#neutral	PT-NT ratio	#URLs
12.711	529,63	1.018	2016	9.677	0,50	1866
14.511	604,63	2.166	2519	9.826	0,86	18489
17.636	734,83	1.996	3634	12.006	0,55	9052
26.713	1113,04	19.847	3658	3.208	5,43	14518
28.073	1169,71	13.367	5899	8.807	2,27	15284
42.313	1763,04	13.230	16496	12.587	0,80	18043
35.753	1489,71	7.489	10813	17.451	0,69	15448
37.246	1551,92	3.725	27688	5.833	0,13	19170
33.591	1399,63	25.508	5005	3.078	5,10	17174

Tabelle 7.2: Auszug aus Hercules: SVM Ergebnisse

Tabelle 7.2 zeigt einen Auszug aus der Ergebnistabelle der SVM Experimente für den Film Hercules. Die PT-NT Werte schwanken sehr stark und haben einen Durchschnitt von 1,81. Obwohl es an einzelnen Tagen auch viele positive Tweets gibt, wurden generell viele Tweets negativ oder neutral klassifiziert. Dadurch erhält der Film im SVM Modell eine niedrige Bewertung.

[7] Einige Filme boten zu wenige Tweetdaten für das SVM Modell und wurden außer Acht gelassen.

Es könnten noch Parameter optimiert werden, doch das Problem liegt wahrscheinlich an der Trainingsdatenmenge. Für diese Arbeit wurden mehrere Trainingsdatensätze für das SVM Modell erstellt und getestet (13.000 Tweet Sample, 6.000 Tweet Sample, 250 Tweet Sample). Generell scheint das Modell noch mehr Trainingsdaten zu brauchen. Das weitere Anpassen und Optimieren der modellierten SVM kann in einer zukünftigen Arbeit thematisiert werden. Vor allem könnten gemischte Trainingsdaten von SentiStrength und kostenpflichtige Trainingsdaten von Amazon Mechanical Turk vermischt werden, um bessere Ergebnisse zu liefern. Des Weiteren könnten die Optimierungsoperatoren in RapidMiner zur weiteren Verbesserung des Modells beitragen. Da die RapidMiner Prozesse in dieser Arbeit mit sehr großen Datenmengen umgehen mussten, sollte das Zielsystem gut ausgestattet sein. In dieser Diplomarbeit wurden mit begrenzten Ressourcen mehrere Gigabyte an Textdaten (ca. 2.000.000 Tweets) untersucht und klassifiziert. Der Hierarchiebaum des erstellten Modells ist in Abbildung 7.16 dargestellt und wurde für die erwähnten Filme verwendet.

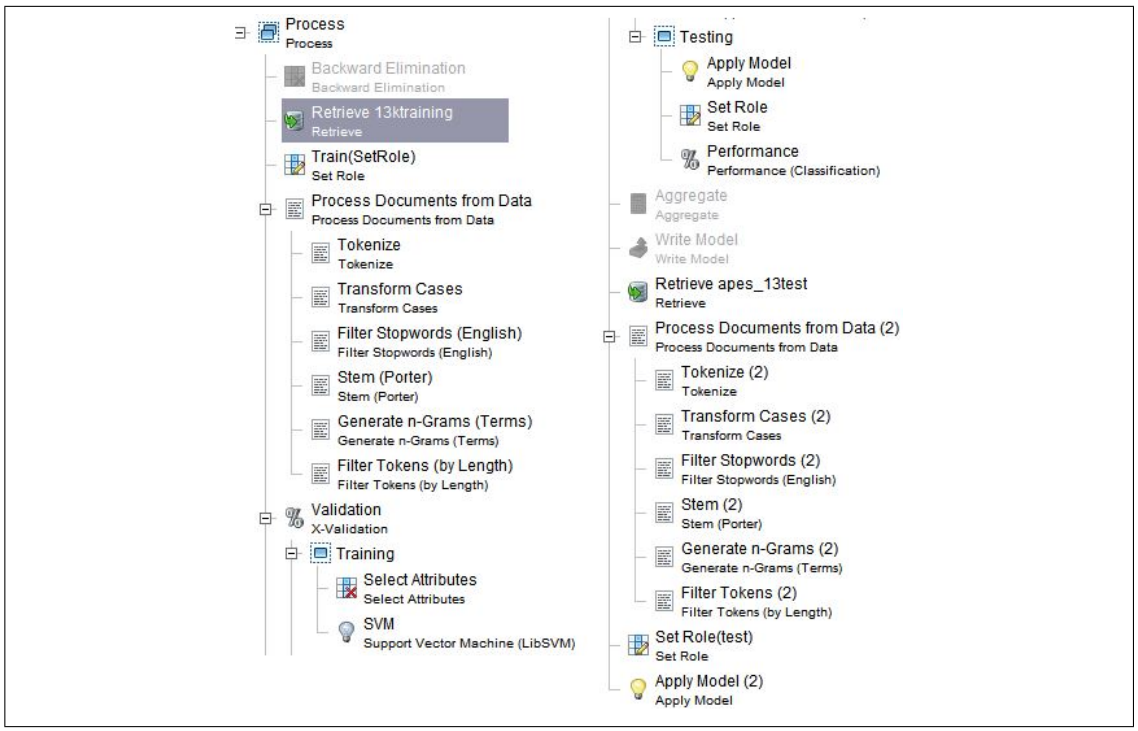


Abbildung 7.16: SVM Model Tree

Im nächsten Kapitel werden die Ergebnisse beider Ansätze (Lexicon und SVM) zur weiteren Verwendung in ein Regressionsmodell eingegeben, um eine Vorhersage der Verkäufe machen zu können.

8 Evaluierung / Success Prediction

Dieses Kapitel verwendet die Ergebnisse des Kapitels *Experimente* 7, um aus diesen Prädiktoren abzuleiten und ein Regressionsmodell aufzubauen.

8.1 Validierung des Extended Lexicon

Das erweiterte Lexicon muss einem Test unterworfen werden. Dadurch können Vergleiche über die Performanz mit dem Originallexicon gemacht werden. Dabei wird der Originalalgorithmus auf einen neuen Datensatz angewandt. Da es keine externe Sentiment Liste gibt, die als dritter *Baseline* Vergleichswert genommen werden kann, wird für diesen Zweck ein eigener manuell erstellter Datensatz als Vergleichswert genommen. Hierbei handelt es sich um 254 manuell gelabelte Tweet-Samples[1]. Der manuell gelabelte Vergleichs-Datensatz wird mit *Herc_base* benannt. Zur Vereinfachung wird in der Validierung nur auf die Polarität der Tweets geachtet, nicht aber auf die Stärke der Polarität. Der Grund für diese Entscheidung ist, dass der Fokus bei diesem Test auf die Polaritätsklassifikation von Tweets gelegt wird. Es interessiert also weniger ob ein Tweet +2 oder +1 erhält. Wichtiger ist es zu sehen, ob insgesamt positive Tweets nun mit der neuen Wortliste negativ klassifiziert werden und umgekehrt. Der von dem Originallexicon erstellte Datensatz heißt Herc_orig. Das Ergebnis des erweiterten Lexiconansatzes wird mit Herc_ext benannt. Die Tabelle 8.1 stellt die berechneten Scores der einzelnen Testdatensätze dar.

Abbildung 8.1 zeigt, dass das Originallexicon 37 % positive Tweets[2] klassifiziert hat. Über 20 % negative Tweets wurden im Evaluierungs-Datensatz gefunden. Im Vergleich dazu gibt es insgesamt tatsächlich 47,63 % positive Tweets. Das bedeutet, das Lexicon

[1] Zufällige Stichproben aus verschiedenen Datensätzen, die bisher nicht gesichtet wurden.
[2] True Positives

Tabelle 8.1: Scores der drei Herc-Datensätze

Datensatz	positive	negative	neutral	average score
Herc_ol	37 %	20,08 %	42,92 %	0,27
Herc_ext	44,5 %	16,14 %	39,36 %	0,50
Herc_base	47,63 %	7,87 %	44,5 %	0,84

hat sich bei über 10 % geirrt und fälschlicherweise als negative oder positive deklariert. Lediglich der Anteil an neutralen Tweets kommt der tatsächlichen Anzahl sehr nah.

