Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Tobias Eichler

Sentiment-Analysis durch überwachtes Lernen: Vergleich und Bewertung von Konzepten zur Vorverarbeitung

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Tobias Eichler

Sentiment-Analysis durch überwachtes Lernen: Vergleich und Bewertung von Konzepten zur Vorverarbeitung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Neitzke
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Eingereicht am: 21. August 2012

Tobias Eichler

Thema der Arbeit
Sentiment-Analysis durch überwachtes Lernen: Vergleich und Bewertung von Konzepten zur Vorverarbeitung

Stichworte
Sentiment-Analysis, Opinion-Mining, Klassifikation, überwachtes Lernen, UIMA, Vorverarbeitung, Subjektivität, Negation

Kurzzusammenfassung
Sentiment-Analysis wird durch die wachsende Anzahl der Informationen in Textform im Internet immer wichtiger. Diese Bachelorarbeit untersucht verschiedene Vorverarbeitungskonzepte zur Verbesserung der Sentiment-Analysis, die durch überwachtes Lernen durchgeführt wird. Dazu zählen die Subjektivitätsanalyse und die Negationsverarbeitung. Zu den einzelnen Konzepten werden aktuelle Ansätze aus der Forschung erläutert und im Anschluss verglichen und bewertet. Im Rahmen der Arbeit wurde eine Testumgebung auf Basis von UIMA, einer Architektur zur Verarbeitung von unstrukturierten Informationen, wie natürlicher Sprache, entwickelt, mit der ein großer Teil der vorgestellten Ansätze getestet wurde. Die im praktischen Teil der Arbeit entwickelte Testumgebung ist wiederverwendbar und die damit erzielten Ergebnisse ermöglichen einen direkten Vergleich der vorstellten Ansätze, der so bisher noch nicht durchgeführt wurde. Die Testergebnisse zeigen, dass die vorgestellten Konzepte zur Vorverarbeitung dazu in der Lage sind, die Ergebnisse der Sentiment-Analysis von Dokumenten spürbar zu verbessern. Die vorgestellten Vorverarbeitungsschritte erreichen zusammen in der Untersuchung eine Verbesserung des F-Score-Wertes um 5% auf etwa 84%.

Tobias Eichler

Title of the paper
Sentiment Analysis by Supervised Learning: Comparison and Evaluation of Concepts for Preprocessing

Keywords
Sentiment Analysis, Opinion Mining, Classification, Supervised Learning, UIMA, Preprocessing, Subjectivity, Negation

Abstract

Due to the increasing amount of written information, sentiment analysis techniques are getting more important. This thesis is about examining different pre-processing concepts to improve sentiment analysis carried out by supervised learning, including subjectivity analysis and negation processing. Regarding each single concept current approaches in research are explained as well as compared and evaluated. As practical part of this work a test environment on a UIMA basis was developed. UIMA is an architecture for processing unstructured information such as natural language. It was used to test a large number of those approaches presented in this paper. The test environment which was developed in the practical part can be re-used and the results make it possible to compare the approaches directly which hasn't been done so far. The test results show that the selected pre-processing concepts improve the results of sentiment analysis to a considerable degree. By means of the pre-processing steps altogether, a 5% increase of the F score to approximately 84% was achieved in this test.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einführung

1.1 Motivation

Durch steigendes Interesse an der automatischen Extraktion und Klassifikation von Meinungen in Texten wird *Sentiment Analysis* immer wichtiger. Mit *Sentiment Analysis* wird versucht, Meinungen in Texten zu analysieren. Zum Beispiel soll festgestellt werden, wo und ob Texte Meinungen enthalten und was diese Meinungen aussagen. Das Interesse an der automatischen Verarbeitung von Texten wird mit der steigenden Anzahl von Informationen in Texten durch das Internet immer größer.

Es gibt viele Anwendungsmöglichkeiten von *Sentiment Analysis*-Verfahren, die in der Praxis immer häufiger benutzt werden. Dazu zählen Aufgaben im Bereich der automatischen Textzusammenfassung, dem automatischen Beantworten von Fragen mit den Informationen aus Texten und die Klassifikation von Dokumenten. Für diese verschiedenen Anwendungsfälle werden Sätze, Absätze oder ganze Dokumente untersucht.

Die Zusammenfassung von Texten erfordert die Analyse jedes Satzes. Hier kann die Information, ob ein Satz positiv oder negativ ist, zum Beispiel für Produktbewertungen dazu benutzt werden, die positiven und negativen Punkte zusammenzufassen. Die automatische Zusammenfassung von Produktbewertungen und anderen Rückmeldungen von Kunden erlaubt es Unternehmen, diese Rückmeldungen besser auszuwerten und potentielle Kunden können sich schneller einen Überblick über die allgemeine Bewertung eines Produkts machen.

Question Answering Applications versuchen für an das Programm gerichtete Fragen, die korrekte Antwort aus Texten zu extrahieren. Dabei kann bei auf bestimmte Meinungen abzielende Fragen *Sentiment Analysis* verwendet werden. Dies kann zum Beispiel für Suchmaschinen eingesetzt werden, die nach Antworten auf die gestellte Frage und nicht nach dem Vorkommen von bestimmten Stichwörtern suchen. Eine Meinungssuchmaschine könnte zum Beispiel die allgemeine Meinung zu einem Thema, wie einem bestimmten Produkt, untersuchen und diese

Informationen dem Benutzer bereitstellen.

Diese Arbeit konzentriert sich auf die Klassifikation von Dokumenten. Ein möglicher, hier für die Auswertung weiterverfolgter Anwendungsfall ist die Klassifikation von Dokumenten nach der enthaltenen Meinung in positiv und negativ. Diese Dokumente könnten zum Beispiel Produktbewertungen von Kunden sein. Durch die Klassifikation ist es auch möglich, Spam, der in den Rückmeldungen enthalten sein kann, zu identifizieren. Für diese Untersuchung werden Filmbewertungen verschiedener Filme von Benutzern aus dem Internet als Dokumente benutzt. Diese sollen danach unterschieden werden, ob der Benutzer eher positiv oder eher negativ über den bewerteten Film denkt.

Für alle diese Anwendungsfälle können Vorverarbeitungsschritte eingesetzt werden, mit denen versucht wird, die Ergebnisse der Analyse zu verbessern. Für die Klassifikation werden die Dokumente in Merkmale, Features genannt, zerlegt. Anhand dieser Features wird versucht, die Klasse zu bestimmen. Als Vorverarbeitung kann die Verarbeitung der Texte vor der Feature-Extraktion und die Feature-Extraktion selber bezeichnet werden. Die Vorverarbeitung bestimmt, wie der Text im Dokument für die Klassifikation repräsentiert wird. Dies hat starke Auswirkungen auf den Erfolg bei der Klassifikation.

Die hier vorgestellten Konzepte zur Vorverarbeitung können auch in anderen Anwendungsgebieten eingesetzt werden. Zum Beispiel ist die Negationsverarbeitung für alle Anwendungen im Bereich der Textverarbeitung interessant.

1.2 Zielsetzung

Mit dieser Arbeit sollen die zwei am häufigsten verwendeten Vorverarbeitungsschritte für *Sentiment Analysis* näher untersucht werden. Bei der *Sentiment Analysis* geht es darum, die enthaltene Meinung in einem Dokument möglichst gut zum Beispiel in positiv und negativ zu klassifizieren.

Die beiden untersuchten Konzepte sind die Subjektivitätsanalyse und die Negationsverarbeitung von Sätzen. Bei der Subjektivitätsanalyse wird versucht, die Fakten, die Teile des Textes, die keine Meinungen enthalten, in einem Dokument zu kennzeichnen, damit sie bei der Verarbeitung ignoriert werden können. Die Negationsverarbeitung versucht, Negationen zu finden und deren Auswirkungen für die spätere Verarbeitung sichtbar zu machen.

Im Rahmen dieser Arbeit soll geklärt werden, welche Ansätze es gibt und welche Ergebnisse diese im direkten Vergleich erzielen können. Außerdem wird versucht, die Ergebnisse aus der Literatur nachzuvollziehen. Für die Untersuchung werden Texte aus der Bewertung verschiedener Filme verwendet. Es wird untersucht, mit welchen Ansätzen sich die Klassifikation dieser Filmbewertungen in die Klassen Positiv und Negativ in welchem Maß verbessern lässt.

Um dieses Ziel zu erreichen, soll zuerst ein Überblick über die dafür relevanten Grundlagen gegeben werden. Dazu zählt der Ablauf der *Sentiment Analysis* und Algorithmen für die Klassifikation mit Hilfe von überwachtem Lernen. Außerdem wird die Repräsentation des Textes für die Klassifikation erläutert, da diese wichtig für die Vorverarbeitung ist.

Für beide der zu untersuchenden Konzepte sollen aktuelle Forschungsansätze möglichst umfassend beschrieben werden. Es soll gezeigt werden, was der Grundgedanke hinter den Konzepten ist und wie diese in der Theorie funktionieren sollen. Die Ansätze sollen im Detail erläutert werden, um die einzelnen Ergebnisse mit diesem Hintergrundwissen gut vergleichen zu können.

Möglichst viele der vorgestellten Ansätze sollen in einer im Rahmen dieser Arbeit erstellten Testumgebung getestet werden. Für die Testumgebung soll das *UIMA*-Framework verwendet werden. *UIMA* ist eine Standardarchitektur für Anwendungen, die mit natürlicher Sprache arbeiten und wird häufig verwendet. Um alle Ansätze in derselben Testumgebung testen zu können, sollen die einzelnen Komponenten der Testumgebung leicht austauschbar sein. Dafür sollen neben der Verwendung von *UIMA* weitere Maßnahmen ergriffen und erläutert werden. Ein weiteres Ziel ist die Wiederverwendbarkeit der im Rahmen dieser Arbeit erstellten Software, um weitere und auf diese Arbeit aufbauende Untersuchungen im Bereich *Sentiment Analysis* zu erleichtern.

Die Ergebnisse der ausgeführten Untersuchungen in der Testumgebung werden umfassend ausgewertet. Dazu sollen sowohl die Ergebnisse der Ansätze untereinander als auch die besten Ergebnisse der einzelnen Konzepte miteinander verglichen werden. Außerdem sollen die einzelnen Ansätze anhand ihrer Ergebnisse mit den Erwartungen und anderen Untersuchungen verglichen und bewertet werden.

1.3 Gliederung

Diese Arbeit besteht zusammen mit der Einleitung aus insgesamt sechs Kapiteln. Kapitel 2 behandelt die theoretischen Grundlagen von *Sentiment Analysis* und Kapitel 3 enthält die Analyse von verschiedenen Vorverarbeitungsschritten. Im darauf folgenden Kapitel 4 wird der Aufbau der Testumgebung und damit der praktische Teil der Arbeit beschrieben. Die Auswertung der Ergebnisse erfolgt in Kapitel 5. Das abschließende Kapitel ist eine Zusammenfassung der Arbeit.

Anschließend an die Einleitung der Arbeit behandelt Kapitel 2 die Grundlagen von *Sentiment Analysis*. Es wird besonders auf die Klassifikation von Texten durch überwachtes Lernen und die eingesetzten Verfahren eingegangen. Außerdem werden die später eingesetzten Metriken für die Bewertung und den Vergleich der Ergebnisse dieser Arbeit erklärt. Die Grundlagen über *Sentiment Analysis* helfen dabei, die Entwicklung von Vorverarbeitungsschritten zu verstehen.

Kapitel 3 enthält den theoretischen Hauptteil der Arbeit und behandelt verschiedene Vorverarbeitungsschritte für *Sentiment Analysis*. Dazu zählen die Analyse von Subjektivität und die Verarbeitung von Negationen. Es werden verschiedene Ansätze für die einzelnen Konzepte vorgestellt und erläutert.

Die hier besprochenen Vorverarbeitungsschritte sollen im Rahmen dieser Arbeit getestet und ausgewertet werden. Der erste Teil des 4. Kapitels enthält eine Erklärung, welche der in Kapitel 3 vorgestellten Ansätze umgesetzt werden sollen und wie diese realisiert wurden. Anschließend wird der Aufbau der Testumgebung dargestellt und die für die Testumgebung eingesetzten externen Bibliotheken vorgestellt. Im letzten Teil des Kapitels werden einige Probleme beschrieben, die beim Erstellen der Testumgebung in Zusammenhang mit den Architekturentscheidungen auftraten und es wird erklärt, wie diese gelöst wurden.

Kapitel 5 besteht aus dem Vergleich und der Bewertung der verschiedenen ausgewählten Ansätze zur Vorverarbeitung. Alle Ansätze werden auf Basis der in den Grundlagen vorgestellten Metriken durch einen Testlauf in der Testumgebung untereinander und mit anderen Untersuchungen verglichen. Es wird versucht, Gründe für die erzielten Ergebnisse zu finden. Abschließend wird die Auswirkung der einzelnen Vorverarbeitungsschritte auf die Dokumentenklassifikation aufgezeigt.

Das letzte Kapitel (Kapitel 6) der Arbeit liefert eine Zusammenfassung aller Kapitel und Ergebnisse der Arbeit. Weiterhin wird versucht, einen kleinen Ausblick auf weitere mögliche Untersuchungen zu geben.

2 *Sentiment Analysis*

Sentiment Analysis oder auch *Opinion Mining* beschäftigt sich mit der Identifikation oder Suche von Meinungen in Texten und mit deren Klassifikation. Informationen werden als Fakten oder Meinungen unterschieden. Fakten enthalten objektive Beschreibungen eines Sachverhaltes, wohingegen Meinungen die persönliche Sicht ihres Autors ausdrücken. Mit *Sentiment Analysis* wird versucht, diese Informationen zu trennen und die Meinungen zu untersuchen. (vgl. Yessenov und Misailovic, 2009)

Intuitiv kann die Aufgabe, die Meinung eines Textes oder Satzes zu bestimmen, leicht erscheinen. Ein leicht nachvollziehbarer Ansatz ist die Einordnung einer Meinung durch eine Menge von Schlüsselwörtern, die auf positive oder negative Meinungen hinweisen. Eine Studie von Pang u. a. (2002) zur Untersuchung von Schwierigkeiten bei der Klassifikation von Meinungen zeigt aber, dass die Erstellung solcher Mengen von Schlüsselwörtern nicht so einfach wie angenommen ist. Es wird gezeigt, dass mit statistischen Mitteln erzeugte Listen bessere Resultate liefern können als per Hand erstellte. Die statistischen Regeln können auch Wörter liefern, die auf den ersten Blick nicht auf eine Meinung hindeuten, aber trotzdem das Ergebnis verbessern. Zum Beispiel ergab die Untersuchung von Pang u. a. (2002), dass ein Fragezeichen für die untersuchten Filmbewertungen eher auf eine negative als eine positive Meinung hindeutet. Dies kann daran erklärt werden, dass in negativen Bewertungen oft rhetorische Fragen, wie „Was hat sich sich der Regisseur dabei nur gedacht?“ gestellt wurden. Dies zeigt, dass eine Analyse des Textkorpus sich mehr lohnen kann als die intuitive Herangehensweise. (vgl. Pang u. a., 2002)

2.1 Ablauf der Verarbeitung

Der Prozess der Klassifikation von Dokumenten kann in mehrere Verarbeitungsschritte untergliedert werden. Diese Schritte finden sich in ähnlicher Form in vielen Untersuchungen oder Anwendungen zu *Sentiment Analysis*. (vgl. Dalal und Zaveri, 2011)

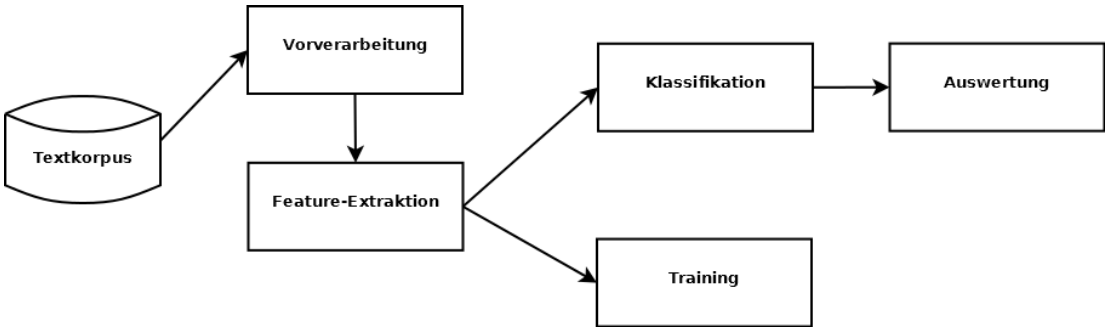


Abbildung 2.1: Allgemeiner Ablauf der Verarbeitung bei *Sentiment Analysis*

2.2 Textkorpus

Der Textkorpus ist eine Sammlung von Dokumenten, die untersucht werden sollen. Aus ihm können die Trainingsdaten und Daten zur Validierung entnommen werden. Untersuchungen des Textkorpus können helfen, die Analyse besser auf eine bestimmte Domäne anzupassen und damit die Ergebnisse zu verbessern.

In dieser Arbeit werden Filmbewertungen untersucht. Bewertungen können leicht in großer Anzahl im Internet gefunden werden. Es handelt sich um eine Sammlung von Filmbewertungen, bereitgestellt durch Pang und Lee. Insgesamt enthält der Korpus 1000 positive und 1000 negative Dokumente, die von einer Internet-Reviewdatenbank ausgelesen und eingeordnet wurden. [1] Dieser Textkorpus wurde schon von den Autoren und vielen Anderen in Untersuchungen verwendet und die einzelnen Dokumente sind bereits in Sätze aufgeteilt worden. Dies erleichtert die spätere Verarbeitung.

2.3 Vorverarbeitung

Der erste Schritt ist die Vorverarbeitung. Hier wird versucht, nicht relevante Teile des Textes herauszufiltern und die Repräsentation des Textes bei der Verarbeitung zu verbessern. Die Vorverarbeitung kann dabei sowohl Teile des Textes entfernen als auch Merkmale, wie zum Beispiel Informationen über Wortarten, hinzufügen. Die Vorverarbeitung umfasst Aufgaben wie die Erkennung von Satzgrenzen oder das Entfernen von Stoppwörtern. Das sind Wörter, zum Beispiel Artikel, die bei der Klassifizierung nicht benötigt werden. (vgl. Dalal und Zaveri,

[1] zu finden unter http://www.cs.cornell.edu/People/pabo/movie-review-data/

2011)

In der einfachsten Form kann so ein Vorverarbeitungsschritt dazu dienen, Sonderzeichen zu entfernen. Bei der Klassifikation werden oft die Struktur des Satzes und damit auch die Satzzeichen nicht mehr benötigt. Ein komplizierteres Beispiel für die Vorverarbeitung ist die Behandlung von Negationen. Negationen kehren die Polarität der Wörter in ihrem Wirkungsbereich um. Deshalb kann es zu besseren Ergebnissen führen, wenn vor der Klassifizierung negierte Wörter gekennzeichnet werden. Das Ergebnis kann auch durch das Entfernen ganzer Sätze bei der Vorverarbeitung verbessert werden. Ein Ansatz besteht darin, Sätze, die keine Meinungen, sondern nur Fakten enthalten, zu identifizieren und für die Klassifikation zu entfernen. Dies kann den Einfluss von Textpassagen, die keine Meinung enthalten, reduzieren und somit die Ergebnisse verbessern.

Diese und weitere Konzepte zur Vorverarbeitung werden in Kapitel 3 im Detail beschrieben und untersucht.

2.4 Feature-Extraktion

Für die maschinelle Klassifikation eines Dokuments müssen Merkmale gefunden werden, die bei der Klassifikation dazu benutzt werden können, das Dokument in eine der Klassen einzuordnen. Diese Merkmale werden als Features f_i und die Menge von Features eines Dokuments als Featurevektor $\vec{F} = (f_1, f_2, ..., f_n)$ bezeichnet. Die Erstellung eines Featurevektors aus einem Dokument wird Feature-Extraktion genannt. Der Wert eines Elements des Featurevektors kann unterschiedlich sein. Möglich ist zum Beispiel ein binärer Wert, der abbildet, ob das Feature im Text vorhanden ist oder nicht. Soll dargestellt werden, wie oft ein Feature vorkommt, kann auch ein Integer-Wert verwendet werden. Außerdem könnte man mit dem Wert die Bedeutung eines einzelnen Features modellieren. (vgl. Yessenov und Misailovic, 2009)

Die Wahl der Features ist ein wichtiger Teil der Textverarbeitung und kann das Ergebnis stark beeinflussen. Ein guter Featurevektor spiegelt den für die Fragestellung relevanten Teil des Dokuments möglichst gut wieder. So wird nur dieser Teil bei der Weiterverarbeitung betrachtet. Die Wahl der Features ist abhängig von der Fragestellung und der Domäne des Textes. Deshalb kann es für diese Aufgabe keinen einheitlichen Algorithmus geben. Die richtige Auswahl der Features kann mit Intuition, Wissen über die Domäne und Experimentieren gefunden werden. (vgl. Pang und Lee, 2008; Yessenov und Misailovic, 2009)

Um Features aus einem Dokument oder Satz zu bilden, wird eine Teilmenge der Wörter als Features genommen. Dieses Vorgehen wird als *Bag of Words*-Modell bezeichnet. Dabei wird davon ausgegangen, dass die einzelnen Features statistisch unabhängig voneinander sind. Dies ist zwar nicht korrekt, aber es vereinfacht die Verarbeitung, weil ein Feature unabhängig von seiner Position im Satz betrachtet werden kann. Statistische Abhängigkeiten zwischen Wörtern entstehen zum Beispiel durch grammatikalische Regeln der Sprache. Zum Beispiel folgt hinter einem Artikel meistens ein Nomen. Jedes Element in dem Featurevektor kann für ein Wort darstellen, ob es in dem Text vorhanden ist oder nicht oder anzeigen, wie oft dieses Wort im Text vorhanden ist. Die Anzahl der Elemente in dem Vektor wird also durch die Wahl der Features bestimmt. Wird jedes Wort als Feature verwendet, kann die Datenmenge sehr groß werden, deshalb kann es notwendig sein, Features auszuwählen. Eine Methode dafür ist, nur Features aus Wörtern zu bilden, die prozentual häufiger in den Texten auftreten als ein festgelegter Grenzwert. Dies entfernt vor allem Wörter, die nur sehr selten auftreten. (vgl. Yessenov und Misailovic, 2009)

Ein weit verbreiteter Ansatz ist es, n-Gramme zu bilden. Unigramme ($n = 1$) als Features zu nehmen bedeutet, dass jedes Feature genau aus einem Wort besteht. Bigramme ($n = 2$) bestehen aus Wortpaaren von zwei aufeinanderfolgenden Wörtern. Satzanfang und Satzende können hierbei dargestellt werden, indem ein Satzanfang- und Satzendezeichen mit in die n-Gramme einfließen. Je höher n gewählt wird, umso besser zeigt die Abbildung die Einordnung eines Wortes im Satz. Für Features, die aus n Wörtern bestehen, wird es bei großem n immer unwahrscheinlicher, ein weiteres Vorkommen in den Texten zu finden. Die Wahl des Wertes für n beeinflusst also die Ergebnisse und sollte abgewägt werden. Es ist auch möglich, Kombinationen verschiedener n-Gramme oder Kombinationen mit anderen Features zu verwenden.

„Dies ist ein Beispiel“ würde bei Bigrammen folgende Features ergeben:

- _ Dies
- Dies ist
- ist ein
- ein Beispiel
- Beispiel _

_ steht hierbei für den Beginn oder das Ende eines Satzes.

Für spezielle Anforderungen kann es sich positiv auswirken, nur bestimmte Wortarten als Features zu verwenden. Hierfür kann ein Part-of-Speech-Tagger, der jedem Wort seine Wortart zuordnet, eingesetzt werden. Bei der Analyse von Subjektivität zum Beispiel haben die Adjektive eine große Bedeutung. Hier könnte man einen Featurevektor aus allen Adjektiven im Dokument erstellen. (vgl. Pang und Lee, 2008)

2.5 Klassifikation

Ein Klassifikationsverfahren ordnet dem Betrachtungsgegenstand eine Klasse zu. Hierfür wird oft ein maschinelles Lernverfahren eingesetzt. Es gibt zwei Arten von Lernverfahren, die hierfür gut geeignet sind: Überwachtes und unüberwachtes Lernen. Beim überwachten Lernen wird der Klassifikator vor dem Einsatz trainiert. Für das Training werden Trainingsdaten benötigt. Diese bestehen meistens aus per Hand den einzelnen Klassen zugeordneten Daten und sollten den eigentlichen Daten möglichst ähnlich sein. Existieren keine Trainingsdaten, kann ein unüberwachtes Lernverfahren eingesetzt werden. Dabei können zum Beispiel ähnliche Dokumente zu einer Klasse gruppiert werden. Dies wird auch als Clustering bezeichnet. (vgl. Yessenov und Misailovic, 2009)

In dieser Arbeit wird der erste Ansatz, das überwachte Lernen, weiterverfolgt. Die hier relevanten Klassen für Dokumente heißen Positiv und Negativ, die für Sätze Objektiv und Subjektiv.

Zwei populäre Verfahren für maschinelles Lernen sind der Naive-Bayes-Klassifikator und Support-Vektor-Maschinen (SVM).

2.5.1 Naive-Bayes-Klassifikator

Der Naive-Bayes-Klassifikator arbeitet mit Wahrscheinlichkeitswerten. Der Klassifikator versucht, die Wahrscheinlichkeit für die Zugehörigkeit eines Textes zu jeder der Klassen zu bestimmen. Danach wird die Zuordnung mit der höchsten Wahrscheinlichkeit ausgewählt.

Die Wahrscheinlichkeit, dass ein Text mit dem Featurevektor x in die Klasse C gehört, wird durch den Ausdruck $P(C|x)$ dargestellt. $P(C|x)$ ist die bedingte Wahrscheinlichkeit dafür, dass ein Text mit dem Featurevektor x in die Klasse C gehört. Es handelt sich hierbei um

eine a-posteriori-Wahrscheinlichkeit, weil die Wahrscheinlichkeit für die Zuordnung zu einer Klasse berechnet wird, nachdem der Wert für x bestimmt wurde. (vgl. Alpaydin, 2008)

Um diese Wahrscheinlichkeit anhand der Featurevektoren aus den Trainingsdaten berechnen zu können, wird der Satz von Bayes verwendet.

$$P(C|x) = \frac{P(C)P(x|C)}{P(x)} \tag{2.1}$$

$P(x|C)$ ist die bedingte Wahrscheinlichkeit dafür, dass in einer festgelegten Klasse C der Featurevektor x auftritt, und wird als Klassen-Likelihood bezeichnet. Dieser Wert lässt sich anhand der Trainingsdaten berechnen. (vgl. Alpaydin, 2008)

Zur Vereinfachung der Berechnung kann angenommen werden, dass die Features des Textes statistisch voneinander unabhängig sind. Dadurch kann die Position des Features im Featurevektor ignoriert werden. Diese Annahme macht den Bayes-Klassifikator „naive“, weil sie nicht korrekt ist. Zum Beispiel ist es durch die Grammatikregeln der Sprache wahrscheinlich, dass nach einem Artikel ein Nomen folgt. Durch diese Annahme gilt $P(x|C) = \prod_{i \in x} P(f_i|C)$ und kann damit leicht bestimmt werden. $P(f|C)$ ergibt sich aus der Anzahl der Texte mit Feature f in Klasse C geteilt durch die Anzahl der Texte in Klasse C. (vgl. Ertel, 2009)

Außerdem werden die a-priori-Wahrscheinlichkeiten $P(C)$ und $P(x)$ benötigt. $P(C)$ ist das Verhältnis von der Anzahl der Texte in der Klasse C zu der Gesamtanzahl der Texte aller Klassen und $P(x)$ ist die Wahrscheinlichkeit dafür, dass der Featurevektor x in einer der Klassen auftritt. Unter der Voraussetzung, dass die Klassen disjunkt sind, gilt damit: $P(x) = \sum_{k=1}^{K} p(x|C_k)P(C_k)$. (vgl. Alpaydin, 2008)

$$P(C_i|x) = \frac{P(C_i)p(x|C_i)}{P(x)} = \frac{P(C_i)p(x|C_i)}{\sum_{k=1}^{K} p(x|C_k)P(C_k)} \tag{2.2}$$

Nach der Berechnung der Wahrscheinlichkeit $P(C_i|x)$ für alle Klassen C_i wird die Klasse mit der größten Wahrscheinlichkeit ausgewählt. Auf diese Weise wird die Zuordnung immer so gewählt, dass die Wahrscheinlichkeit für einen Fehler minimal ist. (vgl. Alpaydin, 2008)
Da der Nenner nicht von der Klasse C_i abhängt und damit konstant ist, kann er zur Vereinfachung bei der Berechnung des Maximums weggelassen werden. (vgl. Ertel, 2009)

Der Naive-Bayes-Klassifikator wird in der Praxis häufig eingesetzt, da er trotz seiner Ein-

fachheit und der falschen Annahme der statistischen Unabhängigkeit der Features sehr gute Ergebnisse liefern kann. Auch im Bereich der Textklassifikation, zum Beispiel bei Spam-Filtern wird der Klassifikator häufig verwendet. (vgl. Ertel, 2009)

2.5.2 Support-Vektor-Maschinen (SVM)

Support-Vektor-Maschinen versuchen anhand der Trainingsdaten, eine Funktion zu erlernen, die die Klassen möglichst gut voneinander trennt. Jeder Featurevektor wird als ein Punkt im Raum betrachtet. Die gesuchte Funktion definiert dann eine Ebene, die die Punkte der unterschiedlichen Klassen voneinander trennt. Die Trennebene in diesem mehrdimensionalen Raum wird als Hyperebene bezeichnet.

Soll ein neuer Text klassifiziert werden, wird nach der Vorverarbeitung zuerst der Featurevektor bestimmt. Dieser Vektor wird in ein Koordinatensystem mit der Trennebene eingezeichnet. Dann wird bestimmt, auf welcher Seite der Ebene der neue Punkt liegt und die Klasse dementsprechend zugeordnet.

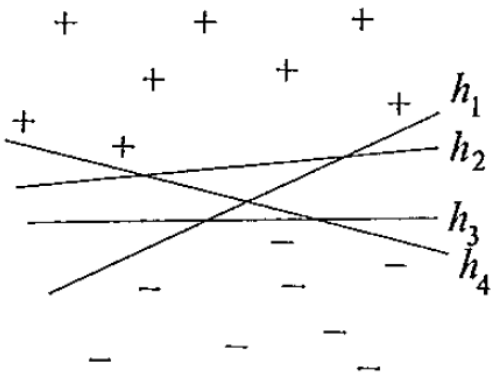


Abbildung 2.2: SVM - Koordinatensystem mit einigen möglichen linearen Diskriminanten (vgl. Görz u. a., 2003)

Die Abbildungen 2.2 und 2.3 zeigen ein einfacheres Beispiel mit zwei Klassen: Positiv und Negativ. Die Featurevektoren aus den Trainingsdaten sind als + und –, je nachdem zu welcher der beiden Klassen sie gehören, im Koordinatensystem eingezeichnet. In diesem Beispiel sind die beiden Klassen durch eine lineare Hyperebene trennbar. Es gibt viele mögliche Ebenen, die die beiden Klassen fehlerfrei trennen würden. Einige Beispiele sind in Abbildung 2.2 eingezeichnet. Für das Verfahren wird die optimale Trennebene gesucht. Diese ist definiert als die Ebene,

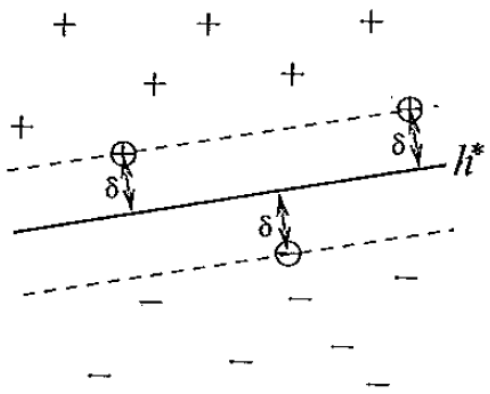


Abbildung 2.3: SVM - Koordinatensystem mit optimaler linearer Diskriminante und Stützvektoren (vgl. Görz u. a., 2003)

die die positiven und negativen Beispiele mit maximalem Abstand zu den Vektoren aus den Trainingsdaten trennt. Alle Punkte haben dann einen Mindestabstand δ von der Hyperebene. Durch diese Eigenschaft sind es allein die Punkte mit genau dem Abstand δ, die die Lage der Hyperebene definieren. Diese Vektoren werden Stützvektoren oder Support-Vektoren genannt. Alle anderen Vektoren könnten für die Bestimmung der Hyperebene weggelassen werden, ohne das Ergebnis zu beeinflussen. Das Koordinatensystem in Abbildung 2.3 zeigt die optimale Hyperebene und ihre Stützvektoren. (vgl. Görz u. a., 2003)

Dies bedeutet, dass die Klassifikation nur von den Vektoren an der Grenze der Klasse abhängig ist. Das macht eine Überanpassung an die Trainingsdaten unwahrscheinlicher. (vgl. Ertel, 2009)

Für nicht linear trennbare Klassen muss das Verfahren angepasst werden. Dazu werden die Featurevektoren durch eine nicht lineare Funktion in einen höher dimensionalen Raum abgebildet. Mit diesen so berechneten Vektoren wird eine lineare Hyperebene bestimmt. Das Verfahren zur Bestimmung der optimalen Hyperebene ist also identisch. Obwohl die Hyperebene auf den angepassten Daten linear ist, ergibt sich für die Eingabevektoren, den Featurevektoren der Trainingsdaten, eine beliebig komplexe Trennebene. (vgl. Görz u. a., 2003)

Im allgemeinen Fall ist eine solche Abbildung in einen höher dimensionalen Raum, ineffizient zu berechnen. Bei Support-Vektor-Maschinen wurde jedoch eine besondere Eigenschaft

gefunden, die die Berechnung sehr effizient macht. Dazu werden sogenannte Kernfunktionen benutzt, die die Berechnung des Skalarprodukts auf den angepassten Vektoren beim Trainieren und Klassifizieren ersetzt. Durch diese Kernfunktionen kann die explizite Berechnung der Abbildung umgangen werden. Der Einsatz leicht austauschbarer Kernfunktionen macht das Verfahren sehr anpassungsfähig. In der Praxis gibt es verschiedene Standard-Kernfunktionen, wie zum Beispiel für Polynome beliebigen Grades. (vgl. Görz u. a., 2003; Alpaydin, 2008)

Support-Vektor-Maschinen sind auch gut für Klassifikationsaufgaben geeignet, bei denen die Anzahl der Features sehr groß ist. Deshalb eignet sich das Verfahren auch gut für die Klassifikation von Texten. (vgl. Görz u. a., 2003)

2.6 Training

Für das Training des Verfahrens werden bereits eingeordnete Dokumente benötigt. Diese Einordnung kann per Hand gemacht werden. Zum Beispiel würde man den Filmbewertungen, die in dieser Arbeit untersucht werden, jeweils die Klasse Positiv oder Negativ zuordnen.

Der Ablauf des Trainings ist sehr ähnlich zum Ablauf der Klassifikation. Vorverarbeitung und Feature-Extraktion sind identisch. Danach werden dem Verfahren die Featurevektoren aus den Dokumenten und die dazugehörige Klasseneinordnung übergeben. So kann das Auftauchen von bestimmten Features in einer Klasse gespeichert und dieses Wissen später eingesetzt werden. Die Verfahren haben unterschiedliche Vorgehensweisen für die Generierung dieses Wissens und unterschiedliche Arten der Repräsentation oder Speicherung. Eine Möglichkeit ist zum Beispiel zu speichern, wie oft ein Feature insgesamt und wie oft es in einem Dokument einer bestimmten Klasse vorkommt. Dies wäre ein Ansatz mit dem später durch statistische Auswertungen und Wahrscheinlichkeitswerte klassifiziert werden kann, ähnlich wie es beim Naive-Bayes-Klassifikator der Fall ist.

2.7 Ergebnisse und Auswertung

Als Ergebnis ordnet eine Klassifikation jedem Dokument eine der Klassen zu. Um untersuchen zu können, wie gut das Verfahren arbeitet, werden weitere vorsortierte Daten benötigt. Es bietet sich an, hierfür den Teil des Textkorpus zu nehmen, der nicht in die Trainingsdaten eingeflossen ist. Die Bewertung geschieht statistisch. Es wird gezählt, wie viele Dokumente das trainierte Verfahren richtig und wie viele falsch zuordnet.

Bewertungsverfahren werden benötigt, um Verfahren zu vergleichen. So ist es möglich, Verbesserungen an einem Verfahren zu erkennen oder dieses Verfahren mit anderen zu vergleichen. Verschiedene Bewertungsverfahren können die Qualität eines Verfahrens unterschiedlich gut beschreiben und anhand von unterschiedlichen Eigenschaften bewerten.

Häufig für die Bewertung eingesetzte Werte sind Precision, Recall und der F-Score. Die Berechnung dieser Werte ist vor allem dann sehr einfach, wenn es nur zwei Klassen gibt und jeder Klassifikationsfehler gleich bewertet wird. Ist dies der Fall, kann jedes Ergebnis des Klassifikators, abhängig von der wahren Klasse, der Klasse aus den Trainingsdaten, in eine von vier Kategorien eingeteilt werden. In Tabelle 2.1 ist ein Beispiel für die Klassen Positiv und Negativ dargestellt. Die Kategorien „wahres Positiv“ und „wahres Negativ“ beinhalten richtige Ergebnisse des Klassifikators. Der Unterschied besteht nur darin, ob die Klasse Positiv oder Negativ richtig zugeordnet wurde. Genau so verhält es sich mit den Kategorien „falsch Negativ“ und „falsch Positiv“, abhängig davon, wie die falsch vorhergesagte Klasse lautet. Bei der Verifikation eines Verfahrens werden die Zuweisungen der einzelnen Kategorien gezählt und die Ergebnisse können unter anderem zu den folgenden Werten kombiniert werden. (vgl. Alpaydin, 2008)

	vorhergesagte Klasse	
wahre Klasse	Positiv	Negativ
Positiv	wp: wahres Positiv	fn: falsches Negativ
Negativ	fp: falsches Positiv	wn: wahres Negativ

Tabelle 2.1: Kategorien für die Zuordnung von Dokumenten durch einen Klassifikator bei der Bewertung (vgl. Alpaydin, 2008)

2.7.1 Precision

Der Precision-Wert beschreibt, wie korrekt ein Verfahren arbeitet. Er beschreibt das Verhältnis zwischen der Anzahl der korrekt zugeordneten Dokumente zu der Gesamtanzahl der vom Verfahren in eine Klasse eingeordneten Dokumente.

$$precision = \frac{wp}{wp + fp} \tag{2.3}$$

Ein Precision-Wert von 100% zeigt an, dass alle Zuordnungen von Dokumenten zu der positiven Klasse korrekt sind. Zuordnungen zu anderen Klassen werden nicht betrachtet. Wird einer Klasse nur ein Dokument zugeordnet und ist diese Zuordnung korrekt, ist ein Wert von 100% schon erreicht.

2.7.2 Recall

Der Recall-Wert beschreibt das Verhältnis von korrekt klassifizierten positiven Dokumenten zu der Gesamtanzahl der Dokumente in dieser Klasse. Der Wert zeigt also an, wie vollständig ein Verfahren arbeitet und die Klasse Positiv erkennt.

$$recall = \frac{wp}{wp + fn} \tag{2.4}$$

Ein Recall-Wert von 100% wird erreicht, wenn alle Dokumente der Klasse Positiv korrekt klassifiziert wurden.

2.7.3 F-Score

Precision und Recall sind einzeln nicht sehr aussagekräftig. Es ist einfach, die Ergebnisse von einem Wert auf Kosten des anderen zu verbessern. Dazu kann zum Beispiel das Verfahren durch einen Schwellenwert für die Zuweisung zu einer Klasse ergänzt werden.

Der F-Score ist das harmonische Mittel von Precision und Recall. Durch die Kombination erhöht sich die Aussagekraft der Bewertung, weil dadurch sowohl Verfahren mit schlechtem Recall-Wert als auch mit schlechtem Precision-Wert schlecht bewertet werden.

$$F = 2 * \frac{precision * recall}{precision + recall} \tag{2.5}$$

3 Konzepte zur Vorverarbeitung

3.1 Subjektivitätsanalyse

Für die Klassifikation der Meinung eines Textes kann es hilfreich sein zu wissen, wo genau, zum Beispiel in welchen Sätzen, die Meinung geäußert wird oder ob überhaupt eine Meinung in dem Text ausgedrückt wird. Verfahren zur Subjektivitätsanalyse versuchen, dies herauszufinden.

Die hier vorgestellten Verfahren können allgemein zur Erkennung von Subjektivität und auch Objektivität eingesetzt werden. Im *Information Retrieval* werden diese Verfahren ebenfalls häufig eingesetzt, um zum Beispiel beim Suchen von Meinungen, diese zu identifizieren oder um Fakten für eine automatische Textzusammenfassung zu finden. Diese Arbeit wird sich nur mit der Verwendung der Subjektivitätsanalyse in der Vorverarbeitung für die Klassifikation beschäftigen.

Bei der Vorverarbeitung für eine Klassifikation von Meinungen wird oft eine Subjektivitätsanalyse auf Satzebene durchgeführt. Die Klassifikation soll entscheiden, ob ein Dokument eine positive oder eine negative Meinung ausdrückt. Es wird vorausgesetzt, dass eine Meinung in dem Dokument existiert. Um die Untersuchung zu verbessern, wird in der Vorverarbeitung versucht herauszufinden, welche Teile des Dokuments Meinungen enthalten und welche nicht. Dies wird auf der nächst kleineren zusammenhängenden Einheit, den Sätzen, gemacht.

Sätze können entweder subjektiv oder objektiv sein. Subjektive Sätze enthalten persönliche Meinungen, Gefühle oder Überzeugungen und objektive Sätze bestehen aus Fakten. Explizite Meinungen sind Meinungen über ein Objekt in einem subjektiven Satz. Auch objektive Sätze können Meinungen enthalten, diese werden als implizite Meinungen bezeichnet.