Der erweiterte Wortschatz und der dazugehörige Datensatz Herc_ext zeigt bei den positiven Tweets eine hohe Übereinstimmung von 44,5 %. Die Klassifikation der neutralen und negativen Tweets hingegen ist schwächer. Auch wenn das neue Lexicon mit 16,14 % deutlich näher an der Baseline liegt, besteht immer noch ein Unterschied. Der durschnittliche Score ist mit 0,50 besser geschätzt worden, als in der Variante Herc_ol.

Insgesamt gibt es in Herc_ol 35,43 % *False positives bzw. negatives*. Das führt zu einer Class-Accuracy von 64,57 %. Der Durschnittsscore stimmt zu 32,14% mit der Baseline überein.

Der Datensatz Herc_ext beeinhaltet 28,7 % False positives bzw. negatives. Dadurch wird eine Accuracy on 71,3 % erreicht, was zu einer Verbesserung von fast 7 % im Vergleich zur Ursprungsvariante darstellt. Auch der durschnittliche Score liegt viel näher an der Baseline: 59,52 %.

Es lässt sich also schließen, dass die Erweiterung des Filmdomänen-Lexicons eine spürbare Verbesserung der Klassifizierungsgenauigkeit gebracht hat. Es bleibt nun im nächsten Kapitel zu überprüfen, ob und wie stark das zu einer erhöhten Vorhersagbarkeit mit dem Box Office führt.

8.2 Vorhersage mittels ausgewählter Prädiktoren

Es können in der Filmdomäne bereits ohne Berechnung einige vorausgehende Aussagen über den Verlauf und die mögliche gegenseitige Beeinflussung der Variablen gemacht werden. So ist es beispielsweise zu erwarten und zu zeigen, dass stärkere Sentiments eher nach dem Veröffentlichungsdatum auftreten, oder dass die Tweetrate kurz vor und nach Veröffentlichung rapide steigt. Der Grund dafür ist, dass Menschen auf Twitter eher dazu tendieren bereits gemachte Erfahrungen zu teilen. Der Grund für die erhöhte Anzahl kurz vor Erscheinen ist, dass in den Datensätzen viele Tweets festgestellt wurden, welche die Absicht den Film zu sehen wiedergaben. Darunter sind nicht nur neutrale Bekundungen, sondern auch antizipative positive Stimmungen. Insgesamt ist aufgefallen, dass über Filme auf Twitter generell mehr positiv als negativ getwittert wird.

Wichtig ist bei der Korrelation im Allgemeinen, dass eine nachgewiesene Korrelation nicht zwangsläufig eine Aussage über die Kausalität der beiden Variablen macht. Das bedeutet, dass die Korrelation auch Zufall sein könnte oder dass die Korrelation noch von einer dritten Variable abhängt. Die Prädiktoren für alle Filme entsprechen denen der Abbildung 8.1. Eine in bisherigen verwandten Arbeiten nicht verwendete Prädiktorvariable ist die Anzahl der URLs. Typischerweise enthalten viele Tweets mit Previewinformationen oder Meinungen zu Filmen Links zu Trailern, Interviews und Artikeln. Es kann mit dieser Variablen überprüft werden, ob ein linearer Zusammenhang zum Box Office besteht.

date	sales	#tweets	tweetrate	#pos	#neg	#neutral	sum(tot.score)
-5	-		0			0	
-4	-	996	42	372	121	503	464
-3	-	3310	138	1151	500	1659	1245
-2	-	30704	1279	10101	3675	16928	11320
-1	-	55953	2331	23640	7358	24955	30362
0	37.845.336	126666	5278	59160	16795	50711	86322
1	30.989.857	117285	4887	50322	15699	51264	70953
2	25.485.690	103560	4315	44569	12507	46484	65547
3	11.722.356	77334	3222	31011	9484	36839	45372
4	11.907.821						
5	8.808.382						
6	7.631.397						
7	12.312.683						

sum(pos.score)	sum(neg.score)	Tot.score avg	pos avg	neg avg	PT-NT ratio	#URLs
627	-163	0,466	1,013	-1,347	3,07	600
1897	-652	0,376	0,813	-1,304	2,30	2168
16248	-4928	0,369	0,830	-1,341	2,75	17913
40258	-9896	0,543	1,036	-1,345	3,21	26747
110289	-23967	0,681	1,090	-1,427	3,52	46918
92215	-21262	0,605	1,064	-1,354	3,21	44372
81914	-16367	0,633	1,129	-1,309	3,56	42051
58288	-12916	0,587	1,110	-1,362	3,27	35721
					avg	
		0,47	0,90	-1,20	2,77	

Abbildung 8.1: Guardians of the Galaxy Statistiken

Die Vorhersage wird derart modelliert, dass die a priori Informationen der Variablen, die Verkäufe ab dem Erscheinungswochenende vorhersagen wollen. In einem idealen linearen Modell mit unkorrelierten unabhängigen Variablen, die stark mit der abhängigen Variable korrelieren, würde die Vorhersagekraft des Modells mit der Anzahl der nicht-multikollinearen Regressoren steigen. In diversen Tests wurden in dieser Arbeit verschiedene Prädiktoren ermittelt und es wurden umfangreiche Statistiken für jeden Film erstellt. Die im Kapitel *Multikollinearität* 6.2.2 erwähnten Maßnahmen zur Auffindung von etwaigen Multikollinearitäten wurden auf sämtliche Prädiktoren angewendet. Es stellten sich einige Variablen als besonders multikollinear heraus, die nicht in das Regressionsmodell übernommen werden.

Zur Feststellung der statistischen Signifikanz wird der *t-Test* verwendet. Das lineare Regressionsmodell basiert auf der multiple linearen Regression. Der Signifikanztest liefert p-Values bei einem Signifikanzniveau von 0.05. Es soll untersucht werden, ob es einen Zusammenhang zwischen Stimmungsanalyse und Box Office gibt. Dazu werden zur Vorhersage ausgewählte Prädiktoren verwendet.

Die signifikanteste Variable war in allen t-Tests die **Tweetrate** zu einem Film. Zum Film Hercules ergeben sich beispielsweise mit den beiden Prädiktoren Tweetrate und negativer Gesamtscore des Tages folgende Ergebnisse:

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    10.9775606  1.0969288  10.008 5.77e-05 ***
```

```
tweetrate      -0.0089397  0.0020762  -4.306  0.00506 **
sum.neg.score.  0.0007078  0.0005051   1.401  0.21069
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.253 on 6 degrees of freedom
Multiple R-squared:  0.9024,Adjusted R-squared:  0.8698
F-statistic: 27.72 on 2 and 6 DF,  p-value: 0.0009311
```

Die Tweetrate zeigt eine hohe Signifikanz des p-Value von 0.00506 (** entspricht <= 0.01) über dem Niveau 5%. Das korrigierte Bestimmtheitsmaß R^2 (*Adjusted R-squared*) kann zur Evaluation verwendet werden[8] und ist mit 0.8698 sehr hoch. Diese Ergebnisse zeigen, dass vor allem die Tweetrate für eine akkurate Vorhersage verwendet werden kann, denn bei den Prädiktoren handelt es sich um Variablenwerte, die bereits **vor** der Filmausstrahlung und dem Box Office vorliegen. Das Modell liefert für die konkreten Vorhersagewerte der nächsten 9 Tage folgende Werte:

```
   Day 1        Day 2       Day 3      Day 4
10.05467405  8.63511281  6.42254223  4.50006001
Day 5           Day 6      Day 7        Day 8       Day 9
3.27559434  0.04689545  1.64369191  2.52915328  3.08672092
```

Die vorhergesagten Werte approximieren die tatsächlichen Werte gut und bilden eine Basis, um weitere Variablen in die Regression einzubringen und zu testen. Die beiden Verläufe der Vorhersagen und tatsächlichen Werte sind in Abbildung 8.2 dargestellt.

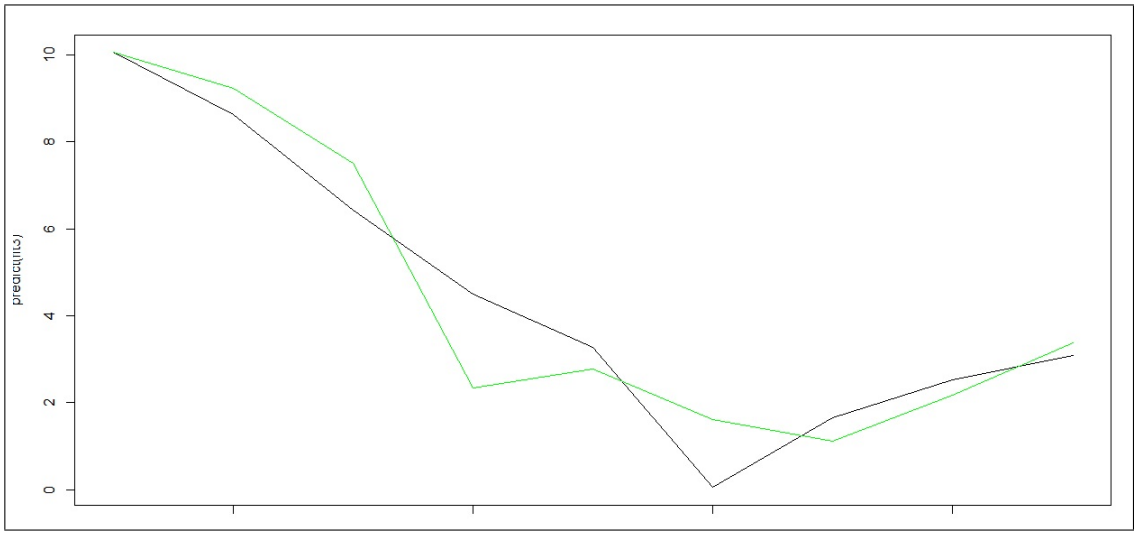


Abbildung 8.2: Green= Prediction, Black= Box Office

Die Steigungen der beiden Linien ähnelt sich und auch der tiefe Knick am 6. Tag der Filmausstrahlung findet in der Vorhersage eine Entsprechung. Der Knick in der Vorhersage tritt etwas früher auf.

Es wird nun versucht ein allgemines Modell und ein Feature Subset zu erstellen, so dass für jeden Film ein hohes korrigiertes R^2 (adjusted R^2) und niedrige p-Values erhalten werden. Die Signifikanzcodes „*“,“**“ und „***“ stehen jeweils für Signifikanz bei Niveau 0.05, 0.01 und 0.001. Das heißt der Threshold zur Annahme einer Variablen liegt bei maximal 0.05. Zur Auffindung eines geeigneten Subsets wird Forward Selection 1, Backward Elimination verwendet. AIC wird in mehreren Iterationen durchlaufen und wählt eine geeignete Submenge der Prädiktoren aus. Die Eignung der Prädiktoren wird anhand des adjusted R^2 und der p-Values bestimmt.

Mittels Forward Selection / Backward Elimination haben sich 3 verschiedene Subsets als allgemein beste Auswahl herausgestellt. Vor der Auswahl wurden Multikollinearitäten mittels Korrelationsmatrix, Variance Inflation Factor und Tolerance(TOL) beachtet[3]. Das erste Subset besteht aus dem einzelnen Prädiktor *Tweetrate*. Dieser korreliert zur Anzahl der Tweets und schließt daher diese Variable aus. Tatsächlich stellte sich heraus, dass alleine die Tweetrate bereits eine sehr hohe Aussagekraft hat. Beim Titel Guardians of the Galaxy ergibt sich beispielsweise bei diesem ersten Subset ein adjusted R^2 von 0.7 und ein p-Value von 0.01110 *, was eine Signifikanz bei 0.01 aufweist. Diese Werte änderten sich beim erweiterten Lexicon nicht. Beim Film Hercules beträgt der adjusted R^2 0.852 mit p-Value 0.00024 ***, was eine sehr starke Vorhersagekraft dieses Prädiktors andeutet.

Das zweite Subset wurde aus den Prädiktoren #URLs, PT-NT Ratio und Total Score Average gebildet. Einzeln betrachtet lieferten diese Features kaum signifikante Ergebnisse. Doch mittels schrittweiser AIC Anwendung wurde diese geeignete kombinierte Untermenge an Prädiktoren gefunden, die bei einigen Filmen eine sehr hohe Aussagekraft bezüglich des Box Office aufweist. Da dieses Subset eine Kombination mehrerer potenziell multikollinearer Prädiktoren ist, wurde für jedes Regressionsmodell bei jedem Film auch der Variance Inflation Factor (VIF) und die Tolerance (TOL) berechnet. Als Akzeptanzmaximum wird $VIF < 10$ gewählt, was in der Literatur als Standard-Threshold gilt (siehe Kapitel Modellierung: Multikollinearität 6.2.2 für Details). Bei den Berechnungen wurde diese Grenze von 10 nie überschritten, so dass dieses Subset sich gut eignet. Durch Einbeziehung des PT-NT Werts fließen in dieser Variante auch die Sentiments der Tweets direkt in das Modell ein.

Die dritte und letzte Variante ergibt sich aus dem einzelnen Prädiktor *Sum.Pos.Score* . Diese Variable ist die Summe aller positiv klassifizierten Tweetscores. Da das SVM Modell keine Skalawerte beeinhaltet, wird diese Variante nur für die beiden Lexiconmodelle verwendet. Jeder Film wurde mit den 3 Feature Subsets vorhergesagt. Zusätzlich wird zwischen Regressionsmodell mit Lexicon-Input, Extended Lexicon Input und SVM Input unterschieden. Es ergeben sich somit je Film 3 Subsetausgaben zu je 3 Modellen. Insgesamt wurden in der Evaluierung somit **über 80** verschiedene Ergebnisse aus der Regression abgeleitet, um möglichst viele Daten zum Vergleich zu haben. Aus diesen Ergebnissen werden nun einige vorgestellt.

[3]Es wurde auch testweise die Ridge Regression angewendet, um die Stabilität der Variablen zu prüfen. Dies fließt aber im weiteren Verlauf nicht in die Analyse ein.

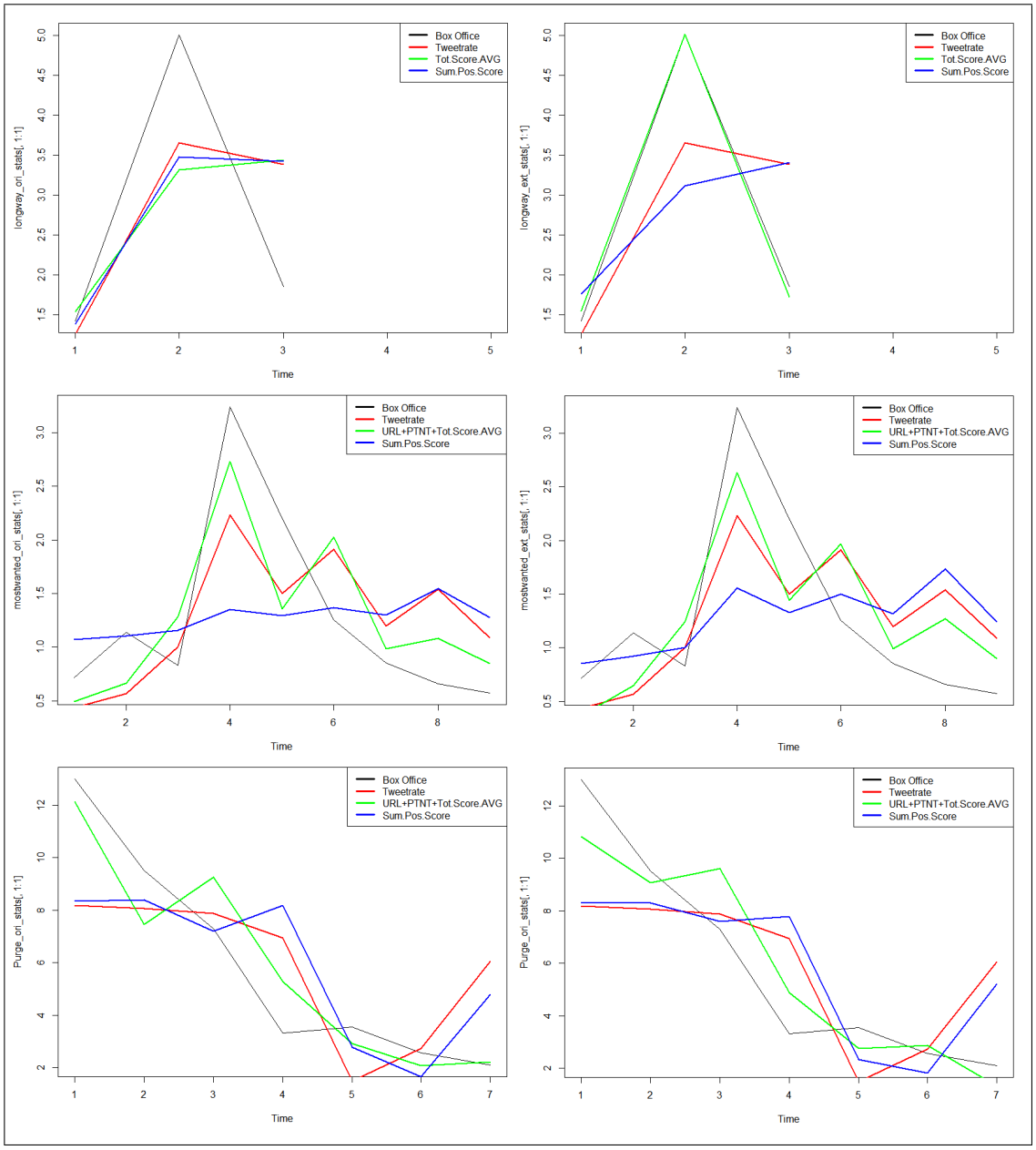