Abbildung 3.1 zeigt zwei Beispielsätze. Satz 1 ist ein subjektiver Satz. Der Autor drückt hiermit seine Meinung zu der Sprachqualität des Telefons aus. Die Meinung ist explizit und

bezieht sich auf die Sprachqualität. Im Gegensatz dazu ist Satz 2 objektiv. Der Satz beschreibt eine Tatsache und keine Meinung. Dennoch lässt sich eine implizite Meinung erkennen. Es ist zu vermuten, dass der Autor die kurze Funktionsdauer seines Ohrhörers als negativ empfindet. Objektive Sätze mit impliziter Meinung enthalten oft den Grund oder die Ursache für diese Meinung. (vgl. Liu, 2010)

1. „Die Sprachqualität dieses Telefons ist erstaunlich“
2. „Der Ohrhörer ging innerhalb von zwei Tagen kaputt“

Abbildung 3.1: Beispielsätze für die Erklärung von Subjektivität (vgl. Liu, 2010)

Meinungen können in unterschiedlicher Form in einem Satz enthalten sein. Beispiele hierfür sind Wünsche, Überzeugungen, Verdachte und Erwartungen.

Filmreviews enthalten oft neben einer Bewertung des Films viele andere Informationen, die für die Klassifikation nicht hilfreich oder sogar hinderlich sein können. Die Filmreviews aus den hier verwendeten Trainingsdaten enthalten oft eine Zusammenfassung der Handlung und eine Auflistung einiger oder aller bekannten Schauspieler, die in dem Film mitwirken. Diese Fakten enthalten meist keine Meinungen über den Film. Indirekte Meinungen in Fakten sind schwierig, richtig zu erkennen. Deswegen kann es die Ergebnisse beim Klassifizieren verbessern, wenn die Fakten erkannt und herausgefiltert werden.

Im Folgenden werden verschiedene Ansätze zur Klassifikation von Subjektivität erläutert.

3.1.1 Lexika und Datenbanken für die Subjektivitätsanalyse

Lexika für die Subjektivitätsanalyse enthalten eine Liste von Wörtern mit zusätzlichen Informationen, zum Beispiel dazu, wie stark sie auf Subjektivität beziehungsweise Objektivität hindeuten. Lexika können per Hand oder automatisch, zum Beispiel mit Klassifikatoren, erstellt werden. Bei der Textklassifikation kann dann auf die Informationen in einem Lexikon zurückgegriffen werden.

OpinionFinder-Lexikon

Eines der meist benutzten Lexika ist das Lexikon, das mit der *OpinionFinder*-Software ausgeliefert wird. *OpinionFinder* ist ein Programm, welches subjektive Sätze auffindet und bestimmte Merkmale in dem Satz markiert. Dazu zählen zum Beispiel der Autor der Meinung und Wörter,

die die Meinung positiv oder negativ beeinflussen. Diese Informationen können von anderen Programmen benutzt werden, die Informationen über Subjektivität in Texten benötigen. Dies können zum Beispiel Programme zum automatischen Beantworten von Fragen über Meinungen in einem Text sein. Außerdem können *Information Extraction*-Systeme das Programm nutzen, um Fakten herauszufiltern. Das Programm verwendet einen Naive-Bayes-Klassifikator, der Features aus dem Lexikon benutzt. (vgl. Wilson u. a., 2005)

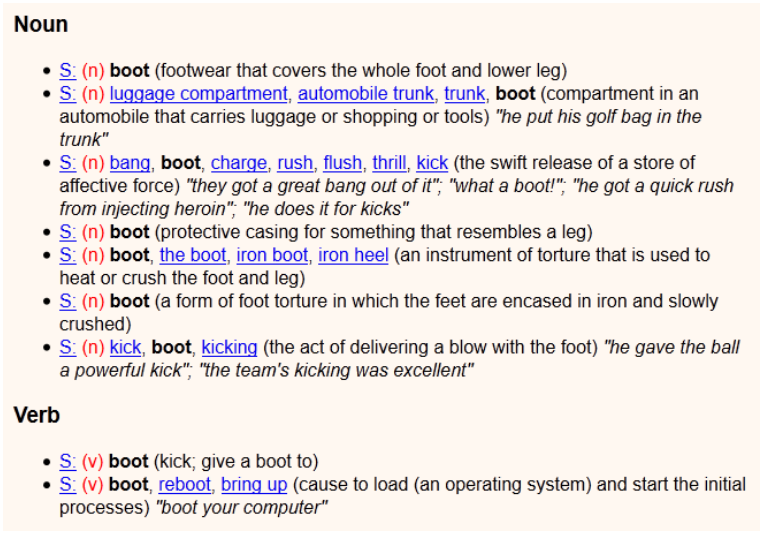
Die Trainingsdaten für den Klassifikator wurden nicht ausschließlich per Hand erstellt, sondern auch automatisch durch einen regelbasierten Klassifikator. Regelbasierte Klassifikatoren haben meist einen hohen Precision- und einen niedrigen Recall-Wert. Dies liegt daran, dass durch die implementierten Regeln vergleichsweise sicher auf Subjektivität oder Objektivität geschlossen werden kann, aber Sätze, auf die diese Regeln zutreffen, in den Daten selten auftreten. Deshalb eignen sich regelbasierte Verfahren gut, um Trainingsdaten für andere Klassifikatoren zu erstellen. Ein Beispiel für einen regelbasierten Klassifikator ist ein Verfahren, welches jedem Satz, in dem zwei oder mehr stark subjektive Wörter enthalten sind, die Klasse Subjektiv zuordnet. Wie stark subjektiv ein Wort ist, kann in einem Lexikon abgelesen werden. Als objektiv könnte ein Satz klassifiziert werden, wenn er kein stark subjektives Wort enthält und zusammen mit den beiden angrenzenden Sätzen nicht mehr als zwei schwach subjektive Wörter beinhaltet. Alle anderen Sätze werden nicht eingeordnet. An dem Beispiel lässt sich erkennen, dass die Regeln nur eine Klasse bestimmen, wenn die Zuordnung sehr sicher ist. Es werden viele Sätze nicht klassifiziert, deswegen ist der Recall-Wert sehr niedrig, dafür ist die Klassifikation eines Satzes mit hoher Wahrscheinlichkeit korrekt, was den Precision-Wert erhöht. (vgl. Wilson u. a., 2005; Banea u. a., 2011)

WordNet

WordNet ist eine von dem *Cognitive Science Laboratory* der *Princeton University* zusammengestellte Datenbank, die semantische und lexikalische Beziehungen zwischen Wörtern enthält. Zu jedem Wort enthält die Datenbank unter anderem Informationen über die Wortart, die verschiedenen möglichen Bedeutungen mit Beispielsätzen und Worterklärungen. Synsets bilden eine Sammlung von Wörter gleicher Begrifflichkeit. Sie können sowohl semantische Relationen, wie zum Beispiel Synonyme, als auch lexikalische Relationen wie Ober- und Unterbegriffe enthalten. Durch die Kombination von lexikalischen und semantischen Informationen zu jedem Wort, kann *WordNet* sowohl als einfaches Lexikon als auch als eine Art Thesaurus verwendet werden. (vgl. Fellbaum, 1998)

Abbildung 3.2 zeigt einen Beispieleintrag aus der *WordNet*-Datenbank für das englische Wort „boot“. Es ist gut zu erkennen, dass dieses Wort sowohl als Nomen als auch als Verb benutzt werden kann. In beiden Fällen kann das Wort unterschiedliche Bedeutungen haben. Als Nomen kann es zum Beispiel für einen „Tritt“ oder einen „Schuh“ stehen und als Verb kann es „treten“ oder „hochfahren eines Betriebsystems“ bedeuten. Die blauen Links nach dem Wort führen zu verschiedenen Synonymen. Dahinter steht in Klammern eine Erklärung der Wortbedeutung, gefolgt von Beispielen für die Verwendung.

WordNet und die dazugehörige Software sind frei verfügbar und werden in der Forschung oft verwendet.



Noun

- S: (n) **boot** (footwear that covers the whole foot and lower leg)
- S: (n) luggage compartment, automobile trunk, trunk, **boot** (compartment in an automobile that carries luggage or shopping or tools) *"he put his golf bag in the trunk"*
- S: (n) bang, **boot**, charge, rush, flush, thrill, kick (the swift release of a store of affective force) *"they got a great bang out of it"; "what a boot!"; "he got a quick rush from injecting heroin"; "he does it for kicks"*
- S: (n) **boot** (protective casing for something that resembles a leg)
- S: (n) **boot**, the boot, iron boot, iron heel (an instrument of torture that is used to heat or crush the foot and leg)
- S: (n) **boot** (a form of foot torture in which the feet are encased in iron and slowly crushed)
- S: (n) kick, **boot**, kicking (the act of delivering a blow with the foot) *"he gave the ball a powerful kick"; "the team's kicking was excellent"*

Verb

- S: (v) **boot** (kick; give a boot to)
- S: (v) **boot**, reboot, bring up (cause to load (an operating system) and start the initial processes) *"boot your computer"*

Abbildung 3.2: *WordNet*-Beispieleintrag für das englische Wort „boot“ (Princeton University, 2010)

SentiWordNet

Ein weiteres häufig benutztes Lexikon ist *SentiWordNet*. Es stellt eine Erweiterung der Informationen von *WordNet* dar. *SentiWordNet* enthält Informationen zu den verschiedenen Wortbedeutungen der Wörter, die zur Unterscheidung von positivem und negativem als auch von subjektivem und objektivem Text eingesetzt werden können. Jede Wortbedeutung aus *WordNet* wird mit zusätzlichen Informationen, den Werten P, N und O angereichert. P und N beschreiben, wie stark ein Wort auf die Klassen Positiv und Negativ hindeutet. O beschreibt dies für Objektivität. Alle Werte liegen im Bereich von $0,0$ bis $1,0$ und ergeben in der Summe

1. Subjektivität wird also durch die Summe der Werte P und N ausgedrückt. (vgl. Esuli und Sebastiani, 2006)

SentiWordNet wurde durch mehrere ternäre Klassifikatoren für *WordNet*-Synsets erstellt. Die Ausgabeklassen entsprechen P, N und O. Ein Wert von 1 wird erreicht, wenn alle Klassifikatoren dasselbe Ergebnis erzielen. Ist dies nicht der Fall, zeigt der Wert, wie viel Prozent der Klassifikatoren dieses Ergebnis geliefert haben. Trainiert wurden die Klassifikatoren mit Trainingsdaten, die nur zu einem kleinen Teil per Hand in die Klassen Positiv und Negativ eingeordnet wurden. Die beiden Klassen wurden dann in einem iterativen Prozess durch *WordNet*-Relationen erweitert. Die Menge der in die Klasse Objektiv eingeordneten Wörter ergab sich aus allen Wörtern, die nicht als positiv oder negativ klassifiziert wurden. Da der Vorgang selbst an der Erstellung der Trainingsdaten beteiligt ist, wird dieses Verfahren semi-überwachtes Lernen genannt. (vgl. Esuli und Sebastiani, 2006)

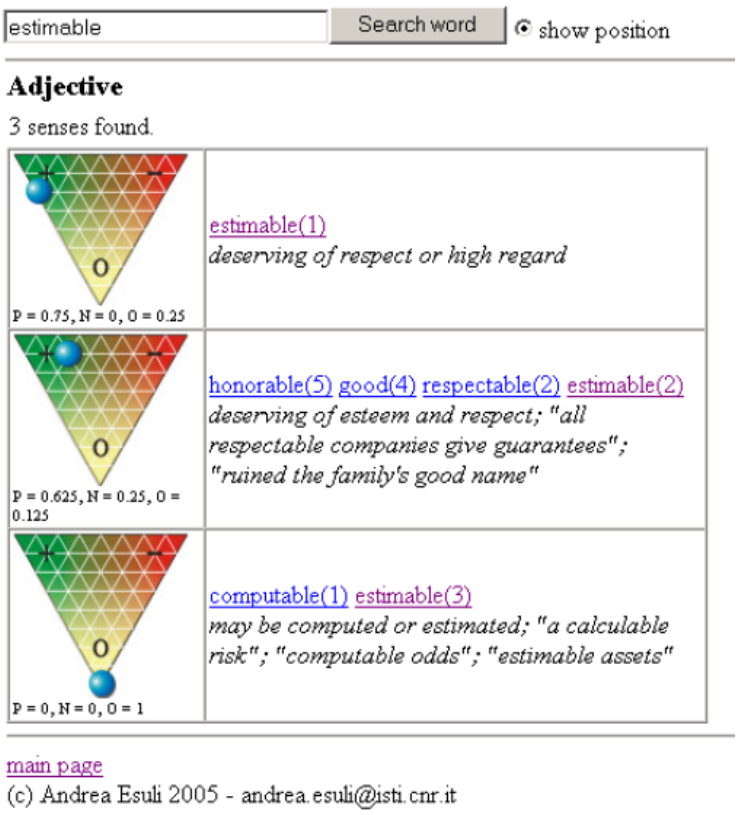


Abbildung 3.3: *SentiWordNet*-Beispieleintrag (Esuli und Sebastiani, 2006)

Abbildung 3.3 zeigt einen Beispieleintrag aus der *SentiWordNet*-Datenbank. Die einzelnen Zeilen zeigen verschiedene mögliche Bedeutungen von „estimable“ zusammen mit einer Erklärung, Wörtern gleicher oder sehr ähnlicher Bedeutung und teilweise Beispielen. Diese Informationen stammen aus *WordNet*. Die angereicherten Informationen werden hier in Form eines Dreiecks dargestellt. Jede Ecke steht für einen der Werte P, N und O. An der Position des blauen Punktes und den Angaben unter dem Dreieck können die genauen Werte abgelesen werden. In diesem Fall kann das Wort nur als Adjektiv in einem Satz stehen. Für den Fall, dass ein Wort Wortbedeutungen hat, die in unterschiedlichen Wortarten im Satz stehen können, werden diese Bedeutungen getrennt aufgeführt.

3.1.2 Adjektive und Subjektivität

Mit Adjektiven wird in der Sprache ausgedrückt, wie etwas ist. Es ist naheliegend, einen Zusammenhang von Subjektivität, der Anzahl und der Wahl der Adjektive zu vermuten. Untersuchungen zeigen, dass ein statistischer Zusammenhang zwischen dem Auftreten von Adjektiven und der Subjektivität eines Satzes besteht.

In einer Untersuchung von Bruce und Wiebe (1999) wurden zunächst von mehreren voneinander unabhängigen Personen manuell Sätze als subjektiv oder objektiv klassifiziert. Danach wurden Übereinkünfte bei der Klassifikation statistisch untersucht und den Sätzen, basierend auf dem Ergebnis, eine endgültige Klasse zugeordnet. An diesen so erstellten Trainingsdaten ist zu erkennen, dass Adjektive deutlich öfter in subjektiven Sätzen vorkommen. Das heißt, dass es für eine Untersuchung auf Subjektivität sinnvoll ist, das Auftreten von Adjektiven zu betrachten. Außerdem wurde gezeigt, dass verschiedene Arten von Adjektiven, zum Beispiel dynamische Adjektive, unterschiedlich gut auf Subjektivität hindeuten. Die verschiedenen Eigenschaften von Adjektiven werden im Folgenden genauer beschrieben.

Hatzivassiloglou und Wiebe (2000) untersuchten weitere Eigenschaften von Adjektiven, um die Zusammenhänge zwischen Adjektiven und Subjektivität besser nutzen zu können. Zwei Eigenschaften sind die semantische Orientierung und die Steigerbarkeit der Adjektive.

Getestet wurde ein sehr einfacher Klassifikator, der Sätze als subjektiv deklariert, sobald sich ein oder mehrere Wörter aus dem Satz in der Menge S befinden. Die Menge S kann aus verschiedenen Teilmengen der Adjektive bestehen und zum Beispiel die steigerbaren und positiven Adjektive enthalten. Die Ergebnisse der Untersuchung für verschiedene Mengen S zeigen, dass die Klassifikation mit diesen zusätzlichen Informationen im Vergleich zu der

Klassifikation nur auf Basis von Adjektiven bessere Ergebnisse liefert. Außerdem ist zu erkennen, dass eine Klassifikation mit den automatisch erstellten Adjektiven und den bestimmten Eigenschaften meistens gleiche oder sogar bessere Ergebnisse liefert als eine Klassifikation mit per Hand zugeordneten Daten. (vgl. Hatzivassiloglou und Wiebe, 2000)

Dynamische Adjektive

Statische Adjektive beschreiben sich nicht ändernde Zustände. Dynamische Adjektive können potenziell von ihrem Bezugsobjekt beeinflusst werden und die beschriebenen Zustände können sich über die Zeit verändern. Ob ein Adjektiv sich dynamisch oder statisch benutzen lässt, kann syntaktisch getestet werden. Kann ein Adjektiv mit dem Imperativ und *Progressing Aspect* benutzt werden, ist es dynamisch verwendbar. Der *Progressing Aspect* drückt im Englischen aus, dass eine Handlung nicht abgeschlossen ist und hat keine direkte Entsprechung im Deutschen. Ein Beispiel für ein statisches Adjektiv ist „tall“. Es kann nicht in Sätzen wie „He’s being tall“ oder „Be tall“ stehen und ist damit statisch. Die Sätze „He’s being careful“ und „Be careful“ zeigen, dass „careful“ ein dynamisches Adjektiv ist. (vgl. Quirk u. a., 1985)

Viele Adjektive mit statischer Bedeutung sind trotzdem dynamisch nutzbar. Ein weiterer Test kann helfen, diese Adjektive mit statischer Bedeutung zu identifizieren, obwohl sie dynamisch verwendet werden können. Dazu wird geschaut, ob die Adjektive bei dynamischer Verwendung eine dynamische Bedeutung haben und diese im Satz vor einem Nomen beibehalten. Der Vergleich von dem Satz „He’s being important“ mit „the important man“ weist darauf hin, dass „important“ sich dynamisch verwenden lässt, aber eine eher statische Bedeutung hat. (vgl. Bruce und Wiebe, 1999)
Alle dynamischen und die meisten statischen Adjektive sind steigerbar. (vgl. Quirk u. a., 1985)

Die Ergebnisse von Bruce und Wiebe (1999) zeigen, dass dynamische Adjektive bessere Indikatoren für Subjektivität sind als die Menge aller Adjektive.

Semantische Orientierung

Die semantische Orientierung oder Polarität eines Wortes beschreibt, ob seine Bedeutung positiv oder negativ ist. Wörter, die einen gewünschten Zustand beschreiben, haben eine positive Orientierung und Wörter, die einen ungewünschten Zustand beschreiben, eine negative. Bei Adjektiven kann die semantische Orientierung als die Fähigkeit betrachtet werden, einem Bezugsobjekt eine positive oder negative Eigenschaft zuzuweisen, die ohne das Adjektiv nicht vorhanden wäre. (vgl. Hatzivassiloglou und Wiebe, 2000)

Verschiedene Adjektive können unterschiedlich stark positiv oder negativ oder auch neutral sein. Adjektive wie „übel“ und „schmutzig“ sind zum Beispiel stark negativ. Das Wort „wunderschön“ ist stärker positiv als „schön“. Die Polarität eines Adjektivs ist abhängig vom Kontext und der Domäne des Textes. Negationen und Ironie zum Beispiel können die Polarität von Wörtern umkehren. Wörter können in einer bestimmten Domäne auf eine positive Meinung hindeuten und in einer anderen negativ gesehen werden. Zum Beispiel kann das Adjektiv „groß“ bei Handys auf eine negative und bei Häusern auf eine positive Meinung hindeuten.

Die semantische Orientierung von Adjektiven kann mit einem Verfahren von Hatzivassiloglou und McKeown (1997) automatisch bestimmt werden. Dazu wurden durch Konjunktionen verbundene Adjektive in einem großen Textkorpus, bestehend aus Artikeln des *World Street Journals* mit einer Gesamtlänge von etwa 21 Millionen Wörtern, mit einem Parser gesucht. Adjektive, die durch eine Konjunktion verbunden sind, haben meistens, abhängig von der Wahl der Konjunktion, dieselbe oder entgegengerichtete Polarität. Ein Beispiel hierfür wäre „brutal und korrupt“. Basierend auf diesen Informationen wurde ein Klassifikator erstellt, der mit einer Genauigkeit von 82% bestimmt, ob zwei Adjektive die gleiche semantische Orientierung haben. Die beiden so erstellten Klassen wurden per Hand zugeordnet. Dazu wurden den am häufigsten zusammen aufgetretenen Adjektiven manuell eine semantische Orientierung zugeordnet.

Steigerbarkeit

Adjektive können gesteigert werden. Diese Eigenschaft wird als *Gradability* bezeichnet. Steigerungen von Adjektiven ermöglichen es zu vergleichen. Steigerbare Adjektive können Bedeutungen unterschiedlich stark, abhängig vom Bezugsobjekt, stärken oder abschwächen. Zum Beispiel ist ein kleiner Planet immer noch deutlich größer als ein großes Haus. Die Größe der Verstärkung ist also relativ zum Objekt. Diese Abhängigkeit weist darauf hin, dass Steigerbarkeit ein guter Indikator für Subjektivität sein könnte. Ein Beispiel für ein nicht steigerbares Adjektiv ist „zivil“. (vgl. Hatzivassiloglou und Wiebe, 2000)

Um automatisch bestimmen zu können, ob ein Adjektiv steigerbar ist, wird gezählt, wie oft ein Adjektiv in der Grundform, dem Positiv, und wie oft es gesteigert, also im Komparativ oder Superlativ, im Text vorkommt. Außerdem wird für jedes Adjektiv bestimmt, wie oft es zusammen mit steigernden Wörtern, wie zum Beispiel „sehr“, auftritt. Für jedes Adjektiv ergeben sich so vier Zahlen, die für die Klassifikation in steigerbar und nicht steigerbar

verwendet werden können. Der erstellte Klassifikator erreichte bei einem Test mit einem Textkorpus, bestehend aus Artikeln des *World Street Journals*, eine Genauigkeit von über 85%. (vgl. Hatzivassiloglou und Wiebe, 2000)

3.1.3 Klassifikation durch überwachtes Lernen

Ein weiterer Ansatz für die Klassifikation von Subjektivität und Objektivität ist die Verwendung von überwachtem Lernen. Die Klassifikation durch die Lernverfahren wird auf genau dieselbe Art durchgeführt wie bei Dokumenten. Mit dem Unterschied, dass mit bereits als subjektiv oder objektiv markierten Sätzen trainiert wird und Sätze statt Dokumente klassifiziert werden.