Abbildung 8.3: Prediction vs Real Box Office und Lexicon vs Extended Lexicon

In der Abbildung 8.3 sind die Prediction Ergebnisse im Vergleich zu den tatsächlichen Box Office Verkäufen dargestellt. Die Abbildung ist für jeden Film zeilenweise zu lesen: In der ersten Zeile ist im linken Plot die Original Lexicon Variante von SentiStrength zu sehen. Der rechte Plot zeigt die selben Prädiktoren mit Ergebnissen aus dem Extended Lexicon an. Der Box Office mit schwarz gekennzeichnet. Die zuvor genannten Subsets der Featuressind mit Rot, Grün und Blau gezeichnet. Die Verläufe zeigen die Gegenüberstellungen von tatsächlichen Verkäufen (y-Achse) im Zeitverlauf. Da der Film *A Long Way Down* nur begrenzte Tweets hatte, ist für den Film kein längerer Verlauf zu sehen. Es fällt allerdings bei diesem Film auf, dass das Verfahren mit Extended Lexicon bei dem Prädiktor Tot.Score.Avg fast vollständig auf dem tatsächlichen Verkaufsverlauf liegt. Der entsprechende p-Value für den Prädiktor ist 0.041 * mit einem adjusted R^2 von 0.992. Der hohe Wert des Bestimmtheitsmaßes ist in der Abbildung an der Überlappung der beiden Verläufe zu erkennen.

Bei den Titeln *A Most Wanted Man* und *The Purge: Anarchy* zeigt der Prädiktor URL/PTNT/Tot.Score.Avg die besten Werte mit p-Value 0.043 * für die Anzahl der URLs und $R^2 = 0.412$. Der p-Value ist signifikant, doch der adjusted R-Square ist nicht sehr hoch. Trotzdem ist an der Abbildung zu erkennen, das die Verläufe Box Office und Prediction mit URLs sich ähneln.

Generell gibt es ein hohes Maximum bei Erscheinen eines Films. Danach folgen kleinere Maxima, bis am Ende die Werte kleiner werden. Das zeigt, dass die Aufmerksamkeitsspanne der Kundschaft für einen Film höchstens einige wenige Tage nach Release andauert. Daraus könnte die Konsequenz gezogen werden, dass es im Interesse eines Filmtitels it, in den ersten 2 bis 3 Tagen des Releases den Großteil der Verkäufe zu erzielen, da danach wie in den Abbildungen zu sehen ist, die Verkäufe und der *Social Media Buzz* (Tweets und Diskussionen in Social Media) rapide sinken. Hinzu kommt, dass nach dieser **Aufmerksamkeitsspanne** bereits der nächste Film am Freitag der jeweiligen Woche anläuft. Es liegt also im Interesse des Rechteinhabers eines Films, so viel Aufmerksamkeit wie möglich mit einem möglichst großen Zeitabstand zum nächsten *Releasefenster* zu erzeugen. Dazu könnten virale Werbekampagnen oder aktive Teilnahmen der Schauspieler (wie am Beispiel Hercules mit Retweets gesehen) an Twitter beitragen.

Die Abbildungen 8.4 und 8.5 zeigen die Vorhersagen der restlichen Filme. Wie auch im vorigen Bild zu sehen, ist das Modell mit dem zweiten Subset generell sehr nah an den tatsächlichen Verkäufen. Es scheint, dass die Prädiktoren des zweiten Subsets mittels Extended Lexicon die größte prädiktive Aussagekraft haben (grüne Linien). Dies würde bestätigen, dass sowohl die Anzahl der Tweets bzw. URLs als auch die Stimmungsanalyse starke Prädiktoren sind. Die Gesamtergebnisse der t-Tests können der Tabelle 8.2 entommen werden. Die Tabelle verwendet als Signifikanzintervalle folgende Werte: Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1. [4]

Die Abbildung 8.6 zeigt die Ergebnisse der multiplen linearen Regression unter Verwendung der Ergebnisse der Support Vector Machine. Das SVM Modell erreicht keine hohen adjusted R-Squared Werte. Auch die p-Values sind zum großen Teil insignifikant. Es

[4] Nur korrigierte (adjusted R^2 werden genannt, da diese niedriger sind, aber auch die Anzahl der Prädiktoren miteinbeziehen.

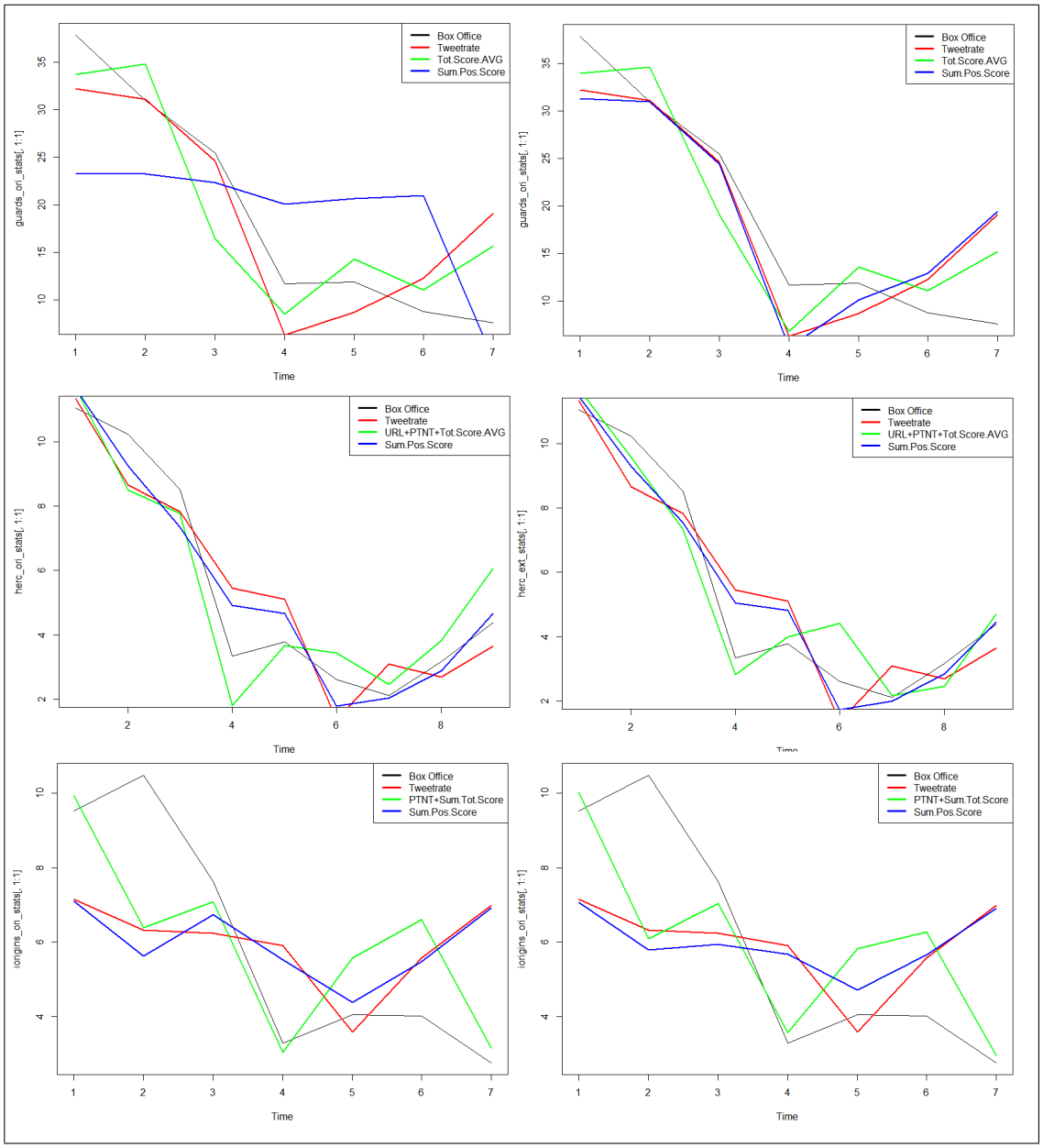


Abbildung 8.4: Prediction vs Real Box Office und Lexicon vs Extended Lexicon

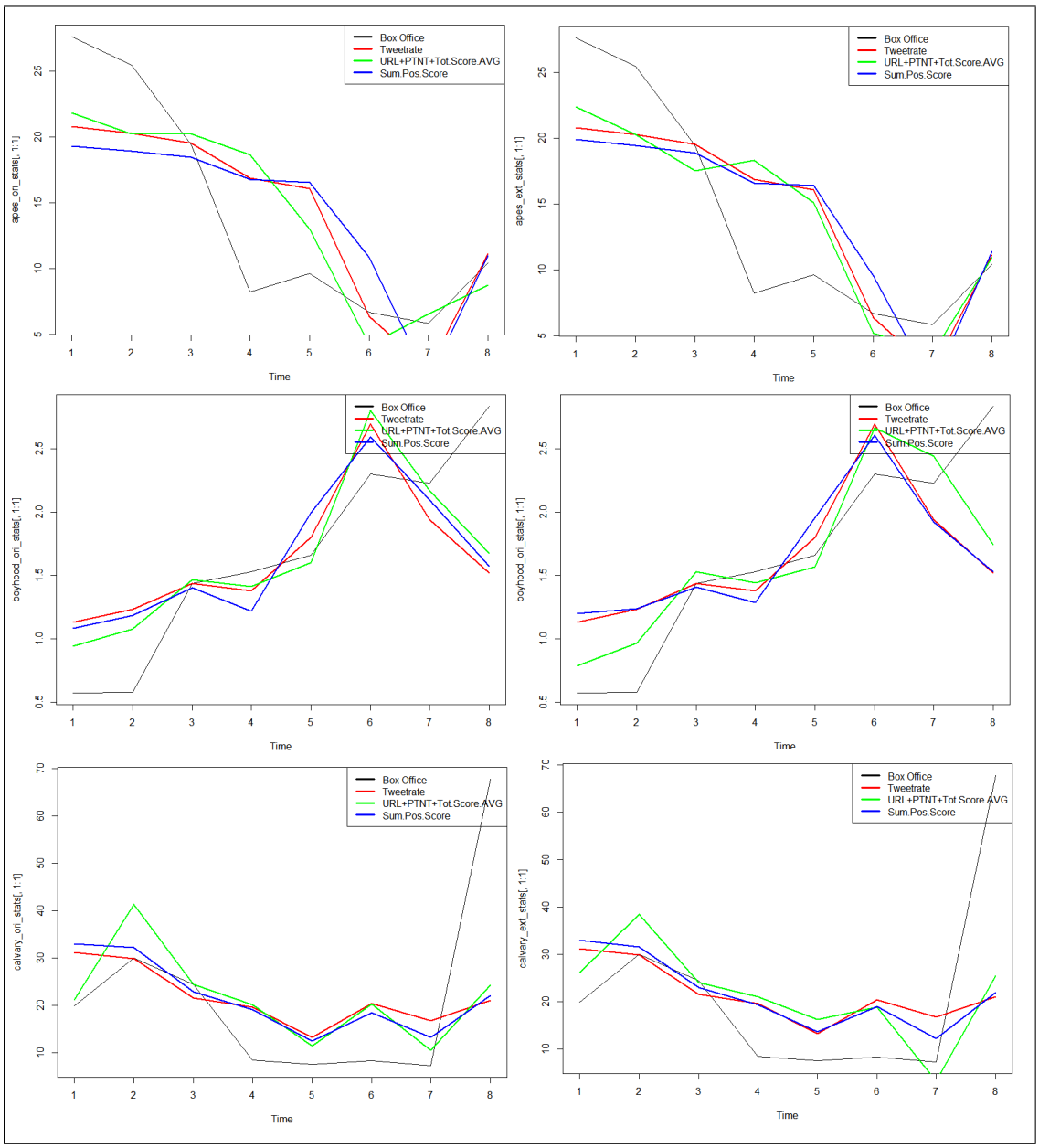


Abbildung 8.5: Prediction vs Real Box Office und Lexicon vs Extended Lexicon

Tabelle 8.2: p-Values und adjusted R^2für Extended und Original Lexicon

Filmtitel	p-Value: Sub1	adj R^2: Sub1	p-Value: Sub2	adj R^2: Sub2	p-Value: Sub3	adj R^2: Sub3
Hercules(orig)	0.00024 ***	0.852	0.0073 **	0.829	2.7e-05 ***	1
Hercules(ext.)	0.00024 ***	0.852	0.00229 **	0.893	2.9e-05 ***	1
Planet of Apes(orig)	0.02048 *	0.556	0.211	0.371	0.05273 .	0.407
Planet of Apes(ext.)	0.02048 *	0.556	0.12	0.355	0.03652 *	0.469
Calvary(orig)	0.478	-0.0651	0.60	-0.375	0.362	-0.00399
Calvary(ext.)	0.478	-0.0651	0.58	-0.337	0.37	-0.0118
Most Wanted Man(orig)	0.052 .	0.359	0.043 *	0.412	0.667	-0.111
Most Wanted Man(ext.)	0.052 .	0.359	0.062 .	0.367	0.37	-0.0116
Guardians of Galaxy(orig)	0.01110 *	0.706	0.0091 **	0.727	0.1806	0.191
Guardians of Galaxy(ext.)	0.01110 *	0.706	0.0050 **	0.784	0.0161 *	0.662
The Purge:Anarchy(orig)	0.110	0.315	0.043 *	0.744	0.0910 .	0.359
The Purge:Anarchy(ext.)	0.110	0.315	0.055 .	0.726	0.092 .	0.358
Boyhood(orig)	0.096 .	0.292	0.033 *	0.483	0.073 .	0.347
Boyhood(ext.)	0.01110 *	0.706	0.0050 **	0.784	0.0161 *	0.662
I Origins(orig)	0.408	-0.0321	0.088 .	0.345	0.498	-0.0843
I Origins(ext.)	0.408	-0.0321	0.099 .	0.302	0.587	-0.124
Long Way Down(orig)	0.53	-0.0994	0.63	-0.41	0.58	-0.257
Long Way Down(ext.)	0.53	-0.0994	0.041 *	0.992	0.70	-0.599

gibt allerdings auch weniger mögliche Prädiktoren als bei den Lexicon Varianten, da das Modell eine 3-Class SVM ist (positiv, negativ,neutral), wohingegen die SentiStrength Ergebnisse auf den Skalawerten $[-4, 4]$ beruhen. Insbesondere der Prädiktor **Pos+Neg** erweist sich als sehr schwach bezüglich der Vorhersagen.

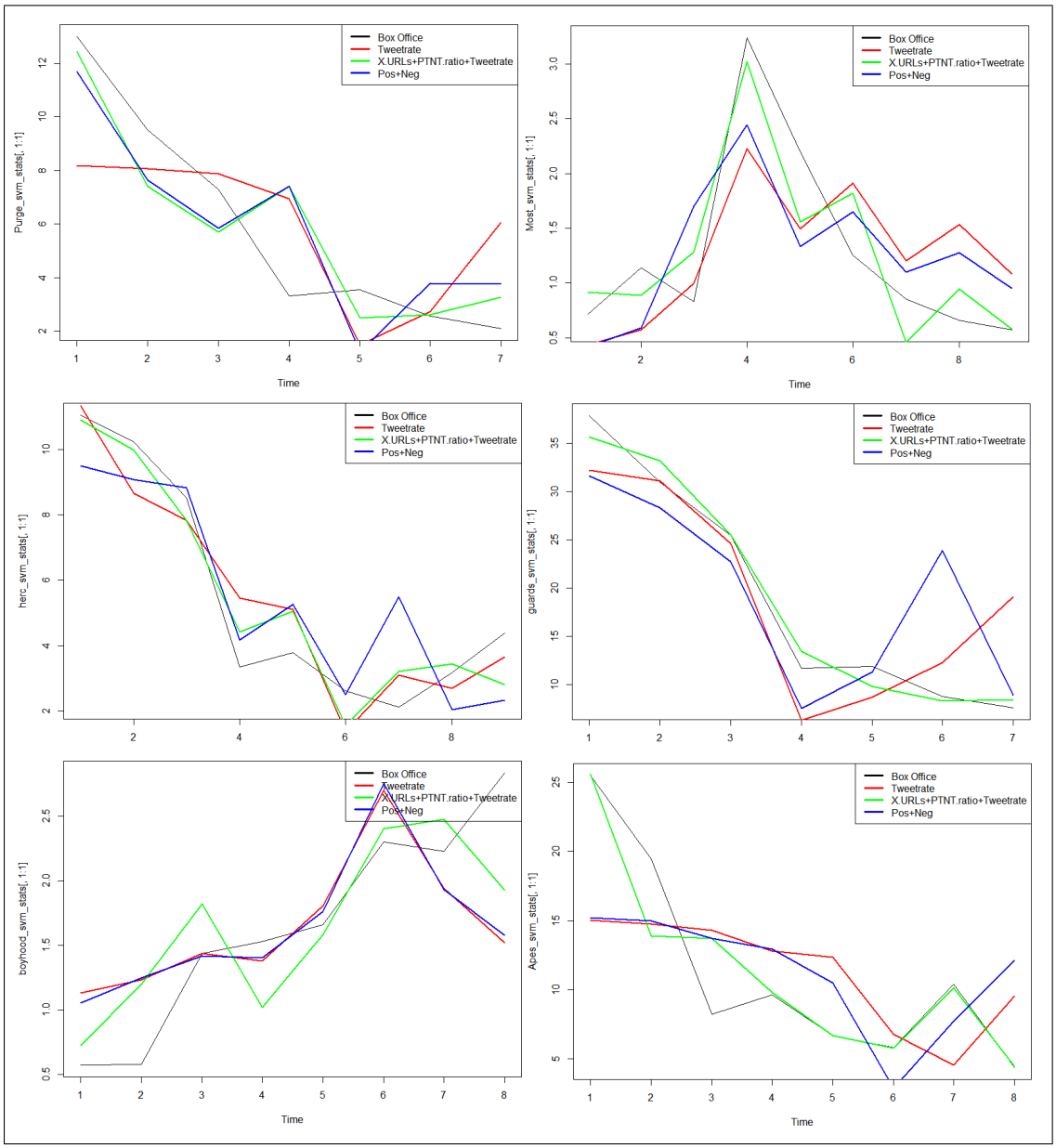


Abbildung 8.6: Prediction vs Real Box Office: SVM Modell

Die Tabelle 8.3 zeigt die Signifikanzwerte der Abbildung 8.6. Die Subsets entsprechen denen der Abbildung 8.6

Insgesamt sind fast nur die Prädiktoren mit der Tweetrate (Subset 1) im Fall der SVM Variante signifikant. Hierbei handelt es sich aber um die identischen Urpsrungstweets wie

Tabelle 8.3: p-Values und adjusted R^2 für SVM Modell

Filmtitel	p-Value: Sub1	adj R^2: Sub1	p-Value: Sub2	adj R^2: Sub2	p-Value: Sub3	adj R^2: Sub3
Hercules(svm)	0.00024 ***	0.8519	0.003018 **	0.8647	0.01476 **	0.673
Planet of Apes(svm)	0.175	0.1625	0.0241 *	0.7182	0.3792	0.0501
Most Wanted Man(svm)	0.052 .	0.359	0.0384 .	0.6623	0.1405	0.3069
Guardians of Galaxy(svm)	0.01110 *	0.706	0.004992 **	0.9588	0.07330 .	0.4842
The Purge:Anarchy(svm)	0.110	0.315	0.2034	0.4857	0.1061	0.5115
Boyhood(svm)	0.096 .	0.292	0.155	0.3358	0.2456	0.2016

die Lexicon Daten. Das SVM Modell ist weniger aussagestark als das Lexicon Modell. Bei dem Extended Lexicon ist zu sehen, dass vor allem viele Prädiktoren der Subsets 1 und 2 statistische Signifikanz aufweisen.

8.3 Vergleich mit IMDB Portal

Zur Berechnung einer möglichen Korrelation von Sentiment Scores und dem Erfolg eines Films durch die Bewertungen auf IMDB und den Verkaufszahlen (Box Office) wurde in dieser Arbeit eine Metrik eingeführt. Die PT-NT Metrik könnte eine Vergleichbarkeit der Stimmungsanalyse mit IMDB Bewertungen liefern. Die Aufstellung der einzelnen Werte ist der Abbildung 8.7 zu entnehmen. Die Profit Ratio ist der Quotient $BoxOffice - Budget/Budget$. Durch diesen Quotienten ist eine bessere Erkennung des Erfolges eines Films sichtbar, da der reine Box Office nur die Verkäufe nennt, nicht aber ob ein Gewinn oder Verlust entstanden ist. Laut der Profit Ratio waren die beiden Titel *Boyhood* und *The Purge:Anarchy* sehr erfolgreich. Sie brachten mehr als den 10-fachen Betrag des Budgets wieder ein. Bei der IMDB Bewertung spiegelt sich dieser große Erfolg aber nur bei Boyhood wieder. Beide Filme hatten ein relativ kleines Budget und erhalten trotz geringerer Verkäufe als beispielsweise Guardians of Galaxy mit fast 800 $ Mio. eine höhere Profit Ratio. Dieser Erfolg ist bei The Purge auch bei dem PT-NT Wert zu sehen, der am höchsten von allen Filmen ist. Boyhood allerdings erhält zwar eine hohe PT-NT Ratio, diese steht aber in keinem Verhältnis zu der IMDB Bewertung.

Eine simple Korrelationsanalyse ohne Annahmen einer Kausalität zeigt bei der Relation Lexicon(orig) zu IMDB Score einen Koeffizienten von $0,51$. Bei dem Vergleich SVM mit IMDB erhält man einen Koeffizienten von $-0,711$. Da allerdings die Experimente und Beobachtungen den Schluss zulassen, dass das SVM Modell optimiert werden muss und im t-Test kaum statistische Signifikanz aufweist, hat diese mögliche Relation keine Bedeutung. Insgesamt scheint der PT-NT Wert keine vergleichbare Basis zur IMDB Bewertung zu bieten. Da es zum Vergleich keine verfügbaren kontinuierlichen IMDB Scores gibt, ist keine Zeitreihenanalyse möglich. In einer zukünftigen Arbeit könnte allerdings ein simpler HTML-Crawler zum Extrahieren von IMDB User-Foren implementiert werden, der über eine bestimmte Zeitspanne Daten sammelt. Eine ähnliche Methode verwenden Oghina et. al. in [18].