Eine Untersuchung zu diesem Thema wurde von Wiebe u. a. (1999) durchgeführt. Hier werden weiterverarbeitete Ergebnisse der manuellen Klassifikation dazu benutzt, einen automatischen Klassifikator zu erstellen. Verwendet wurde ein Naive-Bayes-Klassifikator, der unter anderem mit binären Features arbeitet, die anzeigen, ob bestimmte Wortarten, wie Pronomen, Adjektive und Adverbien, in dem Satz vorhanden sind. Weiterhin wurde ein Feature erstellt, welches anzeigt, ob ein Satz einen neuen Textabschnitt beginnt. Außerdem wurde ein Wert benutzt, der beschreibt, wie oft ein Wort in dem Satz in einer der Klassen beim Training vorgekommen ist. Der Klassifikator erzielte auf dem eingesetzten Textkorpus eine durchschnittliche Genauigkeit von über 70%.

Yu und Hatzivassiloglou (2003) benutzten auch einen Naive-Bayes-Klassifikator. Trainiert wurde der Klassifikator mit Features aus Unigrammen, Bigrammen, Trigrammen und Informationen eines Part-of-Speech-Taggers. Zusätzlich wurde die Anzahl der positiven und negativen Wörter und Sequenzen von aufeinanderfolgenden Wörtern gleicher Polarität jedes Satzes als Feature verwendet. Die Polarität von Wortgruppen wird ausgehend von dem Ergebnis von Hatzivassiloglou und Wiebe (2000), dass positive oder negative semantische Orientierungen ein Indikator für Subjektivität sind, verwendet. Weiterhin wurden Features hinzugefügt, die die Polarität von bestimmten Wortarten und wichtigen Satzteilen repräsentieren. Zum Beispiel die semantische Orientierung des Hauptverbs oder des Subjekts. Zusätzlich wurde ein Feature aus der durchschnittlichen semantischen Orientierung der Wörter im Satz gebildet. Zur Untersuchung wurden verschiedene Kombinationen dieser Featuremengen getestet und verglichen. Test- und Trainingsdaten wurden aus einem Textkorpus mit 1,7 Millionen Artikeln verschiedener Nachrichtenagenturen, unter anderem dem *World Street Journal*, entnommen. Die Auswertung der Ergebnisse des Naive-Bayes-Klassifikators ergaben Precision- und Recall-Werte

von 80% bis 90% für die Klasse Subjektiv. Für die Klasse Objektiv ergaben sich Precision- und Recall-Werte im Bereich von 50%. Die n-Gramme hatten kaum positive Auswirkungen auf die Ergebnisse für die Klasse Subjektiv. Bei der Klasse Objektiv gab es einen deutlich erkennbaren Effekt im Vergleich zu den Ergebnissen der Untersuchung von Wiebe u. a. (1999). Allgemein ist zu erkennen, dass zusätzliche Informationen - in Form von zusätzlichen Features - die Ergebnisse der Klassifikation eher verbessern als verschlechtern. (vgl. Yu und Hatzivassiloglou, 2003)

3.1.4 Wortbedeutung und Subjektivität

Ein weiterer Ansatz ist der Versuch von der Wortbedeutung auf Subjektivität oder Objektivität zu schließen. Wörter können abhängig vom Kontext im Satz unterschiedliche Bedeutungen haben. Ein Beispiel ist das Wort „boot“. Es kann sowohl als Verb als auch als Nomen gebraucht werden und verschiedene Bedeutungen haben. Als Nomen kann es zum Beispiel „Schuh“ oder „Tritt“ bedeuten. Abbildung 3.2 zeigt einen *WordNet*-Beispieleintrag für dieses Wort, welcher alle Bedeutungen auflistet.

Die Idee ist es, Informationen über die Wortbedeutung jedes Wortes in einem Satz zu benutzen, um die Klassifikation durch diese zusätzlichen Informationen zu verbessern. Verfahren, die nicht zwischen verschiedenen Wortbedeutungen unterscheiden, sondern nur mit Wörtern arbeiten, klassifizieren einen Satz mit subjektiven Wörtern, die in einer objektiven Bedeutung stehen, falsch. Dies stellt eine signifikante Fehlerquelle dar. Mit der Unterscheidung von Wortbedeutungen wird versucht, dieses Problem zu beheben. (vgl. Akkaya u. a., 2009)

Die möglichen Bedeutungen von Wörtern müssen manuell festgelegt werden oder können einer bereits vorhandenen Datenbank entnommen werden. Eine Möglichkeit ist es *WordNet* (Kapitel 3.1.1) zu benutzen.

Jede mögliche Bedeutung eines Wortes wird unabhängig voneinander einer der zwei Klassen Subjektiv oder Objektiv zugeordnet. Die Bedeutung eines Wortes soll genau dann in die Klasse Subjektiv eingeordnet werden, wenn die Verwendung des Wortes in dieser Bedeutung, darauf hinweist, dass der Satz, in dem es steht, subjektiv ist. (vgl. Wiebe und Mihalcea, 2006)

Wiebe und Mihalcea (2006) beschreiben einen möglichen Weg zur automatischen Klassifikation von Wortbedeutungen. Dazu wird ein Wert namens *Distributionally Similarity* benutzt, der anzeigt, wie oft zwei Wörter im selben Kontext benutzt werden. Ausgehend von einem

vorher annotierten Textkorpus, kann dann für oft im selben Kontext auftretende Wortbedeutungen die Klasse übernommen werden.

Um festzustellen, welche Bedeutung ein Wort in einem vorliegenden Satz hat, kann ein Verfahren zur *Word Sense Disambiguation (WSD)* eingesetzt werden. Für die Subjektivitätsanalyse ist es unwichtig, welche exakte Bedeutung ein Wort wirklich hat. Es reicht herauszufinden, ob diese Bedeutung subjektiv oder objektiv ist. Deswegen kann hier ein abgewandeltes Verfahren namens *Subjectivity Word Sense Disambiguation (SWSD)* eingesetzt werden. Durch die eingeschränkten Anforderungen kann das angepasste Verfahren bessere Ergebnisse erzielen. (vgl. Akkaya u. a., 2009)

Akkaya u. a. (2009) setzen als *SWSD*-Verfahren einen Naive-Bayes-Klassifikator ein. Dabei wird ein eigener Klassifikator für jedes Wort verwendet. Für die Featureauswahl wurde auf Erfahrungen mit der Entwicklung von *WSD*-Verfahren zurückgegriffen. Es werden unter anderem aus der Wortart des zu klassifizierenden Wortes, den umliegenden Wörtern, ihrer Part-of-Speech-Informationen und wichtigen Wörtern des Satzes, wie zum Beispiel dem Subjekt, Features gebildet. Der Klassifikator ordnet einem Wort nur die Bedeutung zu, in der es verwendet wird. Die Information, ob es sich um eine subjektive oder objektive Wortbedeutung handelt, wird aus einem vorher erstellten Lexikon abgelesen. *SentiWordNet* (Kapitel 3.1.1) ist ein Beispiel für ein solches Lexikon.

Obwohl die Verwendung einer als subjektiv eingeordneten Bedeutung eines Wortes in einem Satz auf Subjektivität hinweist, ist dies bei einer als objektiv eingeordneten Bedeutung und objektiven Sätzen nicht der Fall. (vgl. Wiebe und Mihalcea, 2006)

Ein Satz mit subjektiven Wortbedeutungen kann also direkt als subjektiv klassifiziert werden. Für objektive Wortbedeutungen kann so keine Klasse zugeordnet werden. Deswegen wird dieses Verfahren nicht alleine verwendet, sondern in Zusammenarbeit mit anderen Verfahren, wie zum Beispiel dem überwachten Lernen. Ordnet das andere Verfahren einem Satz die Klasse Objektiv zu und finden sich in dem Satz Wörter in subjektiver Bedeutung, wird die Klasse auf Subjektiv geändert, um den oben beschriebenen Fehlerfall bei subjektiven Wörtern in objektiven Verwendungen zu verhindern. Wird von dem anderen Verfahren die Klasse Subjektiv gewählt, obwohl alle Wörter des Satzes objektive Bedeutungen haben, wird betrachtet, wie sicher die Entscheidung des überwachten Lernverfahrens ist. Dazu wird ein Schwellenwert definiert, der anzeigt, wie sicher die Zuordnung sein muss, damit sie an dieser Stelle beibe-

halten wird. Der eingesetzte Wert und der Schwellenwert sind abhängig vom eingesetzten Lernverfahren. Dies wird gemacht, weil es zwar nicht sicher, aber sehr wahrscheinlich ist, dass ein Satz mit vielen objektiven Wortbedeutungen auch objektiv ist. Einzelne objektive Bedeutungen in zuerst als subjektiv eingeordneten Klassen verändern die Zuweisung nicht. Hier wird davon ausgegangen, dass die Subjektivität von anderen Merkmalen ausgeht. (vgl. Akkaya u. a., 2009)

Akkaya u. a. (2009) zeigen, dass diese zusätzlichen Informationen über die Wortbedeutungen Ergebnisse von den Klassifikationsverfahren deutlich verbessern können.

3.1.5 Zusammenfassung

Bei der Subjektivitätsanalyse geht es darum, Sätze mit Meinungen von denen zu trennen, die nur Fakten enthalten. Fakten können so bei der Klassifikation nach der Ausrichtung von Meinungen ignoriert werden.

Es gibt viele verschiedene Ansätze für die Subjektivitätsanalyse. Ein erster Ansatz ist die Klassifikation durch überwachtes Lernen. Die Verfahren, die dafür eingesetzt werden können, wie zum Beispiel der Naive-Bayes-Klassifikator, sind bereits für die Dokumentenklassifikation vorhanden. Für eine solche Klassifikation müssen aus den Sätzen Features extrahiert werden. Ausgehend von Features aus n-Grammen wurden weitere Features für die Klassifikation gefunden. Zum Beispiel konnten die Ergebnisse mit Features verbessert werden, die anzeigen, ob eine bestimmte Wortart in einem Satz enthalten ist.

Es wird versucht, bessere Indikatoren für Subjektivität und Objektivität zu finden, um diese beiden Klassen besser unterscheiden zu können. Ein vielversprechender Ansatz ist die Untersuchung von Adjektiven. Forschungsarbeiten zu diesem Thema haben gezeigt, dass Adjektive mit bestimmten Eigenschaften ein sehr guter Indikator für Subjektivität sind. Zum Beispiel deuten Adjektive mit positiver oder negativer semantischer Orientierung mehr auf Subjektivität hin als neutrale. Eine weitere Eigenschaft von Adjektiven ist die Steigerbarkeit. Steigerbare Adjektive werden häufiger für den Ausdruck von Meinungen verwendet als nicht steigerbare. Aus diesen Ergebnissen wurden eigene Klassifikatoren erstellt, die nach der Anzahl bestimmter Adjektive in einem Satz klassifizieren. Außerdem werden Adjektive und ihre Eigenschaften benutzt, um neue Features für das überwachte Lernen zu bilden. Zum Beispiel indem Features aus Informationen über die semantische Orientierung von Wörtern oder Wortgruppen gebildet werden.

Ein weiterer, neuerer Ansatz ist die Verbesserung der Klassifikation durch überwachtes Lernen mit Hilfe der Betrachtung von Wortbedeutungen. Anstatt jedes Wort daraufhin zu untersuchen, ob es auf Subjektivität oder Objektivität hindeutet, werden hier Wortbedeutungen benutzt. Dies macht die Unterscheidung genauer und kann helfen, bestimmte Fehlerarten bei der Klassifikation zu beheben. Hierzu wird allerdings ein Verfahren benötigt, welches in der Lage ist für ein Wort in einem Satz zu bestimmen, welche Bedeutung es hat. Informationen über Wortbedeutungen können einer Datenbank namens *WordNet* entnommen werden, die für viele Wörter die Bedeutungen mit einer Erklärung und Verweisen auf Synonyme enthält. *SentiWordNet* enthält zusätzlich zu diesen Informationen noch Werte, aus denen abgelesen werden kann, wie stark die Verwendung einer Wortbedeutung auf Subjektivität oder Objektivität hindeutet.

Eine Gemeinsamkeit aller vorgestellten Ansätze ist der Versuch, die Eigenschaften von subjektiven und objektiven Texten besser abzubilden und zu verstehen. Dazu wird untersucht, was die beiden Klassen ausmacht und was sie unterscheidet. Die Ergebnisse zeigen, dass sich ein Verfahren so meistens durch zusätzliche und genauere Informationen verbessern lässt.

3.2 Negationsverarbeitung

Verneinungen oder Negationen kommen häufig in Sätzen vor. Sie können starke Auswirkungen auf die Bedeutung des Satzes haben. Deswegen ist die Betrachtung von Negationen bei der *Sentiment Analysis* wichtig. Eine Negation kann die semantische Orientierung eines oder mehrerer Wörter ändern. Dabei ändert sich die Polarität von positv auf negativ oder umgekehrt.

Abbildung 3.4 zeigt ein paar Beispielsätze. Der erste Satz ist positiv. Dies ist für ein Verfahren zum Beispiel durch das Adjektiv „gut“ mit einer positiven Polarität zu erkennen. Im Satz ist die Polarität mit einem + gekennzeichnet. Eine Negation ist in der Lage, die Polarität dieses Adjektivs umzukehren. Damit kehrt sich auch die Aussage des Satzes um und der Satz wird negativ. Bei komplizierteren Satzkonstruktionen ist die Negation leider nicht so leicht zu behandeln. Nicht jedes Negationswort, wie zum Beispiel „nicht“ (im Englischen „not“), deutet darauf hin, dass die Meinung, die der Satz ausdrückt, umgekehrt wird. In Satz 3 zum Beispiel bleibt die Meinung des Satzes unabhängig vom „nicht“ negativ. Dies liegt daran, dass Negationsworte auch in Satzkonstruktionen, wie „nicht nur [...] sondern auch“, verwendet werden können, die keine Negationen sind. Negationen haben einen bestimmten Einfluss-

bereich, der für einen Menschen gleich verständlich, aber für eine Maschine schwerer zu erkennen ist. An Satz 4 lässt sich erkennen, dass ein Negationswort sich nicht auf den ganzen Satz beziehen muss. Nur der erste Teil des Satzes wird von dem „nicht“ negiert, der zweite Teil behält seine positive Bedeutung. Das bedeutet, dass es für die Negationsverarbeitung wichtig ist, den Bereich, dessen Meinung von der Negation beeinflusst wird, zu erkennen und von anderen Bereichen im Satz abzugrenzen. (vgl. Wiegand u. a., 2010)

1. „Ich finde diesen Film gut+.“
2. „Ich finde diesen Film [nicht gut+]-“
3. „Dieser Film ist nicht nur schlecht besetzt, sondern besitzt auch keine Handlung.“
4. „Ich mag diesen Film nicht, aber die Hauptrolle war sehr gut besetzt.“
5. „Ich fand den Film weniger gut.“
6. „Möglicherweise ist dies ein guter Film, aber ich kann nicht erkennen wieso.“
7. „Laut Filmbeschreibung sollten in diesem Film fähige Schauspieler mitspielen.“

Abbildung 3.4: Beispielsätze zur Erklärung der Negation (vgl. Wiegand u. a., 2010)

Die meisten Negationen in zu untersuchenden Texten sind viel komplizierter als hier in den ersten beiden Beispielsätzen gezeigt. Es gibt viele weitere Negationswörter neben den offensichtlichen, wie im Englischen „not“, „neither“ und „nor“. Die Sätze 5, 6 und 7 zeigen andere Möglichkeiten für Negationen. Mit abschwächenden oder verstärkenden Wörtern wie „weniger“ (Satz 5) und Konjunktionen (Satz 6) können Aussagen negiert werden. Außerdem ist es möglich, Negationen mit Modalsätzen zu bilden (Satz 7). (vgl. Wiegand u. a., 2010)

Die Verarbeitung von Negationen wird typischerweise in zwei Teilaufgaben aufgeteilt. Zuerst müssen Negationen identifiziert werden. Dazu werden Wörter gesucht, die eine Negation einleiten können. Im zweiten Schritt wird versucht, den Wirkungsbereich dieses Negationswortes zu bestimmen, um festlegen zu können, welche Wörter von der Negation beeinflusst werden.

Im Folgenden wird zuerst beschrieben, wie die Negationsverarbeitung als Vorverarbeitungsschritt bei überwachtem Lernen eingesetzt werden kann. Weiterhin wird dargestellt, wie die Verarbeitung durch Parsing verbessert werden kann. Anschließend wird auf die Bestandteile einer Negation eingegangen. Es wird veranschaulicht, wie Negationswörter, die eine Negation einleiten können, und der Bereich von Wörtern, auf den sie Auswirkung haben, identifiziert

werden können. Abschließend werden verschiedene Bewertungsmethoden für die Negationsverarbeitung vorgestellt und die einzelnen Abschnitte im Fazit kurz zusammengefasst.

3.2.1 Merkmale für Negationen

Negationswörter sind Wörter, die Negationen einleiten können. Sie können als Merkmal oder Hinweis auf eine Negation gesehen werden und sind deshalb in der Negationsverarbeitung sehr wichtig. Neben den offensichtlichen Negationswörtern wie „not“ gibt es viele weitere. Ein Beispiel hierfür ist „to lack“ (im Deutschen „fehlen“). Mit diesem Wort können auch Negationen gebildet werden, wie zum Beispiel „Diesem Film fehlt es an guten Schauspielern.“. Hier bekommt der Satz eine negative Bedeutung durch die Negation des Wortes „gut“.

Das Auffinden von Negationswörtern wird meistens über Listen realisiert. Diese Listen enthalten manuell ausgesuchte Wörter, die auf Negationen hindeuten. Dieses Vorgehen bietet sich an, weil eine automatische Erkennung von Negationswörtern aufgrund ihrer Unterschiedlichkeit schwierig ist und ihre Anzahl und damit die nötige Arbeit zur Erstellung einer solchen Liste gering ist.

Tabelle 3.1 zeigt ein paar Beispiele für Negationswörter, die in der Untersuchung von Councill u. a. (2010) benutzt wurden. Im Englischen wird „not“ oft mit einem Verb verbunden, indem „n’t“ angehängt wird. Um diese Negationen trotzdem erkennen zu können, werden diese Verben mit in die Listen mit Negationswörtern aufgenommen.

hardly	lack	lacking	lacks
neither	nor	never	no
nobody	none	nothing	nowhere
not	n’t	aint	cant
cannot	darent	dont	doesnt
didnt	hadnt	hasnt	havent
havent	isnt	mightnt	mustnt
neednt	oughtnt	shant	shouldnt
wasnt	wouldnt	without	

Tabelle 3.1: Beispiele für Wörter, die im Englischen auf explizite Negationen hindeuten. (Councill u. a., 2010)

Eine weitere Möglichkeit eine Aussage zu negieren, ist die Verwendung von Ironie. Es handelt sich dabei um eine implizite Negation, weil kein Negationswort für den Ausdruck der Ironie

gebraucht wird. Der ironische Satz kann aber trotzdem Negationswörter enthalten. Eine Untersuchung von Carvalho u. a. (2009) zeigt, dass ironische Aussagen in Texten eine häufige Fehlerquelle bei der Klassifikation von Meinungen sind. Ironie ist sehr schwer zu erkennen, weil dazu zusätzliches Wissen über die Welt nötig ist, das selten aus dem Text entnommen werden kann. Eine Möglichkeit, Ironie zu erkennen, vor allem in Texten von Benutzern, wie zum Beispiel Filmbewertungen, ist die Betrachtung einer außergewöhnlichen Verwendung von Satzzeichen. Zum Beispiel kann eine Anhäufung von Ausrufezeichen auf Ironie hindeuten. Smileys, die ebenfalls mit Satzzeichen dargestellt werden können, sind möglicherweise ein sehr guter Hinweis auf Ironie. (vgl. Carvalho u. a., 2009)

3.2.2 Wirkungsbereich einer Negation

Der Wirkungsbereich einer Negation wird *Scope of Negation (SoN)* genannt. Alle Wörter eines Satzes, die von einer Negation beeinflusst werden, gehören zu diesem Bereich. Verfahren für die Negationsverarbeitung unterscheiden sich vor allem in der Methode, den Wirkungsbereich festzustellen und alle Wörter in diesem entsprechend zu behandeln.

Ein sehr einfacher Ansatz, den Wirkungsbereich einer Negation festzulegen, ist eine feste Länge zu definieren. Alle Wörter, die hinter einem Negationswort stehen, werden zu dem Bereich hinzugezählt, wenn sie weniger als der definierte maximale Wortabstand entfernt sind. Dabei sollte nicht über das Satzende hinausgegangen werden, weil hier höchstwahrscheinlich der Wirkungsbereich nicht weitergehen wird. Auch bei dem Beginn von Nebensätzen kann dies gemacht werden, um zu verhindern, dass sie in die Bearbeitung einfließen. Dieses Verfahren ist sehr ungenau, weil nicht betrachtet wird, wie weit der Wirkungsbereich wirklich geht. Bei nicht zusammenhängenden Wirkungsbereichen wird ein Teil nicht erkannt, wenn sie zum Beispiel durch einen Nebensatz unterbrochen werden.

Bei anderen Sprachen als Englisch muss beachtet werden, dass hier der Wirkungsbereich oft nicht auf das Negationswort folgen muss und dieses Verfahren somit nicht so gut für die Sprachen geeignet ist. Im Deutschen zum Beispiel kann der Wirkungsbereich einer Negation sowohl vor als auch hinter dem Negationswort stehen. Ein Beispiel hierfür ist der Vergleich der Sätze „Peter mag den Kuchen nicht.“ und „Peter findet den Kuchen nicht köstlich.“. Hier müssen Informationen über die Grammatik des Satzes benutzt werden, um eine Entscheidung treffen zu können. (vgl. Wiegand u. a., 2010)

Die optimale maximale Länge eines Wirkungsbereiches kann durch Ausprobieren gefunden

werden. Dazu wird dieses Verfahren mit verschiedenen Werten vor der Dokumentenklassifikation ausgeführt und die Ergebnisse miteinander verglichen.

Pang u. a. (2002) zählten alle Wörter zwischen Negationswort und Satzende zum Wirkungsbereich dazu. Schon dieser sehr einfache Ansatz führte zu einem verbesserten Ergebnis.

Dadvar u. a. (2011) untersuchten die Auswirkung einer Negationsverarbeitung mit festen Fenstergrößen von 1 bis 5 und verschiedenen Negationswörtern aus einer vorgefertigten Liste. Die eingesetzten Dokumente zur Auswertung entsprechen den hier eingesetzten Trainingsdaten von Filmbewertungen. Zur Klassifikation der Dokumente in die Klassen Positiv und Negativ wurden zwei Wortlisten mit positiven und negativen Wörtern, meist Adjektive, erstellt. Ein Dokument wurde klassifiziert, indem gezählt wurde, wie oft Wörter aus den Listen in diesem vorkommen. Ohne Negationsverarbeitung erreichte dieser Klassifikator eine Genauigkeit von 65%. Mit der Verarbeitung von Negationen, die durch „not“ eingeleitet werden, konnten die Ergebnisse nicht merklich verbessert werden. Erst durch weitere Negationswörter, wie „no“, „rather“, „hardly“ und durch „n't“ verneinte Verben, konnte eine Verbesserung von bis zu 5% auf eine Genauigkeit von 70% erzielt werden. Die Untersuchung von verschiedenen Größen des Wirkungsbereiches ergaben dabei bei Werten von 1 bis 5 nur geringe Unterschiede. Eine Untersuchung dieses Ansatzes zur Verbesserung einer Klassifikation durch überwachtes Lernen wurde von Dadvar u. a. (2011) noch nicht durchgeführt.

3.2.3 Negationsverarbeitung mit Hilfe von Parsing

Parsing ist eine Methode, die Struktur einer Eingabe zu erkennen und nach einer gegebenen Grammatik zu zerlegen. Beim Parsing von natürlicher Sprache wird versucht, die grammatikalische Struktur eines Satzes zu erkennen. Die Grammatik von natürlichen Sprachen ist sehr komplex. Deswegen lässt sich diese Grammatik nicht einfach definieren. Es gibt Mehrdeutigkeiten, die der Mensch aus dem Kontext erkennen kann, und Ausnahmen, die von festen Regeln falsch behandelt werden würden. Deswegen kann ein Parser die Grammatik eines Satzes nicht immer korrekt erkennen und verschiedene Parser können unterschiedliche Ergebnisse liefern.

Es gibt mehrere frei verfügbare Parser für englische Sätze. Ein häufig eingesetzter ist der Stanford-Parser[1] von der *Stanford Natural Language Processing Group*. Es handelt sich dabei um eine Java-Implementierung eines statistischen Parsers. Das heißt, es wurden Wahrscheinlichkeiten für bestimmte grammatikalische Strukturen aus per Hand annotierten Sätzen berechnet.

[1] Der Stanford-Parser ist verfügbar unter: http://nlp.stanford.edu/software/lex-parser.shtml

Diese Wahrscheinlichkeiten können dann von dem Parser benutzt werden, um die wahrscheinlichste Struktur eines Satzes zu berechnen. Ein weiterer Parser, der für diese Aufgabe benutzt werden kann, ist der *OpenNLP*-Parser[2] aus dem *OpenNLP*-Projekt von Apache. Es handelt sich hierbei ebenfalls um einen statistischen Parser, der in Java implementiert wurde.

Bei der Negationsverarbeitung kann es sehr nützlich sein, die grammatikalische Struktur eines Satzes zu kennen, um den Wirkungsbereich einer Negation festlegen zu können.

Abbildung 3.5 zeigt einen Parsebaum eines Beispielsatzes aus dem *BioScope*-Textkorpus. Der Satz S wird in seine grammatikalischen Bestandteile zerlegt. Der Beispielsatz beginnt mit einer Nominalphrase NP, gefolgt von einer Verbalphrase VP. Phrasen bezeichnen eine Gruppe von grammatikalisch zusammenhängenden Wörtern. Nominalphrasen haben als Hauptbestandteil ein Nomen und Verbalphrasen ein Verb. Diese Phrasen werden dann bis auf die Ebene der einzelnen Wörter zerlegt. An dem Aufbau eines Parsebaumes lassen sich die Struktur und die zusammenhängenden Einheiten des Satzes erkennen. Die Negation, eingeleitet durch „lack“, und ihr Wirkungsbereich sind in dem Satz markiert. Es ist gut zu erkennen, dass sich die Begrenzungen des Wirkungsbereiches der Negation im Parsebaum wiederfinden. Die Negation verneint in diesem Fall die direkt folgende Nominalphrase „aktive NF-kappa B“. Das darauf folgende „but“ leitet einen anderen Satzteil ein und gehört nicht mehr zum Bereich dazu.
Ausgehend vom Parsebaum eines Satzes, lässt sich eine Negation und ihr Wirkungsbereich durch syntaktische und lexikalische Regeln finden. Diese Regeln müssen per Hand von Experten erstellt werden und können dann für eine automatische Verarbeitung genutzt werden. Das manuelle Erstellen dieser Regeln ist sehr aufwendig und der Erfolg der Verarbeitung stark abhängig davon, wie gut die Ergebnisse des Parsers sind. Bei Fehlern im Parsebaum können die Regeln nicht korrekt angewendet werden und es kommt zu falschen Ergebnissen, da die Regeln auf Basis der richtigen Grammatik erstellt wurden und die möglichen Fehler des Parsers dabei nicht berücksichtigt wurden.

In einer Untersuchung von Apostolova u. a. (2011) wird versucht, automatisch syntaktische Regeln zur Identifikation von Negationen und ihrer Wirkungsbereiche aus Texten zu extrahieren. Diese Regeln wurden dann nicht mehr auf Basis der eigentlichen Grammatik erstellt, sondern auf der möglicherweise fehlerhaften, aber einheitlichen Ausgabe des Parsers. Ein weiterer Vorteil ist, dass die komplizierte manuelle Erstellung der Regeln nicht mehr notwendig ist. Dazu wurden zuerst alle Sätze mit Negationswörtern mit dem Stanford-Parser untersucht.

[2] Der *OpenNLP*-Parser ist verfügbar unter: http://opennlp.apache.org/

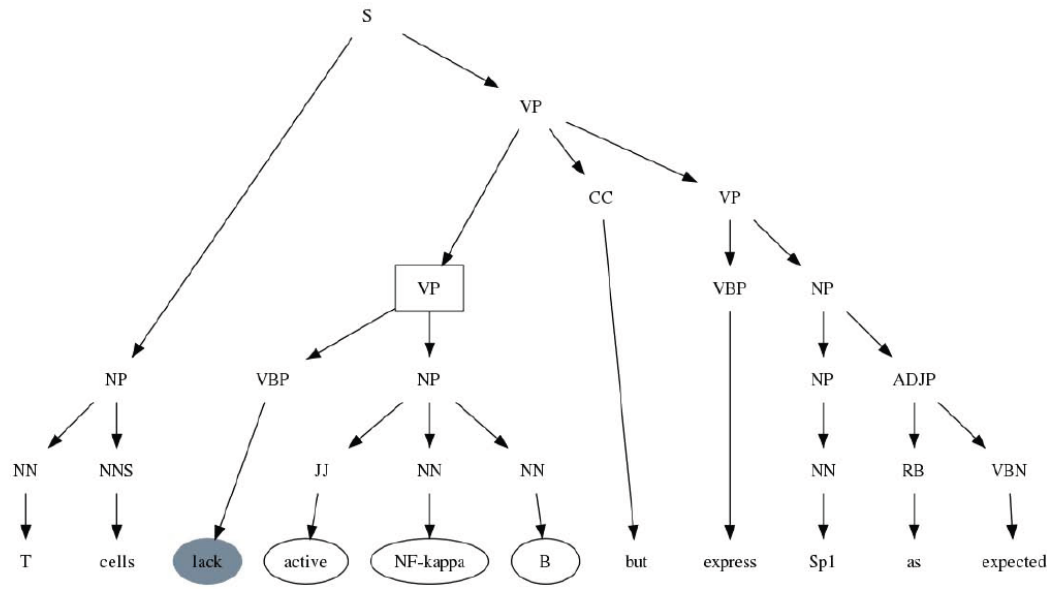


Abbildung 3.5: Parsebaum eines Beispielsatzes aus dem *BioScope*-Textkorpus (Apostolova u. a., 2011)

In den so generierten Parsebäumen wurde der tiefste gemeinsame Knoten der Negation und der Wörter im Wirkungsbereich identifiziert. Dieser Knoten definiert dann eine Regel für ein Negationswort und eine bestimmte grammatikalische Struktur, die den Wirkungsbereich dieser Negation zeigt.

Abbildung 3.6 zeigt eine Beispielregel, die aus dem Parsebaum aus Abbildung 3.5 erstellt werden kann. Die Regel enthält die grammatikalische Struktur des relevanten Bereiches und kann dazu verwendet werden, neue Sätze mit gleicher Struktur nach dem Negationswort „lack“ zu verarbeiten.
Auf diese Weise wurden aus dem *BioScope*-Korpus über 1600 Regeln extrahiert. Diese Regeln sind sehr spezifisch, weil sie sehr genau auf die grammatikalische Struktur des Wirkungsbereiches eingehen. Zum Beispiel lässt sich die Regel aus Abbildung 3.6 nur auf Sätze anwenden, die das Wort „lack“, gefolgt von einer Nominalphrase, bestehend aus genau drei Wörtern, enthält. Deswegen werden allgemeinere Regeln aus der großen Anzahl spezifischer erstellt, um den Recall-Wert zu erhöhen. Dazu wird eine Transformation auf die Regeln angewendet. Hat ein Knoten der Regel nur **scope**-Kindknoten, wird er selbst zu einem **scope**-Knoten und die Kindknoten werden entfernt. Benachbarte Knoten vom Typ **scope** werden zu einem Knoten zusammengefügt. Dieser Schritt kann gemacht werden, weil bei der späteren Verarbeitung

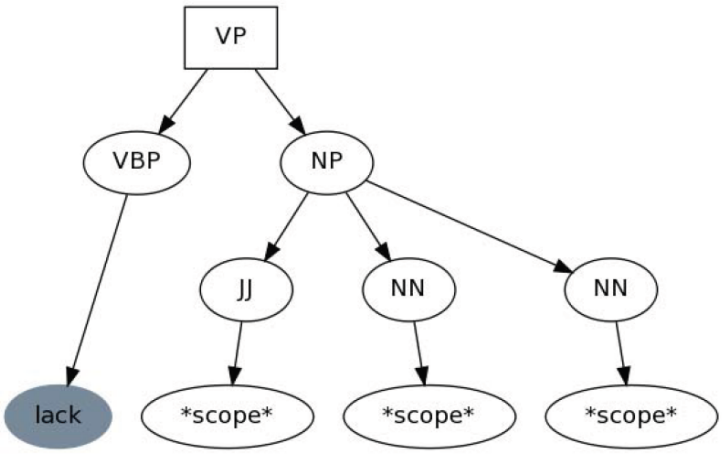


Abbildung 3.6: Regel, erstellt aus dem Parsebaum in Abbildung 3.5 (Apostolova u. a., 2011)

ein **scope**-Knoten in einer Regel für eine beliebige Anzahl von Wörtern steht. Diese Weiterverarbeitung der Regeln reduzierte die Anzahl auf 439 allgemeinere Regeln, die jetzt dazu benutzt werden können, neue unbekannte Sätze mit Hilfe ihres Parsebaums zu verarbeiten. Der Parsebaum des neuen Satzes wird mit den vorhandenen Regeln verglichen. Passt eine Regel, werden das darin enthaltene Negationswort und der Wirkungsbereich im Satz markiert. (vgl. Apostolova u. a., 2011)

Abbildung 3.7 zeigt die verallgemeinerte Regel des benutzten Beispielsatzes. Die Regel sagt aus, dass alle Wörter in einer Verbalphrase nach dem Wort „lack“ zur Negation gehören. Der Wirkungsbereich endet mit der Verbalphrase. Die restliche Struktur des Satzes ist für die Regel nicht relevant.
Apostolova u. a. (2011) erzielten mit den extrahierten Regeln sehr gute Ergebnisse bei der Negationsverarbeitung. Bei einem Test mit den gesamten unangepassten Regeln erreichten sie einen sehr hohen Precision-Wert in Höhe von 95%. Allerdings bei einem eher schlechten Recall-Wert, sodass der F-Score bei etwa 40% lag. Dies liegt daran, dass die extrahierten Regeln sehr speziell sind und sehr selten auf eine Negation im Textkorpus, der für die Auswertung benutzt wurde, passen. Gibt es allerdings eine passende Regel, führt dies mit einer hohen Wahrscheinlichkeit zu einer korrekten Verarbeitung. Mit den verallgemeinerten Regeln wurde ein F-Score-Wert von etwa 88% bei den untersuchten klinischen Dokumenten erreicht. Die Werte basieren auf der Anzahl der korrekt zugeordneten Wörter zu dem Wirkungsbereich

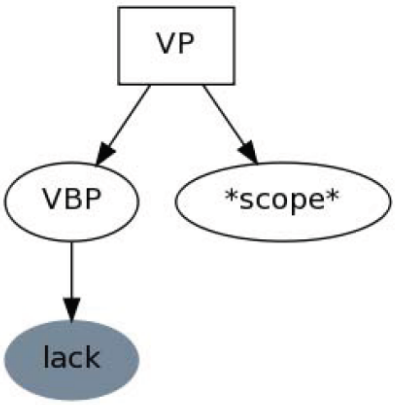


Abbildung 3.7: Weiterverarbeitete Regel, erstellt durch eine Transformation aus der Regel in Abbildung 3.6 (Apostolova u. a., 2011)

einer Negation.

Ein anderer Ansatz, bei dem keine Trainingsdaten notwendig sind, wurde von Jia u. a. (2009) untersucht. Anstatt die Regeln aus Trainingsdaten automatisch zu extrahieren, wurden manuell verschiedene Regeln definiert, mit denen eine Abgrenzung des Wirkungsbereiches möglich ist. Im ersten Schritt wird der Wirkungsbereich durch eine sehr allgemeine Regel, die auf dem Parsebaum basiert, abgegrenzt. Es wird vom Negationswort aus ein Elternknoten gesucht, dessen Blattknoten sowohl das Negationswort als auch mindestens ein Wort rechts davon einschließen. So wird der tiefste gemeinsame Elternknoten des Negationswortes und des Wirkungsbereiches gefunden. In dem Parsebaum aus Abbildung 3.5 würde mit dieser Regel der Knoten VP, der mit einem Rechteck markiert ist, ausgewählt werden.

Ausgehend von dem so mit Hilfe des Parsebaum abgegrenzten vorläufigen Wirkungsbereich kann das Ergebnis weiter eingegrenzt werden. Der aus dem Parsebaum abgeleitete Wirkungsbereich entspricht oft noch nicht dem richtigen Wirkungsbereich. Ein Beispiel hierfür ist in Abbildung 3.8 zu sehen. Bei dem Satz „I don't like this movie because the actors are bad“ würde die vorgestellte Regel dazu führen, dass alle Wörter nach dem Negationswort „n't“ zum Wirkungsbereich dazu gezählt werden würden. Dies ist aber nicht korrekt, denn es würde zum Beispiel die Aussage über die schlechten Schauspieler mit negiert werden. Es gibt Wörter, die den Wirkungsbereich einer Negation begrenzen können. Beispiele hierfür sind „when”,

„whenever", „whether", „because", „unless", „until", „since" und „hence". Mit dieser Regel würde der vorläufige Wirkungsbereich aus dem Beispiel auf „like this movie" begrenzt werden, weil hier „because" als Begrenzer wirkt. (vgl. Jia u. a., 2009)

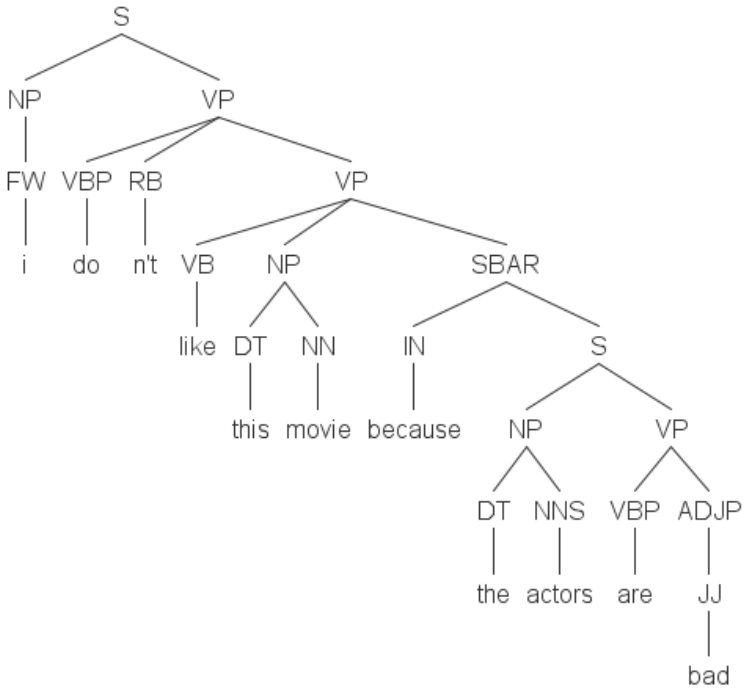


Abbildung 3.8: Parsebaum für den Satz „I don't like this movie because the actors are bad" (erstellt mit dem Stanford-Parser)

Auch andere Wörter können den Wirkungsbereich einer Negation begrenzen. Folgt auf das Negationswort ein Nomen, Verb oder Adjektiv, das eine Meinung ausdrückt, kann dies ebenfalls den Wirkungsbereich beenden. In dem Beispielsatz aus Abbildung 3.8 würde dies dazu führen, dass der Wirkungsbereich des „n't" nur noch „like" umfasst, weil „like" ein Verb ist, welches eine Meinung ausdrückt.

Außerdem definieren Jia u. a. (2009) Ausnahmen für Negationen. Diese Ausnahmen können dazu führen, dass der gesamte vorläufige Wirkungsbereich der Negation verworfen wird. Zum Beispiel gibt es keinen Wirkungsbereich, wenn das Negationswort Teil eines Ausdrucks ist, der keine Negation bewirkt. Solche Ausdrücke können "not only", "not just", "not to mention" und "no wonder" sein. Eine weitere Ausnahme stellen rhetorische Fragen dar. Diese können nur schwer automatisch erkannt werden. Eine Näherung besteht darin, alle Sätze, die

eine Frage sind, und ein Negationswort in den ersten drei Wörtern enthalten, als rhetorische Fragen zu betrachten. Für diesen Fall würde dann der vorläufige Wirkungsbereich vollständig gelöscht werden. Weitere Regeln für Begrenzer und Ausnahmen können in der Untersuchung von Jia u. a. (2009) nachgelesen werden.

Die Auswertung dieser Vorgehensweise mit vorläufigem Wirkungsbereich und eingrenzenden Regeln wurde auf Basis von mehr als 1000 Sätzen gemacht, die aus verschiedenen Reviews stammen. Jeder Satz wurde zuerst auf Negationen untersucht und dann als positiv oder negativ klassifiziert. Dabei ergab sich bei der Verwendung des vorläufigen Wirkungsbereiches eine Genauigkeit von ungefähr 80%. Durch die definierten Regeln zur Eingrenzung des Wirkungsbereiches wurde die Genauigkeit um etwa 6% verbessert. Dies zeigt, dass diese Regeln sehr gut dazu geeignet sind, den Wirkungsbereich einer Negation zu finden. Damit ist diese Methode eine gute Alternative zu den automatisch extrahierten Regeln, wenn keine Trainingsdaten vorhanden sein sollten.