	SVM	Lexicon PTNT (orig.)	Lexicon PTNT (ext.)	IMDB (1.0-10)	Budget (in Mio $)	Box Office (in Mio $)	Profit (in Mio $)	Profit Ratio
Boyhood	0,56	2,53	3,36	8,1	4	44,39	40,39	10,098
Dawn of the Planet of Apes	0,61	2,49	3,19	7,7	170	703,5	533,5	3,138
Guardians of Galaxy	1,07	2,6	3,11	8,1	170	774,17	604,17	3,554
Hercules	1,81	2,34	2,9	6,1	100	243,4	143,4	1,434
A Most Wanted Man	1,89	2,15	2,69	6,9	15	17,5	2,5	0,167
The Purge:Anarchy	4,05	3,1	6,46	6,5	9	112	103	11,444
I Origins	n.a.	4,71	5,13	7,3	0,1	0,4	0,3	3,000
Calvary	n.a.	2,66	2,99	7,5	8	16,89	8,89	1,111
Child of God	n.a.	4,48	4,68	5,5	25	0,038	-24,962	-0,998
A Long Way	n.a.	2,72	2,92	6,4	22,7	7,17	-15,53	-0,684

Abbildung 8.7: Prediction vs Real Box Office und Lexicon vs Extended Lexicon

9 Zusammenfassung und Ausblick

In dieser Diplomarbeit wurde erforscht, inwieweit Daten aus dem stetig wachsenden Big Data der sozialen Medien gesammelt, aufbereitet und für lexicale und maschinelle Lernverfahren verwendet werden können. Auch wurden neue Ansätze vorgestellt, wie mittels Expansionsalgorithmus anhand der SentiStrength API der Wortschatz eines Lexicons erweitert werden kann, um bessere Prognosen aufzustellen.

Im Detail wurden Methoden gezeigt, wie die Erstellung eines Textkorpus für Social Media Data Mining funktionieren kann. Dies wurde am Beispiel von Twitter gezeigt. Es konnte festgestellt werden, dass die Vorverarbeitung der Daten vor allem in Social Media sehr wichtig ist, da sehr viel Noise und Spam mit in die Datenkollektion aufgenommen wird. Im Fall von Twitter wurden Möglichkeiten der Nutzung mit der Twitter API erklärt.

Zur Stimmungsanalyse wurde das Lexicon Verfahren anhand von SentiStrength vorgestellt. Der SentiStrength Algorithmus konnte in Umgebungen mit viel Umganssprache erfolgreich die Sentiments erkennen und bewerten. Dazu wurde in dieser Arbeit eine neue domänenspezifische Lexiconerweiterung in die Originalliste von SentiStrength integriert, um die Erkennung von Slang noch weiter zu verbessern. Die Experimente beweisen, dass die Erweiterung die Klassifikationsgenauigkeit und Vorhersagestärke der späteren Modelle etwas verbesserte. Um noch größere Verbesserungen zu erreichen, wurde das Schema eines Algorithmus zur automatischen Expansion erwähnt.

Die Stimmungsanalyse zeigte, dass die Menschen auf Twitter sehr viel und gerne über Filme reden. Die vorliegende Arbeit machte sich diese Erkenntnis zunutze und es wurde eine Datenbasis von rund 2.000.000 Tweets aufgebaut und verarbeitet. Zur besseren Nutzung der Ressourcen und zum Zwecke der Optimierung wurden allgemeine und aktuelle Feature Subset Selection Methoden erläutert und im späteren SVM und Lexicon Modell praktisch angewendet.

Für eine prädiktive Analyse wurden Regressionsmodelle und Klassifikationsmodelle in verschiedenen Variationen erstellt. Die Modelle wurden in R und RapidMiner realisiert und diskutiert.

Die Experimente mit SVM und SentiStrength haben gezeigt, dass noch sehr viel Forschung im Bereich der Sentiment Analysis gemacht werden kann. Lexiconmethoden könnten weiter verbessert werden und mit Machine Learning Verfahren kombiniert werden. Eine mögliche Art der Kombination wurde in dieser Diplomarbeit gezeigt, indem die Trainingsdatensätze für die Support Vector Machine von der Lexiconmethode erstellt wurden. Diese kombinierte *hybride* Methode zeigt hohe Accuracy Werte, leidet aber noch daran, dass das Modell optimiert und die Trainingsdatenmenge vergrößert werden muss. Auch wurde in der Arbeit festgestellt, dass trotz Preprocessing Massnahmen im-

mer noch sehr viele irrelevante Tweets in den Korpus gelangen können. Dies liegt zum Einen an Bugs der Twitter API (z.B. keine Sprachdetektion) und zum Anderen an der sehr großen Menge an Twitterdaten, die gesammelt werden können.