3.2.4 Negationsverarbeitung bei überwachtem Lernen

Die Ergebnisse der Negationsverarbeitung müssen an das Klassifikationsverfahren übergeben werden. Bei einer Klassifikation durch überwachtes Lernen wird ein Satz nicht direkt dem Klassifikationsverfahren übergeben, sondern nur die Features, die daraus extrahiert wurden. Deswegen müssen die Ergebnisse der Negationsverarbeitung in die Features mit einfließen, um eine Verbesserung der Klassifikation durch die Negationsverarbeitung zu ermöglichen. Da meistens aus den Wörtern des Satzes Features gebildet werden, werden hier künstliche Features benutzt, die anzeigen, dass ein Wort im Wirkungsbereich einer Negation stand. Dies kann zum Beispiel dadurch erreicht werden, dass die Negationsverarbeitung negierte Wörter kennzeichnet, indem bei der Feature-Extraktion jedes Feature aus einem negierten Wort ein Präfix wie „NOT_“ bekommt. (vgl. Wiegand u. a., 2010)

Durch die Modifikation von Features aus negierten Wörtern werden normale und negierte Wörter vom Klassifikationsverfahren getrennt voneinander ausgewertet. Dadurch geht die Information verloren, dass das negierte Wort die gegenteilige Polarität hat, weil das normale und das negierte Wort für den Algorithmus unabhängige Features bekommen. Außerdem werden bei dieser Art der Verarbeitung Features aller Wörter in dem Wirkungsbereich verändert, unabhängig davon, ob das Wort überhaupt negierbar ist. Dies ist sprachlich nicht korrekt und führt zu dem Problem, dass die modifizierten Features sehr selten in dem Textkorpus auftreten. Auch ohne Negationsverarbeitung können Negationen schon durch Features

abgebildet worden sein. Zum Beispiel können durch n-Gramme Features gebildet werden, die eine Negation teilweise oder sogar vollständig abdecken. Durch diese Einschränkungen für die Repräsentation der Negationsverarbeitung in den Features ist die Verbesserung der Klassifikationsergebnisse begrenzt. Die Verbesserung durch die Negationsverarbeitung für andere Verfahren, zum Beispiel einer satzbasierten Analyse mit grammatikalischen Regeln, sollte im Vergleich zur Verbesserung der Klassifikation durch überwachtes Lernen also größer ausfallen. (vgl. Wiegand u. a., 2010)

3.2.5 Bewertung von Verfahren zur Negationsverarbeitung

Die Bewertung der verschiedenen Verfahren für die Negationsverarbeitung ist schwieriger als bei einfachen Klassifikationsverfahren. Das Ergebnis der Negationsverarbeitung ist nicht nur eine Klasse, die mit einem Soll-Ergebnis verglichen werden kann, sondern mindestens ein Intervall, welches angibt, wo der Wirkungsbereich der Negation anfängt und wo er endet. Zusätzlich kann noch das Negationswort gekennzeichnet sein, um es besonders zu verarbeiten. Ein Satz kann mehrere, sich überlappende Negationsbereiche haben. Deswegen ist es schwierig, automatisch zu bewerten, wie gut ein Verfahren arbeitet. Verschiedene Fehler können hier unterschiedlich bewertet werden. Zum Beispiel sollte für eine gute Auswertung ein richtig erkanntes Negationswort mit einem leicht abweichenden Wirkungsbereich nicht vollständig als Fehler gewertet werden.

Eine Möglichkeit, diese Probleme zu umgehen, ist nicht, die Verfahren selbst zu messen, sondern zu untersuchen, wie stark die einzelnen Verfahren zur Vorverarbeitung die Ergebnisse der Klassifikation von Dokumenten verbessern. Dazu wird eine Auswertung ohne eine Negationsverarbeitung ausgeführt und dann werden nacheinander die Verfahren zur Negationsverarbeitung eingesetzt. Die Differenzen der Ergebnisse der Auswertung können als Bewertung für die Verfahren genommen werden. Diese Art der Bewertung ist sehr ungenau, kann aber für alle Vorverarbeitungskonzepte eingesetzt werden. Es wird nur untersucht, ob die Klassifikation verbessert wird. Wie gut das Verfahren Negationen wirklich erkennt und ob die Wirkungsbereiche korrekt bestimmt werden, wird nicht betrachtet. Es muss davon ausgegangen werden, dass eine richtige Verarbeitung der Negationen die Ergebnisse der Klassifikation am stärksten verbessert. Dadvar u. a. (2011) wählten bei der Untersuchung der Auswirkung verschiedener fester Fenstergrößen diese Bewertungsmethode.

Eine genauere Untersuchung ist mit Verifikationsdaten möglich. Diese Daten bestehen aus Sätzen, deren Negationen und Wirkungsbereiche per Hand markiert wurden. Es muss defi-

niert werden, wie die Bewertung durch den Vergleich von Ergebnissen der Verarbeitung mit den Verifikationsdaten berechnet wird. Der einzige mir bekannte Textkorpus, der für diese Aufgabe eingesetzt werden kann, ist der *BioScope*-Korpus. Er besteht aus biomedizinischen Veröffentlichungen und umfasst insgesamt über 20.000 Sätze, in denen Negationen und ihre Wirkungsbereiche markiert wurden (vgl. Szarvas u. a., 2008). Eine mögliche Bewertung mit diesem Korpus ist der F-Score-Wert für die richtig als Negation erkannten Bereiche. Hier wird nicht betrachtet, ob der Negationsbereich wirklich vollständig korrekt erkannt wurde. Dies kann mit einem weiteren Wert abgebildet werden, der anzeigt, wie viele Wirkungsbereiche von Negationen vollständig korrekt erkannt wurden. Der Wert für vollständig übereinstimmende Negationsbereiche ist deutlich geringer. Da selbst Menschen den Wirkungsbereich einer Negation nicht immer eindeutig bestimmten können, sind damit auch die Verifikationsdaten nicht perfekt. Für die Vorverarbeitung ist es außerdem nicht ausschlaggebend, ob der Wirkungsbereich immer vollständig erkannt wird. Dies sollte bei der Interpretation dieser Werte beachtet werden.

3.2.6 Zusammenfassung

Die Negationsverarbeitung besteht aus zwei Schritten. Zuerst müssen die Negationen in dem zu untersuchenden Text gefunden werden. Dazu wird nach sogenannten Negationswörtern gesucht. Ein Negationswort ist ein Wort, welches eine Negation einleiten kann. Ein einfaches Beispiel hierfür ist das Wort „not“.

Im zweiten Schritt wird ausgehend von jedem Negationswort versucht, die Wörter zu identifizieren, die von dem Negationswort beeinflusst werden. Diese Wörter stehen im Englischen meist hinter dem Negationswort. Deshalb besteht ein einfacher Ansatz zur Abgrenzung des Wirkungsbereiches darin, eine feste Fenstergröße anzunehmen. Zum Beispiel werden bei einer festen Fenstergröße von 5 die fünf auf das Negationswort folgenden Wörter zum Wirkungsbereich dazu gezählt. Da es unwahrscheinlich ist, dass der Wirkungsbereich über einen Satz hinausgeht, wird spätestens am Satzende abgebrochen. Bei anderen Sprachen, wie zum Beispiel im Deutschen, kann der Wirkungsbereich eines Negationswort auch davor stehen, deswegen ist dieser Ansatz hier nicht zu empfehlen.

Obwohl dieser Ansatz sehr ungenau ist, weil er unabhängig von der Satzstruktur und den Wörtern im Satz arbeitet, zeigen Untersuchungen, wie zum Beispiel von Dadvar u. a. (2011), dass er durchaus in der Lage ist, die Ergebnisse der *Sentiment Analysis* zu verbessern.

Eine genauere Abgrenzung des Wirkungsbereiches ist mit Parsing-Informationen möglich. Ein Parser für natürliche Sprache kann die grammatikalische Struktur eines Satzes analysieren. Diese Informationen können bei der Negationsverarbeitung benutzt werden. Eine Möglichkeit, dies zu tun, besteht darin, einen Textkorpus per Hand mit Annotationen für Negationswörter und ihre Wirkungsbereiche zu versehen. Aus dem Trainingskorpus können dann aus diesen Annotationen und einem Parsebaum des Satzes, Regeln erstellt werden, wo ausgehend von dem Negationswort, in der grammatikalischen Struktur des Satzes der Wirkungsbereich zu finden ist. Mit diesen Regeln ist es möglich, die Abgrenzung für dieses Negationswort in anderen Sätzen sehr genau vorzunehmen. Dieser Ansatz wird in einer Untersuchung von Apostolova u. a. (2011) beschrieben und sollte sich auch für andere Sprachen einsetzen lassen, wenn hierfür ein Parser gefunden wird.

Um die Ergebnisse der Negationsverarbeitung bei der Klassifikation der Dokumente einfließen zu lassen, werden alle Features, die aus Wörtern gebildet werden, welche im Wirkungsbereich der Negation liegen, verändert. Dazu kann zum Beispiel ein „NOT_“ vor das Wort gesetzt werden. Ein Problem dabei ist, dass dabei Informationen verloren gehen. Der Klassifikationsalgorithmus behandelt die Features „NOT_Wort“ und „Wort“ aufgrund der Veränderung wie zwei vollkommen unterschiedliche und nicht wie zwei Wörter mit entgegengesetzter semantischer Orientierung.

Für die Bewertung, wie genau die Ergebnisse eines Verfahrens zur Negationsverarbeitung sind, können die Ergebnisse mit per Hand erstellten Annotationen verglichen werden. Sind keine per Hand erstellten Vergleichsdaten vorhanden, kann die Auswertung auf Grundlage der Verbesserung der *Sentiment Analysis* gemacht werden. Dazu werden die Ergebnisse mit und ohne Negationsverarbeitung miteinander verglichen.

4 Testumgebung

4.1 Realisierung

Die vorgestellten Konzepte zur Vorverarbeitung sollen in der Testumgebung einzeln getestet werden, um später die Ergebnisse zu vergleichen. Im Folgenden wird beschrieben, welche Verfahren für diese Auswertung ausgewählt wurden und wie diese umgesetzt werden.

4.1.1 Subjektivitätsanalyse

Adjektive

Für die Klassifikation auf Basis verschiedener Arten von Adjektiven werde ich einen Klassifikator schreiben, der basierend auf einer gegebenen Menge von Adjektiven zählt, wie viele Wörter aus der Menge in dem Satz vorkommen. Enthält ein Satz mehr als eine definierte Anzahl dieser Wörter, wird er als subjektiv und andernfalls als objektiv eingeordnet.

Über die Wahl der Adjektive in der Menge wird gesteuert, welche Eigenschaften, wie zum Beispiel semantische Orientierung und Steigerbarkeit, der Adjektive untersucht werden. Durch eine Veränderung des Schwellenwertes, der definiert, wie viele Adjektive aus der Menge vorkommen müssen, kann gesteuert werden, wie sicher die Klassifikation sein muss. Eine Erhöhung des Schwellenwertes sollte den Precision-Wert verbessern und dafür den Recall-Wert verschlechtern, weil die Entscheidung bei mehreren vorkommenden Wörtern zwar sicherer ist, es aber immer seltener vorkommt, dass ein subjektiver Satz so viele Adjektive aus der definierten Menge enthält.

Für die Untersuchung der Auswirkung der semantischen Orientierung werde ich die *SentiWordNet*-Datenbank benutzen. Mit den Informationen aus der Datenbank werden alle Adjektive als positiv oder negativ einsortiert. Danach wird untersucht, wie gut der Klassifikator mit den positiven und negativen Mengen arbeitet.

Für die Untersuchung von dynamischen und steigerbaren Adjektiven habe ich leider kei-

ne Daten gefunden. Da die Unterscheidung von dynamischen und statisches Adjektiven sehr schwer sein kann und mir kein automatisches Verfahren für diese Unterscheidung bekannt ist, werde ich sie bei der Untersuchung weglassen. Semantische Orientierung und Steigerbarkeit liefern laut Literatur bessere Ergebnisse. Bei meiner Untersuchung werde ich deshalb die am häufigsten vorkommenden Adjektive in den Trainingsdaten bestimmen und diese per Hand in steigerbar und nicht steigerbar einordnen. Die Implementierung eines automatischen Verfahrens wäre sehr aufwendig und würde einen größeren Textkorpus erfordern. Außerdem müssten auch hier eine große Menge Adjektive per Hand eingeordnet werden.

Klassifikation durch überwachtes Lernen

Bei der Klassifikation von Subjektivität durch überwachtes Lernen kann derselbe Klassifikator wie bei der Dokumentenklassifikation eingesetzt werden. Das Verfahren trainiert und klassifiziert ausschließlich auf Basis der Features, die aus dem Text extrahiert wurden.

Da die Trainingsdaten und Daten für die Verifikation bereits vorliegen, muss nur noch das Verfahren für die Feature-Extraktion realisiert werden. Hier werde ich neben den vorgestellten Featuremengen aus anderen Untersuchungen, wie Features für die Präsenz von bestimmten Wortarten und die semantische Orientierung von Wortgruppen, auch einfachere Kombinationen testen, um vergleichen zu können. Dazu wird zuerst mit Unigrammen, Bigrammen, Trigrammen und Kombinationen davon getestet.

Um die Untersuchungen von Wiebe u. a. (1999) nachzuvollziehen, werde ich einen Testdurchlauf mit verschiedenen Kombinationen aus Features machen, die anzeigen, ob bestimmte Wortarten im Satz vorhanden sind oder nicht. Da die eingesetzten Trainingsdaten ausschließlich aus Filmbewertungen bestehen und diese meistens keine Einteilung in Textabschnitte besitzen, werde ich das Feature dafür weglassen.

Weiterhin werde ich die Featureauswahl von Yu und Hatzivassiloglou (2003) testen. Hier kommen zu n-Grammen Features hinzu, die die semantische Orientierung von bestimmten Satzbestandteilen anzeigen. Diese werde ich mit Hilfe der Informationen aus der *SentiWordNet*-Datenbank bilden.

Wortbedeutung

Die Verbesserung der Klassifikation durch die Unterscheidung verschiedener Wortbedeutungen erscheint mir sehr vielversprechend. Leider lässt sich dieser Ansatz nicht leicht umsetzen.

Für das Verfahren müssen Wortbedeutungen klassifiziert oder per Hand eingeordnet werden. Für die Klassifikation muss auch eine erhebliche Menge an Trainingsdaten per Hand erstellt werden. Die verfügbaren Daten für die vorgestellten Untersuchungen sind leider nicht für den Einsatz bei Filmreviews geeignet. Es wurden nur sehr wenige Wortbedeutungen klassifiziert. Fehlen die Wortbedeutungen für viele Wörter eines Satzes, kann keine Klassifikation auf Basis der Bedeutungen gemacht werden und die Ergebnisse können damit auch nicht verbessert werden.

Nachdem diese Daten erstellt worden sind, muss ein Programm geschrieben werden, welches für ein Wort in einem Satz bestimmt, in welcher Bedeutung es steht, beziehungsweise, ob dieses Wort eine subjektive oder objektive Bedeutung hat. Leider habe ich kein frei verfügbares Programm gefunden, welches diese Aufgabe übernehmen kann. Da eine eigene Implementierung sehr aufwendig wäre und *Word Sense Disambiguation* ein eigenes Forschungsgebiet ist, habe ich mich dazu entschieden, dieses Verfahren nicht selbst zu testen.

Kategorie	**Art des Tests**	**Eingesetzte Mittel**
Adjektive	alle Adjektive	Part-of-Speech-Tagger
	alle Adjektive in *SentiWordNet*	*SentiWordNet*
	positive Adjektive	
	negative Adjektive	
	positive und negative Adjektive	
	steigerbare Adjektive	manuell
	nicht steigerbare Adjektive	
	positive und steigerbare Adjektive	Kombination
	negative und steigerbare Adjektive	
überwachtes Lernen	Unigramme	n-Gramm Feature-Extraktor
	Bigramme	
	Trigramme	
	Unigramme, Bigramme und Trigramme	
	Präsenz von Wortarten	Parser
	n-Gramme und sem. Orientierung	Kombination
	n-Gramme, sem. Orientierung und Wortarten	

Tabelle 4.1: Übersicht über die geplanten Tests zu den einzelnen Verfahren für Subjektivitätsanalyse

Kategorie	Art des Tests	Begründung
Adjektive	dynamische Adjektive	Schwierig zu unterscheiden und kein automatisches Verfahren bekannt.
	statische Adjektive	
Wortbedeutung	Klassifikation von Wortbedeutungen	Umfangreiche und zur Domäne passende Trainingsdaten benötigt.
	Verbesserung von überwachtem Lernen durch Wortbedeutungen	Umsetzung von *SWSD*-Verfahren sehr komplex.

Tabelle 4.2: Übersicht über die im Rahmen dieser Arbeit nicht durchgeführten Tests zu den einzelnen Verfahren für Subjektivitätsanalyse

4.1.2 Negationsverarbeitung

Feste Fenstergröße

Ausgehend von den Untersuchungen von Dadvar u. a. (2011) werde ich die Auswirkungen der Negationsverarbeitung mit verschiedenen festen Fenstergrößen testen. Bei der vorgestellten Untersuchung wurde ein Klassifikationsverfahren, basierend auf festen Positiv- und Negativ-Listen, benutzt. Die Auswirkungen auf eine Klassifikation durch überwachtes Lernen möchte ich jetzt hier testen.

Als Negationswort werde ich zuerst nur „not“ verwenden. Die so erzielten Ergebnisse kann ich dann anschließend mit den Ergebnissen der Negationswörter aus Tabelle 3.1 vergleichen.

Getestet werden die Fenstergrößen 1 bis 5. Ist ein Satz vor der Verarbeitung der gesamten Fenstergröße zu Ende, wird dort abgebrochen. Das heißt, ein Negationsbereich kann sich nicht über mehrere Sätze erstrecken.

Parsing

Für das Parsing der Sätze werde ich den *OpenNLP*-Parser benutzen, da ich schon andere *OpenNLP*-Komponenten verwende. Für Filmbewertungen existiert leider kein annotierter Korpus für Negationen und ihre Wirkungsbereiche. Es ist also hier leider nicht möglich, Regeln aus Trainingsdaten automatisch zu extrahieren, deswegen werde ich den zweiten vorgestellten Ansatz verwenden. Dieser arbeitet mit einem vorläufigen Wirkungsbereich, der mit einer allgemeinen Regel aus dem Parsebaum erstellt wird. Der vorläufige Wirkungsbereich wird im zweiten Schritt durch weitere Regeln verkürzt.

Um den vorläufigen Wirkungsbereich festzustellen, verwende ich folgende allgemeine Regel: Ausgehend von der Negation geht der Algorithmus im Parsebaum solange nach oben, bis ein Knoten erreicht wird, der hinter dem Kindknoten, der das Negationswort beinhaltet, noch weitere Wörter enthält. Anschließend werden alle Blattknoten dieses so ausgewählten Elternknotens mit einer Markierung versehen, welche anzeigt, dass sie sich im Wirkungsbereich befinden, wenn sie im Satz nach der Negation auftreten. Im Englischen steht der Wirkungsbereich immer hinter der Negation. Damit Negationen ohne Zuordnung des Wirkungsbereiches verhindert werden, wird darauf geachtet, dass hinter dem Negationswort noch weitere Wörter folgen. Auf diese Weise erhält man einen von der grammatikalischen Struktur abhängigen Wirkungsbereich der Negation.

Danach werde ich die vorgestellten Regeln von Jia u. a. (2009) testen. Dazu wird der vorläufige Wirkungsbereich im ersten Test durch Begrenzer, wie zum Beispiel „since“, „when“ und „because“, verkleinert. Außerdem werde ich hier Kommata als Begrenzer mit aufnehmen, damit der Wirkungsbereich einer Negation keine Wörter aus einem Nebensatz beinhaltet. Weitere Begrenzer ergeben sich aus Wörtern, die eine Meinung ausdrücken. Um diese zu identifizieren, werde ich die Daten aus *SentiWordNet* verwenden. Im zweiten Test werde ich diese Begrenzer zusammen mit den Ausnahmeregeln testen. Diese Ausnahmen entfernen den ganzen Wirkungsbereich einer Negation, wenn das Negationswort in einem Ausdruck steht, der keine Negation auslöst. Ein Beispiel hierfür ist „not only“. Eine weitere Ausnahme stellen rhetorische Fragen dar. Um diese zu erkennen, wird eine Näherung verwendet. Jeder Satz, der mit einem Fragezeichen endet, wird als Frage betrachtet. Enthält eine Frage in den ersten drei Wörtern ein Negationswort, wird die Frage als rhetorisch angesehen.

Insgesamt werde ich also sechs Testdurchläufe machen. Einen Testdurchlauf werde ich nur mit dem vorläufigen Wirkungsbereich durchführen, um einen Vergleichswert zu erstellen. Danach werde ich die Auswirkungen der Begrenzer mit und ohne die Ausnahmeregeln testen. Diese drei Testdurchläufe werde ich jeweils mit dem Negationswort „not“ und dann mit allen Negationswörtern ausführen.

Berechnung der Ergebnisse

Die Auswertung der Ergebnisse wird auf Basis der Verbesserung der Klassifikation der Dokumente gemacht. Für eine genauere Auswertung der Negationsverarbeitung bräuchte ich einen annotierten Textkorpus, der Negationswörter und Wirkungsbereiche enthält. Es lässt sich also nicht messen, wie korrekt oder vollständig die Negationsverarbeitung funktioniert, sondern

nur, welche Auswirkungen dieser Vorverarbeitungsschritt auf die Dokumentenklassifikation hat. Nur Letzteres ist für diese Untersuchung wirklich notwendig.

Kategorie	Negationswörter	Art des Tests	Eingesetzte Mittel
feste Fenstergröße	„not“	Fenstergröße 1	
		Fenstergröße 2	
		Fenstergröße 3	
		Fenstergröße 4	
		Fenstergröße 5	
	Tabelle 3.1	Fenstergröße 1	
		Fenstergröße 2	
		Fenstergröße 3	
		Fenstergröße 4	
		Fenstergröße 5	
Parsing	„not“	vorläufiger Wirkungsbereich	*OpenNLP*-Parser, *SentiWordNet*
		Begrenzer	
		Begrenzer und Ausnahmen	
	Tabelle 3.1	vorläufiger Wirkungsbereich	
		Begrenzer	
		Begrenzer und Ausnahmen	

Tabelle 4.3: Übersicht über die geplanten Tests zu den einzelnen Verfahren für die Negationsverarbeitung

4.2 Eingesetzte Bibliotheken

Im Folgenden wird kurz beschrieben, welche externen Bibliotheken in der Testumgebung eingesetzt werden. Außerdem gehe ich darauf ein, warum ich mich für diese entschieden habe und gebe Hinweise darauf, wo sie zu finden sind.

4.2.1 *UIMA*

UIMA, *Unstructured Information Management Architecture*, ist eine Standardarchitektur für Anwendungen, die unstrukturierte Daten verarbeiten. Der Standard wird von der *OASIS* [1] weiterentwickelt und beschreibt, wie diese Anwendungen in Komponenten aufgeteilt werden können. Außerdem wird eine Datenstruktur namens *CAS*, *Common Analysis Structure*, definiert, die zur Weiterleitung der Informationen zwischen den einzelnen Komponenten benutzt

[1] *Organization for the Advancement of Structured Information Standards* – Die Spezifikation des *UIMA*-Standards ist zu finden unter: http://www.oasis-http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uima

wird. (vgl. The Apache UIMA Development Community, 2011)

Eine *Open Source*-Umsetzung dieses Standards in C++ und Java ist *Apache UIMA*. Das Projekt steht unter der Apache-Lizenz und bietet neben dem Framework auch einige Komponenten an.

Die zu verarbeitenden Daten können aus verschiedenen Quellen geladen werden. Dazu wird ein *Collection Reader* eingesetzt. Beispiele für Quellen sind Texte, aber auch Audio- und Video-Dateien. Die Analyse der Texte geschieht in *Analysis Engines*. Diese können in einer Pipeline hintereinandergeschaltet werden und zu *Aggregated Analysis Engines* kombiniert werden. Jede *Analysis Engine* hat die Möglichkeit, dem Text Annotationen hinzuzufügen oder auf Annotationen vorheriger Verarbeitungsschritte zuzugreifen. Annotationen sind Meta-Informationen über den Text und bieten die Möglichkeit, den Text mit Informationen anzureichern, ohne diesen zu verändern. Die so erstellten Annotationen können von *Consumer* ausgelesen, weiterverarbeitet und die Ergebnisse in beliebigen Formaten abgespeichert werden. Alle diese Komponenten können in einer *Collection Processing Engine* zusammengefügt werden. Die *CPE* steuert die Verarbeitung der Sammlung von zu verarbeitenden Daten und übernimmt die Fehlerbehandlung. Einzelne Komponenten können durch diese Architektur wiederverwendet und zu beliebigen Pipelines kombiniert werden. (vgl. The Apache UIMA Development Community, 2011)

Apache UIMA-Komponenten werden in XML-Dateien, sogenannten Deskriptoren, beschrieben. Diese Dateien definieren zum Beispiel die Datentypen der Annotationen und bieten die Möglichkeit, beliebige Parameter an Komponenten zu übergeben. (vgl. The Apache UIMA Development Community, 2011)

Abbildung 4.1 zeigt die Interaktion einer Anwendung mit dem *Apache UIMA*-Framework. Jede Komponente kann von der Anwendung erstellt oder über die Deskriptoren geladen und benutzt werden. Über *Listener* in einer *CPE* ist es möglich, den Fortschritt der Verarbeitung zu verfolgen. *Consumer* können dazu benutzt werden, Informationen nach Verarbeitung einer Datei oder der ganzen Datensammlung an die Anwendung weiterzugeben. (vgl. The Apache UIMA Development Community, 2011)
Ich habe mich dazu entschieden, eine Standardarchitektur zu verwenden, weil so einzelne Komponenten leicht mit anderen Arbeiten ausgetauscht und verglichen werden können. Außerdem konnte ich mich so auf den hier relevanten Kern der Testumgebung konzentrieren.

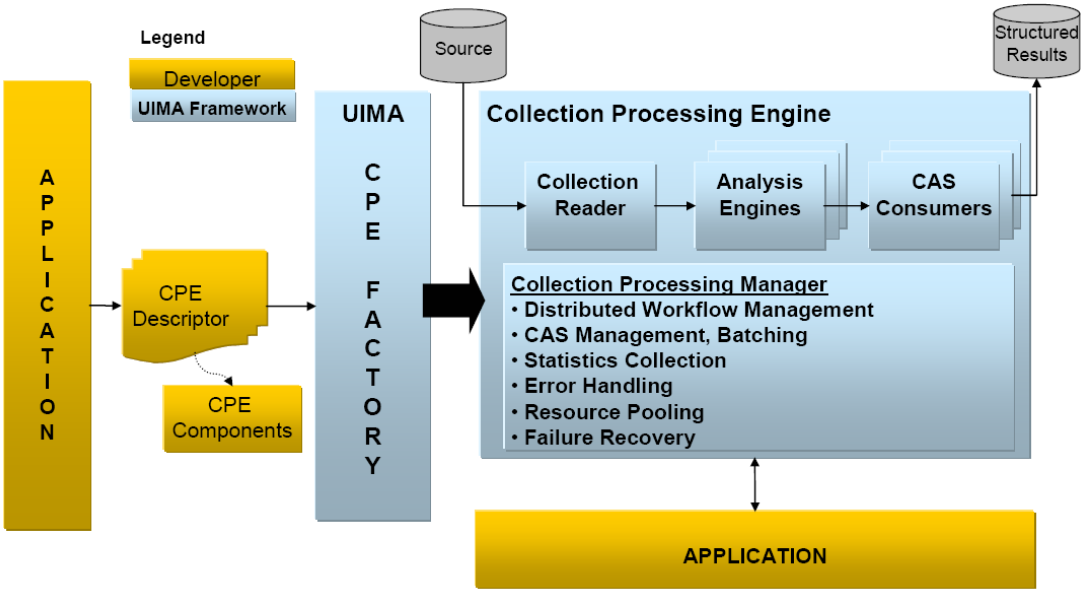


Abbildung 4.1: *UIMA* - Interaktion mit dem Framework (The Apache UIMA Development Community, 2011)

UIMA ist eine der häufigsten Architekturen, die für solche Anwendungsfälle verwendet werden.

4.2.2 *UIMAFit*

UIMAFit ist eine Art Erweiterung für *UIMA*. Es erleichtert die Definition von Komponenten durch zusätzliche abstrakte Klassen. Außerdem können die Parameter für eine Komponente über *UIMAFit* in Java definiert werden. Dies erleichtert die Programmierung, weil nicht für jede kleine Aufgabe komplexe XML-Dateien angelegt werden müssen. Die XML-Dateien können sogar automatisch generiert werden, wenn sie benötigt werden sollten. Weiterhin erleichtert die Bibliothek es, *UIMA*-Pipelines aus dem Programm aufzurufen, was besonders für das schnelle Testen verschiedener Verfahren hilfreich ist.

UIMAFit wird von einer Google-Code-Gruppe weiterentwickelt und steht wie *UIMA* selbst unter der Apache-Lizenz.[1]

[1] UIMAFit ist verfügbar unter: http://code.google.com/p/uimafit/

4.2.3 *OpenNLP*

OpenNLP ist eine Sammlung verschiedener Algorithmen zur Verarbeitung von natürlicher Sprache. *OpenNLP* wird von Apache unter der Apache-Lizenz zur Verfügung gestellt.[2] Unter anderem enthält es Satzerkennung, einen Tokenizer, um zum Beispiel Wörter in Sätzen zu markieren, einen Part-of-Speech-Tagger und einen Parser. Zusätzlich werden für alle Algorithmen *UIMA*-Komponenten bereitgestellt. Dies ist sehr nützlich, weil *UIMA* selber keine Komponenten für solche Aufgaben beinhaltet.

4.2.4 *ClearTK*

ClearTK ist ein Framework für *Natural Language Processing*-Komponenten, basierend auf *UIMA*. Es wird vom *Center for Computational Language and Education Research (CLEAR)* entwickelt und steht unter der *BSD*-Lizenz.[3] *ClearTK* erweitert *UIMA* um Feature-Extraktoren und Klassifizierer. Im Gegensatz zu *UIMA* enthält *ClearTK* viele fertige Komponenten, die verwendet werden können. Dazu zählen verschiedene Support-Vector-Maschinen, Naive-Bayes-Klassifikatoren und verschiedene vorgefertigte Feature-Extraktoren, wie zum Beispiel für n-Gramme. Alle diese Komponenten wurden auf der Basis anderer frei verfügbarer Projekte entwickelt. *ClearTK* enthält meist nur eine einheitliche Kapselung dieser externen Bibliotheken, die so in *UIMA* verwendet werden können. Alle *OpenNLP*-Komponenten lassen sich über *ClearTK* ohne weitere Anpassung verwenden.

Ich verwende *ClearTK*, weil es eine Austauschbarkeit der verschiedenen Verfahren gewährleistet und alle Verfahren, die ich benötige, wie Support-Vector-Maschinen und den Naive-Bayes-Klassifikator, unterstützt.

4.2.5 *Mallet*

Mallet, *Machine Learning for Language Toolkit*, ist eine Java-Bibliothek, die verschiedene Algorithmen zum maschinellen Lernen im Textverarbeitungsbereich bereitstellt. Einer dieser Algorithmen ist eine Implementierung eines Naive-Bayes-Klassifikators, die von *ClearTK* unterstützt wird.[4]

Mallet stellt unter anderem Klassen für die Feature-Extraktion und die Auswertung von

[2] OpenNLP ist verfügbar unter: http://opennlp.apache.org/
[3] *ClearTK* ist verfügbar unter: http://code.google.com/p/cleartk/
[4] *Mallet* ist verfügbar unter: http://mallet.cs.umass.edu/

Klassifikatoren zur Verfügung. Leider können diese nicht direkt mit *UIMA* verwendet werden, weil es keine *ClearTK*-Unterstützung für diese gibt. Deswegen und um Einheitlichkeit bei den verschiedenen Algorithmen zu gewährleisten, verwende ich die *ClearTK*-Klassen für diese Aufgaben, die mit allen unterstützten Bibliotheken zusammenarbeiten können.

4.2.6 *SVMlight*

SVMlight ist eine Implementierung von Support-Vektor-Maschinen in C.[5] Die Bibliothek wurde an der Universität Dortmund entwickelt, um Support-Vektor-Maschinen zugänglicher zu machen. Die Verwendung für die Forschung ist frei, nur für eine kommerzielle Nutzung muss eine Lizenz beantragt werden.

SVMlight wird von *ClearTK* unterstützt. Damit kann die Bibliothek, obwohl sie in C geschrieben wurde, auch leicht in einem Java-Programm über die *ClearTK*-Schnittstelle verwendet werden. Da *ClearTK* die Aufrufe von *SVMlight* über Prozessaufrufe realisiert, muss die Bibliothek vor der Verwendung erst installiert werden. Dazu müssen die für das jeweilige Betriebssystem erstellten Binarys für das Programm zugänglich gemacht werden.

Es wird nur eine binäre Klassifikation unterstützt. Da alle Untersuchungen im Rahmen dieser Arbeit nur genau zwei Klassen benötigen, stellt dies kein Problem dar, sondern macht die Verwendung etwas einfacher.

4.3 Ablauf der Verarbeitung

Der Ablauf der Verarbeitung wird vom *UIMA*-Framework gesteuert. Die einzelnen Verarbeitungsschritte bei der Dokumentenklassifikation werden im Folgenden näher beschrieben. Die Komponenten können ausgewechselt und zum Teil ausgelassen werden. Zum Beispiel können für einen Testlauf ohne Vorverarbeitung die Komponenten für die Subjektivitätsanalyse und die Negationsverarbeitung einfach entfernt werden. Außerdem ist es möglich, an jeder Stelle der Verarbeitung weitere Schritte zu ergänzen. Es kann zum Beispiel nützlich sein, die erstellten Annotationen an einer bestimmten Stelle der Verarbeitung auszugeben. Dazu kann eine der wenigen von *UIMA* mitgelieferten Komponenten verwendet werden, um alle Dokumente mit den Annotationen in ein festgelegtes Verzeichnis zu schreiben.

[5] *SVMlight* ist verfügbar unter: http://www.cs.cornell.edu/People/tj/svm_light/

Das Training für die Dokumentenklassifikation läuft sehr ähnlich ab. Da die Klassifikationskomponente mit anderen Parametern das Training übernimmt, muss nur die Evaluationskomponente entfernt werden.

Das Training und die Auswertung für die Subjektivitätsanalyse laufen ebenfalls ähnlich ab. Für das Einlesen und die Aufteilung in Sätze und Tokens können dieselben Komponenten verwendet werden. Die Negationsverarbeitung wird dafür weggelassen. Für die Feature-Extraktion wird wieder dieselbe Komponente verwendet, nur die Klassifikations- und die Evaluationskomponente werden ausgetauscht.

4.3.1 Einlesen der Dokumente

Das Einlesen der Dokumente ist über einen *Collection Reader* realisiert. Dieser liest alle Dateien aus einem Verzeichnis. Dabei wird rekursiv in alle Unterverzeichnisse gegangen. Die Verzeichnisse der Trainings- und Verifikationsdaten sind so organisiert, dass der Ordnername eines Dokumentes seine Klasse wiedergibt. Das bedeutet, dass zum Beispiel alle positiven Dokumente in dem Ordner „Positiv“ liegen. Vor der weiteren Verarbeitung wird jedes Dokument mit dieser Klasse annotiert. Die Klasse wird ausschließlich für die Evaluation am Ende der Verarbeitung verwendet.

4.3.2 Aufteilen der Dokumente in Sätze

Für das Zerlegen von Dokumenten in Sätze kann ein *Sentence Splitter* verwendet werden. Diese *UIMA*-Komponente wird von der *ClearTK*-Bibliothek aus den *OpenNLP*-Klassen zur Verfügung gestellt. Diese versucht anhand von mitgelieferten Trainingsdaten für die englische Sprache, alle Sätze zu erkennen.

Bei den hier verwendeten Trainingsdaten wurden die Sätze bereits markiert. In den einzelnen Dokumenten befindet sich genau ein Satz pro Zeile. Deswegen ist hier keine Erkennung mehr notwendig und eine Zerlegung ist ausreichend. Für diese Aufgabe habe ich eine eigene Klasse geschrieben.

4.3.3 Einteilung der Sätze in Tokens

Nachdem das Dokument in Sätze aufgeteilt wurde, werden diese weiter zerlegt. Dazu wird jedes Wort mit einer Token-Annotation versehen. Diese Aufgabe wird von einer *OpenNLP*-Klasse übernommen, die über die *ClearTK*-Bibliothek als *UIMA*-Komponente zugänglich ist.

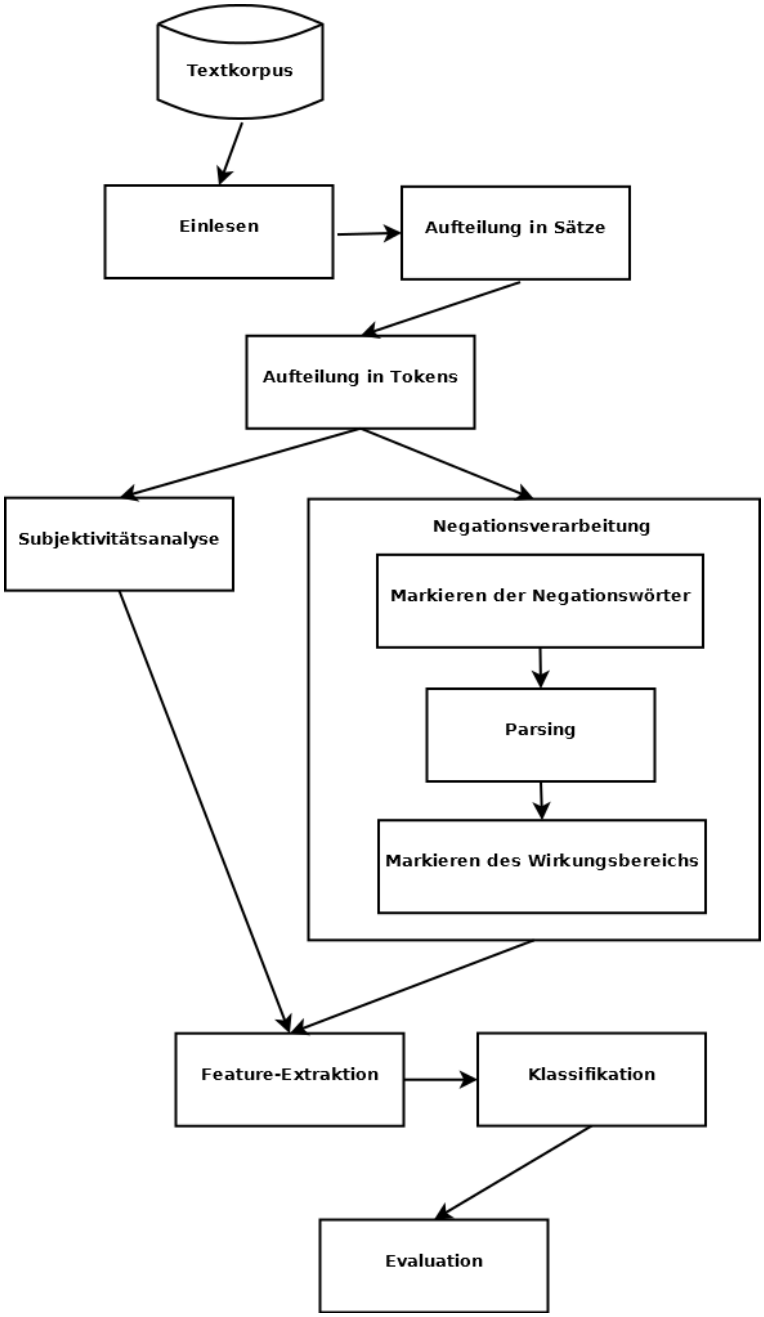


Abbildung 4.2: Ablauf der Dokumentenklassifikation in der Testumgebung

Bei der Aufteilung in Tokens werden die Wörter nicht nur anhand von Leerzeichen zerlegt, auch Sonderzeichen, wie zum Beispiel Kommata, trennen sie. Ein Sonderfall im Englischen stellt die Kombination von Verben und „not“ dar. Zum Beispiel ergibt das Wort „haven't“ die zwei Tokens „have“ und „n't“. Dies ermöglicht es, den zwei Tokens unterschiedliche Wortarten zuzuweisen und diese auch beim Parsing getrennt zu betrachten.

Ein Part-of-Speech-Tagger kann nach der Aufteilung der Sätze in Tokens, jedem Token eine Wortart zuweisen. Diese Information wird beim Parsing verwendet und kann zum Beispiel bei der Feature-Extraktion hilfreich sein.

4.3.4 Subjektivitätsanalyse

Die Subjektivitätsanalyse klassifiziert alle Sätze in eine der beiden Klassen Subjektiv und Objektiv. Das Ergebnis der Klassifikation wird als Eigenschaft in der Satz-Annotation vermerkt. Auf diese Eigenschaft kann dann zum Beispiel bei der Feature-Extraktion zugegriffen werden, um die als objektiv klassifizierten Sätze zu ignorieren.

4.3.5 Negationsverarbeitung

Innerhalb der Negationsverarbeitung werden alle Negationswörter und ihre Wirkungsbereiche markiert. Diese Verarbeitung wird in die im Folgenden erklärten Teilschritte untergliedert.

Markieren der Negationswörter

Im ersten Schritt der Negationsverarbeitung werden alle Negationswörter markiert. Diese werden aus einer vorher definierten Liste von Negationswörtern gelesen, im Text gesucht und dann mit einer eigenen Annotation markiert. Bei diesem Vorgang erhalten alle Sätze, die mindestens eine Negation enthalten, eine weitere Annotation.

Parsing

Alle Sätze, die eine Negation enthalten und im vorangegangenen Schritt markiert wurden, werden geparsed. Für diese Aufgabe wird eine *OpenNLP*-Klasse verwendet, die von *ClearTK* als *UIMA*-Komponente zugänglich gemacht wurde. Das Parsen eines Satzes dauert im Vergleich zu den anderen Verarbeitungsschritten sehr lange, deswegen werden nur die später wirklich benötigten Sätze geparsed. Wenn das folgende Markieren des Wirkungsbereiches nicht auf dem Parsebaum des Satzes basiert, sollte dieser Verarbeitungsschritt entfernt werden.