Insgesamt war die Vorhersage von Box Office Verkaufszahlen mit einigen der Prädiktor Subsets erfolgreich. Es wurde gezeigt, dass es mittels multipler linearer Regression möglich ist, etwaige Zusammenhänge zu finden. Zur Verifikation dieser Ergebnisse könnten in einer zukünftigen Arbeit auf diesem Gebiet alternative Analysemodelle verwendet oder kombiniert werden. Auch wurde festgehalten, dass im Falle von Twitter die Daten entweder von einem Reseller erworben oder sehr lange im Voraus extrahiert werden müssen, da Twitter keine historischen Analysen mehr erlaubt.

In den Experimenten wurde die Feststellung gemacht, dass insbesondere im Zeitfenster von 1 bis 2 Tagen vor und nach Erscheinen eines Films, die höchsten Umsätze und auch die höchsten Tweetzahlen erreicht wurden. Trotzdem können z.B. bei Nischenproduktionen oder Sonderfällen (Hauptdarsteller stirb, abe Film erscheint post mortem) keine sicheren Vorhersagen über den Erfolg gemacht werden. Bei größeren Produktionen können allerdings schon alleine mittels der Tweetrate Zusammenhänge und Indikatoren für den späteren Erfolg eines Films gesehen werden. In dieser Diplomarbeit wurden auch Metadaten wie Anzahl der Follower und User u.ä. gesammelt aber nicht verwendet. Für eine zukünftige Arbeit könnte überprüft werden, ob es besonders einflussreiche User gibt (mit vielen Followern), die die Gesamtsentiments auf Twitter steuern. Dadurch könnten Zusammenhänge aufgedeckt werden.

Alle Methoden in dieser Arbeit könnten in abgeänderter Form in einer anderen Domäne verwendet werden. Es müsste lediglich eine Erweiterung des Lexicons durchgeführt werden. Diese Erweiterung ist keine Pflicht, würde jedoch zur Verbesserung der Performanz in der neuen Domäne führen. Die Erweiterung könnte manuell, halbautomatisch oder vollautomatisch mittels Expansionsalgorithmus laufen.

Literaturverzeichnis

[1] *Predicting Elections with Twitter:What 140 Characters Reveal about Political Sentiment, (AAAI)*, volume 4212, 2010.

[2] *Quantifying Movie Magic with Google Search*, 2012.

[3] *Predicting movie success and academy awards through sentiment and social network analysis (ECIS 2008)*, 2012, USA.

[4] *Combining an SVM Classifier and Character N-gram Language Models for Sentiment Analysis on Twitter Text*, 2013.

[5] *Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data (PLOS One 8(8))*, 2013, Hungary.

[6] Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1):237–260, 1998.

[7] amazon mechanical turk. www.mturk.com, 27.04.2014, 12:45, 2014. [Online; accessed 27.04.2014, 12:30].

[8] Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 492–499, Washington, DC, USA, 2010. IEEE Computer Society.

[9] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC'10*, pages –1–1, 2010.

[10] Yanwei Bao, Changqin Quan, Lijuan Wang, and Fuji Ren. The role of pre-processing in twitter sentiment analysis. In *Intelligent Computing Methodologies*, pages 615–624. Springer, 2014.

[11] PATRICK BEUTH. `http://www.zeit.de/digital/datenschutz/2014-03/bka-data-mining-predictive-policing`, 2014. [Online; accessed 17.09.2014, 14:50].

[12] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

[13] box office mojo. www.boxofficemojo.com, 2014. [Online; accessed 27.04.2014, 12:30].

[14] Moses Charikar, Venkatesan Guruswami, Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. Combinatorial feature selection problems. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 631–640. IEEE, 2000.

[15] Condor. http://www.ickn.org/condor.html, 2014. [Online; accessed 07.05.2014, 00:15].

[16] cyberemotions. `http://www.cyberemotions.eu/`, 2013. [Online; accessed 13.08.2014, 15:20].

[17] Paul Earle, Daniel Bowden, and Michelle Guy. Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics*, 54(6), 2012.

[18] ECIR12. *Predicting IMDB Movie Ratings Using Social Media (ECIR12)*, 2012.

[19] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06*, pages 417–422, 2006.

[20] Donald E Farrar and Robert R Glauber. Multicollinearity in regression analysis: the problem revisited. *The Review of Economic and Statistics*, pages 92–107, 1967.

[21] Matthew S. Gerber. Predicting crime using twitter and kernel density estimation. *Decision Support Systems*, 61(0):115 – 125, 2014.

[22] Arthur Stanley Goldberger et al. Econometric theory. *Econometric theory.*, 1964.

[23] Michael H Graham. Confronting multicollinearity in ecological multiple regression. *Ecology*, 84(11):2809–2815, 2003.

[24] Emma Haddi, Xiaohui Liu, and Yong Shi. The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17(0):26 – 32, 2013. First International Conference on Information Technology and Quantitative Management.

[25] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[26] Prof. Dr. Michael Hess. Vorlesungsskript einführung in die computerlinguistik (i). Vorlesungsskript 1, Universität Zürich, Institut für Computerlinguistik, Zürich, Schweiz, Wintersemester 2005.

[27] Vasu Jain, editor. *Prediction of Movie Success using Sentiment Analysis of Tweets. (JSCSE)*, volume 3, University of California, USA, 2012.

[28] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*. Springer, 2013.

[29] George H John, Ron Kohavi, Karl Pfleger, et al. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129, 1994.

[30] David Reinsel John Gantz. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east, december 2012.

[31] Moshe Koppel and Jonathan Schler. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109, 2006.

[32] Lingpipe. http://alias-i.com/lingpipe/, 2014. [Online; accessed 27.04.2014, 16:15].

[33] Inc. LIWC Pennebaker Conglomerates. `http://www.liwc.net/`, 2014. [Online; accessed 07.08.2014, 15:20].

[34] Andrius Mudinas, Dell Zhang, and Mark Levene. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 5:1–5:8, New York, NY, USA, 2012. ACM.

[35] M.S. Neethu and R. Rajasree. Sentiment analysis in twitter using machine learning techniques. In *Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on*, pages 1–5, July 2013.

[36] NYSE. http://basicsmedia.com/twitter-inc-nysetwtr-continues-to-lag-facebook-despite-strong-q4-results-8547, 2014. [Online; accessed 06.04.2014, 16:35].

[37] Robert M O'brien. A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity*, 41(5):673–690, 2007.

[38] University of Waikato. Weka, may 2013.

[39] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[40] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[41] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[42] Wiebke Peterson. Einführung in die computerlinguistik. Vorlesungsskript 1, Heinrich Heine Universität Düsseldorf, Philosophische Fakultät, Düsseldorf, Deutschland, Wintersemester 2010.

[43] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding domain sentiment lexicon through double propagation. In *IJCAI*, volume 9, pages 1199–1204, 2009.

[44] Rapid-I. `http://www.pressebox.de/pressemitteilung/rapid-i-gmbh/RapidMiner-findet-heraus-Waschmittel-stinkt/boxid/200987`, 2008. [Online; accessed 10.08.2014, 13:20].

[45] RapidMiner. Rapidminer, may 2013.

[46] Hollywood stock exchange. www.hsx.com, 27.04.2014, 12:55, 2014. [Online; accessed 27.04.2014, 12:30].

[47] Wall street journal. `http://online.wsj.com/news/articles/SB10001424052702304441404579118531954483974`, 2014. [Online; accessed 05.04.2014, 16:00].

[48] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[49] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, December 2010.

[50] USA today30. http://usatoday30.usatoday.com/news/politics/twitter-election-meter, 2014. [Online; accessed 05.04.2014, 16:30].

[51] Peter D. Turney. 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*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[52] Twitter. `https://dev.twitter.com`, 2014. [Online; accessed 03.09.2014, 12:18].

[53] David G. Underhill, Luke McDowell, David J. Marchette, and Jeffrey L. Solka. Enhancing text analysis via dimensionality reduction. In *IRI*, pages 348–353. IEEE Systems, Man, and Cybernetics Society, 2007.

[54] University of Texas. *User rating prediction for movies.Technical report*, University of Texas at Austin,USA, 2008.

[55] Yusuke Yamamoto. `http://twitter4j.org/en/`, 2014. [Online; accessed 03.09.2014, 18:41].

Erklärung

Hiermit erkläre ich, Ahmet Celikkaya, die vorliegende Diplom-Arbeit mit dem Titel *Success prediction of upcoming movies through lexicon- and learning-based sentiment analysis algorithms on preview information in social media* selbständig verfasst und keine anderen als die hier angegebenen Hilfsmittel verwendet, sowie Zitate kenntlich gemacht zu haben.

Dortmund, 13. April 2015