Markieren des Wirkungsbereiches

An dieser Stelle wird jedem Negationswort in jedem Satz ein Wirkungsbereich zugeordnet. Hierfür gibt es mehrere austauschbare Implementierungen. Eine Möglichkeit ist es, jedem Negationswort eine feste Anzahl direkt folgender Wörter als Wirkungsbereich zuzuordnen. Dazu wird jedes Token, das auf ein Negationswort folgt und sich in dem definierten Fenster befindet, als im Wirkungsbereich der Negation befindlich markiert. Eine andere Möglichkeit ist hier die Markierung des Wirkungsbereiches anhand des Parsebaums des Satzes. Dafür muss im vorherigen Schritt der Satz geparsed worden sein.

4.3.6 Feature-Extraktion

Nachdem alle Vorverarbeitungsschritte abgeschlossen wurden, werden die Features für die Klassifikation gebildet. Dazu können unterschiedliche Feature-Extraktoren eingesetzt werden, zum Beispiel die Bildung von n-Grammen oder Features aus Informationen über die semantische Orientierung bestimmter Satzteile.

4.3.7 Dokumentenklassifikation

Im vorletzten Schritt der Verarbeitung werden die Dokumente auf Basis der vorher extrahierten Features klassifiziert. Dazu wird keine der erstellten Annotationen verwendet. Anschließend an die Klassifikation wird das Dokument mit der Klasse markiert.

4.3.8 Evaluation

Die Evaluation wird in der letzten Komponente vorgenommen. Dazu wird für jedes Dokument das Ergebnis der Klassifikation mit der eigentlichen Klasse, die beim Einlesen dazugeschrieben wurde, verglichen. Die Anzahl der richtig und falsch klassifizierten Dokumente für die beiden Klassen wird gezählt. Auf Basis dieser Zahlen werden die Precision-, Recall- und F-Score-Werte gebildet und ausgegeben. Diese stellen das Ergebnis eines Testlaufs dar.

4.4 Ausgewählte Architektur-Entscheidungen

Im Folgenden wird auf Probleme und deren Lösungen eingegangen, die bei der Erstellung der Architektur aufgetreten sind. Obwohl mit *UIMA*, einem fertigen Framework, gearbeitet wird, resultierten aus der übernommenen Architektur keine Probleme bei der Umsetzung. Dies liegt daran, dass *UIMA* sehr viele Freiheiten bietet und nur sehr wenig über den reinen Ablauf und die Kommunikation der Komponenten hinaus vorschreibt. Leider bietet *UIMA* aber sehr wenig

fertige und benutzbare Komponenten an, weswegen hier *ClearTK* verwendet wird. *ClearTK* bietet sehr viele Komponenten zur Klassifikation von Text und abstrahiert Klassifikatoren und Feature-Extraktoren, die vieles vorgeben. Daraus ergeben sich aber Probleme für die Umsetzung einer Testumgebung mit austauschbaren Komponenten.

4.4.1 Austauschbarkeit der Klassifikatoren

ClearTK definiert eine Schnittstelle für Klassifikatoren, die von allen Adaptern auf die verschiedenen Klassifikator-Bibliotheken verwendet wird. Da die abstrahierten Klassifikatoren allerdings sehr unterschiedlich sind, konnte die Schnittstelle nicht vollständig unabhängig und einheitlich sein. Ein Beispiel hierfür ist der Typ der Klassen. Bei der verwendeten Mallet-Bibliothek ist es möglich, mehr als zwei Klassen zu verwenden. Hier ist der Typ zur Identifikation der Klassen String. *SVMlight* kann nur mit genau zwei Klassen arbeiten, die mit einem Boolean-Wert unterschieden werden. Deswegen ist hier der Ablauf beim Einlesen und der Definition der Ausgangsklasse beim Training und Klassifizieren unterschiedlich. Das macht es schwieriger, die Klassifikatoren auszutauschen, weil die Klassenidentifikationen mit dem Dokument weitergereicht werden müssen.

Deshalb ist es nicht möglich, beide Klassifikatoren mit einer Implementierung der Klassifikationskomponente abzudecken. Für jeden Klassifikator wird also eine eigene Klasse implementiert und die Gemeinsamkeiten in eine abstrakte Klasse ausgelagert. So ist es möglich zu garantieren, dass die Klassifikationskomponente, die Klasse eines Dokumentes immer als String in das *CAS*-Objekt schreibt.

4.4.2 Austauschbarkeit der Feature-Extraktoren

Aufgrund einer Designentscheidung in der *ClearTK*-Bibliothek wird die Feature-Extraktion nicht vom *UIMA*-Framework direkt als eigenständige Komponente aufgerufen. Da zwischen den Komponenten nur über Annotationen auf dem Dokument kommuniziert wird, würde es sehr unübersichtlich und langsam werden, für jedes Feature eine Annotation anzulegen und diese weiterzureichen. Deswegen werden die Features direkt vor der Klassifikation in derselben Komponente wie für die Klassifikation gebildet.

Um die Feature-Extraktion trotzdem modular zu gestalten, werde ich die Klassifikationskomponente so implementieren, dass ihr die Komponente zur Feature-Extraktion übergeben wird. Da *UIMA*-Komponenten keine Klassen übergeben werden können, weil sie aus Deskrip-

toren geladen werden, wird hier der vollständige Klassenname als Parameter übergeben und die Klasse zur Laufzeit vor der Verwendung geladen.

5 Vergleich und Bewertung

5.1 Subjektivitätsanalyse

Kategorie	Art des Tests	Precision	Recall	F-Score
Adjektive	alle Adjektive	51	98	67
	alle Adjektive in *SentiWordNet*	52	96	67
	positive Adjektive	55	77	64
	negative Adjektive	53	67	59
	positive und negative Adjektive	53	93	67
	steigerbare Adjektive	56	81	66
	nicht steigerbare Adjektive	48	45	46
	positive und steigerbare Adjektive	54	90	67
	negative und steigerbare Adjektive	53	90	67
Naive-Bayes	Unigramme	91	94	92
	Bigramme	90	90	90
	Trigramme	82	75	78
	Unigramme, Bigramme und Trigramme	92	93	93
	Präsenz von Wortarten	58	79	67
	n-Gramme und sem. Orientierung	92	94	93
	n-Gramme, sem. Orientierung und Wortarten	92	94	93
SVM linear	Unigramme	90	90	90
	Bigramme	89	85	87
	Trigramme	90	45	60
	Unigramme, Bigramme und Trigramme	91	91	91
	Präsenz von Wortarten	60	81	69
	n-Gramme und sem. Orientierung	90	89	90
	n-Gramme, sem. Orientierung und Wortarten	90	89	89

Tabelle 5.1: Ergebnisse der Auswertung der Subjektivitätsanalyse in Precision, Recall und F-Score. (Alle Angaben in %)

5.1.1 Adjektive

Die Ergebnisse der Auswertung der Klassifikation von Sätzen in subjektiv und objektiv durch das Vorkommen von Adjektiven sind in Tabelle 5.1 aufgeführt.

Obwohl Untersuchungen, wie von Hatzivassiloglou und Wiebe (2000), zeigen, dass ein starker Zusammenhang zwischen bestimmten Adjektiven und Subjektivität besteht, sind die hier erzielten Ergebnisse nicht vergleichbar mit anderen Ergebnissen, wie zum Beispiel den Ergebnissen vom überwachten Lernen. Das beste Ergebnis wird unter anderem bei der Verwendung aller Adjektive erzielt. Dies könnte daran liegen, dass die Menge aller Adjektive die größte der getesteten ist und ein Vorkommen eines Adjektives aus dieser Menge in einem der Sätze damit sehr wahrscheinlich ist. Da auch in objektiven Sätzen häufig Adjektive vorhanden sind, werden deswegen sehr viele Sätze in die Klasse Subjektiv eingeordnet und der Recall-Wert für diese Klasse erreicht auf diese Weise 98%. Die Precision ist mit 51% bei allen Adjektiven allerdings sehr schlecht.

Die Verwendung von entweder positiven oder negativen Adjektiven erhöht leicht den Precision- und senkt stark den Recall-Wert. Der schlechtere Recall-Wert liegt wohl an der deutlich kleineren Anzahl von Adjektiven in der getesteten Menge im Vergleich zu der Menge aller Adjektive. Bei gemeinsamer Verwendung von positiven und negativen Adjektiven sinkt der Recall-Wert nicht so extrem. Interessant ist die Steigerung des Precision-Wertes bei den Tests. Dies könnte damit erklärt werden, dass ein Satz jetzt nur noch als subjektiv eingestuft wird, wenn er ein nicht neutrales Adjektiv enthält. Dieser Effekt könnte noch gesteigert werden, wenn die Anzahl der benötigten Adjektive oder die minimale Polarität eines Adjektives in der Menge erhöht wird. Diese Maßnahme hätte allerdings wieder eine negative Auswirkung auf den Recall-Wert.

Die Menge der steigerbaren Adjektive erreicht im Test ein Ergebnis von 66%. Dies liegt nur leicht unter dem Ergebnis für die Menge aller positiven und negativen Adjektive, obwohl diese Menge deutlich größer ist. Auch der Unterschied zu dem Test mit allen Adjektiven ist nicht sehr groß, obwohl hier die Differenz der Anzahl der Adjektive noch größer ist. Der Precision-Wert ist mit 56% das beste erzielte Ergebnis. Dies zeigt, dass steigerbare Adjektive im Vergleich zu den Adjektiven mit den anderen getesteten Eigenschaften bessere Indikatoren für Subjektivität sind. Der Test mit den nicht steigerbaren Adjektiven zeigt ein sehr schlechtes Ergebnis. Dies liegt aber vor allem daran, dass die Anzahl der Adjektive in dieser Menge sehr klein ist. Dies führt dazu, dass sehr viele Sätze in die Klasse Objektiv einsortiert werden und

das verringert den Recall-Wert stark. Die Tests mit der Menge der steigerbaren Adjektive zusammen mit den positiven oder negativen Adjektivmengen erreichen ebenfalls einen ähnlichen Wert wie der Test mit steigerbaren Adjektiven. Sehr viele der positiven oder negativen Wörter sind steigerbar, deswegen ist es nicht verwunderlich, dass der Precision-Wert hier ähnlich ausfällt und wegen der erhöhten Anzahl von Adjektiven in der Menge der Recall-Wert steigt.

Aufgrund der schlechten Ergebnisse bei der Klassifikation ist eine Verwendung dieses Ansatzes in der Praxis nicht zu empfehlen. Allerdings kann das Wissen über den Zusammenhang von bestimmten Adjektivgruppen und Subjektivität als Ergänzung und zur Verbesserung anderer Verfahren verwendet werden.

5.1.2 Überwachtes Lernen

Naive-Bayes-Klassifikator

Die besten Ergebnisse bei der Subjektivitätsanalyse erzielt der Ansatz des überwachten Lernens mit einem Naive-Bayes-Klassifikator. Hier erreicht die Klassifikation F-Score-Werte von über 90%. Dies sind äußerst gute Werte. Ein Wert von 100%, also eine vollständig korrekte Klassifikation, ist kaum möglich. Selbst ein Mensch schafft dies nicht, weil die Entscheidung, ob ein Satz subjektiv oder objektiv ist, nicht eindeutig ist. Dies hat mit der Mehrdeutigkeit der Sprache zu tun. Der gleiche Satz kann in einem anderen Kontext oder von verschiedenen Menschen anders verstanden werden. Durch dieses Problem kann auch davon ausgegangen werden, dass die per Hand erstellten Trainingsdaten nicht vollständig korrekt sind. Somit kann von dem Klassifikationsalgorithmus keine vollständige Korrektheit erwartet werden.

Bereits der einfachste Ansatz, der nur mit Unigrammen arbeitet, erreicht einen Precision-, Recall- und F-Score-Wert von über 90%. Dies zeigt, warum diese Art der Featurebildung häufig verwendet wird. Da jedes Wort direkt als Feature verwendet wird, ist die Umsetzung sehr leicht. Bigramme erreichen schlechtere Klassifikationsergebnisse. Noch schlechtere Ergebnisse werden mit Trigrammen erzielt. Bei der Verwendung von n-Grammen ist bei steigendem n also eine schlechter werdende Klassifikation zu erwarten. Dies liegt daran, dass die Wahrscheinlichkeit, ein n-Gramm mehr als x-mal im Textkorpus zu finden, mit steigendem n stark sinkt. Es wird also schwieriger für den Klassifikator, Zusammenhänge zwischen dem Auftreten bestimmter Features und einer Klassenzuordnung zu finden.

Die gemeinsame Verwendung von Unigrammen, Bigrammen und Trigrammen erreicht bei der ausschließlichen Verwendung von n-Grammen im Test die besten Ergebnisse. Der Effekt der seltener werdenden Features der längeren n-Gramme wird durch die Unigramme ausgeglichen. Dafür werden für den Klassifizierer Zusammenhänge, die aus mehreren aufeinanderfolgenden Wörtern bestehen und auf Subjektivität hindeuten, erkennbar. Bei der Klassifikation stehen mehr Informationen bereit als bei der ausschließlichen Verwendung von Unigrammen. Dadurch können die Ergebnisse verbessert werden.

Die Benutzung von binären Features, die die Präsenz von bestimmten Wortarten anzeigen, erzielt nur mäßige Ergebnisse. Wiebe u. a. (1999) verwenden diesen Ansatz gemeinsam mit Features, die beschreiben, wie oft ein Wort in einer der beiden möglichen Klassen vorkommt. Auf diese Weise wurde ein Ergebnis von über 70% erreicht. An den Ergebnissen dieses Tests lässt sich aber schon erkennen, dass das Vorhandensein oder die Abwesenheit bestimmter Wortarten auf Subjektivität hindeutet. Die Features alleine eignen sich aber nicht für eine Klassifikation. Sie sollten zusammen mit anderen Ansätzen verwendet werden.

Die Ergänzung von Features aus n-Grammen durch Features, die die semantische Orientierung aufeinanderfolgender Wörter und die durchschnittliche semantische Orientierung aller Wörter in dem untersuchten Satz repräsentieren, erreicht im Test keine erkennbare Verbesserung. Der Recall-Wert steigt leicht, aber die beiden anderen Werte bleiben gleich. Die semantische Orientierung der Wörter in einem Satz kann ein Indikator für Subjektivität sein. Diese zusätzlichen Informationen sollten es dem Klassifikator ermöglichen, eine bessere Entscheidung zu treffen und das Ergebnis verbessern.

Der letzte Test kombiniert Features aller vorangegangenen Untersuchungen. Der Test zeigt, dass sich das Ergebnis durch die Kombination im Vergleich zu allen anderen Tests nicht weiter verbessert hat. Dies zeigt, dass der Klassifikator zusätzliche Informationen, die etwas mit der Subjektivität des Satzes zu tun haben, unabhängig von den bereits existierenden Features, nutzen kann oder, dass so zumindest kein schlechteres Ergebnis erzielt wird. Die weitere Suche nach guten Indikatoren für Subjektivität könnte also dazu führen, eine noch bessere Klassifikation zu erreichen.

SVM

Die Ergebnisse der Subjektivitätsanalyse mit Support-Vektor-Maschinen sind ähnlich zu den Ergebnissen mit einem Naive-Bayes-Klassifikator. Bei der Verwendung von n-Grammen sind

etwas schlechtere Ergebnisse zu beobachten. Besonders bei der ausschließlichen Verwendung von Bigrammen oder Trigrammen fällt auf, dass SVMs hier bei eher selten vorkommenden Features schlechtere Ergebnisse liefern. Die gemeinsame Verwendung von Unigrammen, Bigrammen und Trigrammen zeigt wie beim Test mit dem Naive-Bayes-Klassifikator eine Verbesserung im Vergleich mit Unigrammen.

Die Klassifikation durch Features, die die Präsenz von bestimmten Wortarten im Satz anzeigen, erzielt mit SVMs leicht bessere Ergebnisse. Mit 69% sind die Ergebnisse hier allerdings immer noch deutlich schlechter als im Vergleich zu den anderen Ansätzen.

Das Hinzufügen von Features, die die semantische Orientierung oder die Präsenz bestimmter Wortarten anzeigen, zu den Features aus n-Grammen führt bei der Verwendung von SVMs zu einer leichten Verschlechterung. Dies kann daran liegen, dass diese Features nicht viel mit den Features aus n-Grammen gemeinsam haben und deswegen für SVMs eher hinderlich sein können. Support-Vektor-Maschinen versuchen die Features der beiden Klassen durch eine Funktion zu trennen. Durch das Hinzufügen anderer Featurearten kann es schwieriger werden, die Klassen ohne angepasste Kernfunktion voneinander abzugrenzen.

Die aufgelisteten Ergebnisse wurden mit einer linearen Kernfunktion erzielt. Verschiedene Tests mit komplexeren Kernfunktionen ergaben auch bei den Kombinationen verschiedener Featurearten keine Verbesserung. Durch eine genauere Untersuchung und Optimierung der Parameter für die Support-Vektor-Maschinen ist sicherlich noch eine Verbesserung der Werte möglich. Besonders bei den späteren Tests verschiedener Featurekombinationen sollte es durch eine passende Kernfunktion möglich sein, zumindest gleich gute Ergebnisse wie bei den n-Grammen zu erreichen.

5.1.3 Auswirkung auf die Dokumentenklassifikation

Die Tabelle 5.2 zeigt die Precision-, Recall- und F-Score-Werte der Dokumentenklassifikation mit und ohne Subjektivitätsanalyse. Die Werte zeigen, dass die Ansätze, die die besten Ergebnisse bei der Subjektivitätsklassifikation erzielen, auch die stärkste Verbesserung der Dokumentenklassifikation erreichen. Interessanterweise zeigt die Verwendung von Unigrammen die besten Ergebnisse, obwohl die Ansätze mit der semantischen Orientierung und der Präsenz bestimmter Wortarten bei der Subjektivitätsanalyse bessere Ergebnisse erzielten. Das bedeutet, dass eine genauere Klassifikation der Sätze in subjektiv und objektiv ein guter Indikator für eine Verbesserung der Dokumentenklassifikation ist, aber keine Verbesserung garantieren kann.

Kategorie	Art des Tests	Precision	Recall	F-Score
ohne Vorverarbeitung	-	80,4	78,0	79,1
Naive-Bayes	Unigramme	83,8 (+3,4)	81,0 (+3)	82,4 (+3,3)
	Unigramme, Bigramme und Trigramme	83,7 (+3,3)	80,3 (+2,3)	82,0 (+2,9)
	Präsenz von Wortarten	81,1 (+ 0,7)	78,7 (+0,7)	79,9 (+0,8)
	n-Gramme, sem. Orientierung und Wortarten	82,7 (+2,3)	81,0 (+3)	81,8 (+2,7)

Tabelle 5.2: Auswirkung ausgewählter Konzepte zur Subjektivitätsanalyse auf die Dokumentenklassifikation. (Alle Angaben in %)

Es sollte also neben einer Untersuchung der Satzklassifikation auch immer die Auswirkung auf die Dokumentenklassifikation untersucht werden.

Obwohl die Subjektivitätsanalyse auf Basis der Präsenz bestimmter Wortarten mit einem F-Score von 67% nur mäßige Ergebnisse erzielte, sind die Auswirkungen auf die Dokumentenklassifikation durchaus positiv und erreichen ungefähr ein Drittel der Verbesserung durch die anderen Konzepte. Ein schlechterer Ansatz zur Vorverarbeitung erreicht hier also ein positives Ergebnis. Dies zeigt, dass solche Ansätze ebenfalls in Betracht gezogen werden können, wenn keine bessere Alternative vorhanden sein sollte und sich ein Testdurchlauf durchaus lohnen kann.

5.2 Negationsverarbeitung

5.2.1 Feste Fenstergröße

Die Negationsverarbeitung mit einer festen Fenstergröße hat positiven Einfluss auf die Dokumentenklassifikation. Dies zeigen die Precision-, Recall- und F-Score-Werte aus der Untersuchung, die in Tabelle 5.3 abgebildet sind.

Bei dem ersten Test wurde nur „not“ als Negationswort genommen. Es war zu erwarten, dass die Auswirkungen auf die Klassifikation nicht sehr groß sein würden, weil auf diese Weise nur sehr wenige der im Text vorhandenen Negationen erkannt werden. Dennoch lässt sich aus den Ergebnissen eine positive Auswirkung erkennen. Die Ergebnisse mit verschiedener Fenstergröße sind sehr ähnlich. Die steigende Fenstergröße bewirkt, dass mehr Wörter hinter dem Negationswort dem Wirkungsbereich zugeordnet werden. Bei einem größeren Fenster werden also mehr Features verändert und die Auswirkung sollte größer sein. Dies ist aber nicht

Kategorie	Negationswörter	Art des Tests	Precision	Recall	F-Score
feste Fenstergröße	„not“	Fenstergröße 1	80,8 (+0,4)	78,7 (+0,7)	79,7 (+0,6)
		Fenstergröße 2	80,3 (-0,1)	78,7 (+0,7)	79,5 (+0,4)
		Fenstergröße 3	80,5 (+0,1)	78,7 (+0,7)	79,6 (+0,5)
		Fenstergröße 4	81,4 (+1)	79,0 (+1)	80,2 (+1,1)
		Fenstergröße 5	80,7 (+0,3)	79,3 (+1,3)	80,0 (+0,9)
	Tabelle 3.1	Fenstergröße 1	80,5 (+0,1)	79,7 (+1,7)	80,1 (+1)
		Fenstergröße 2	81,1 (+0,7)	80,3 (+2,3)	80,7 (+1,6)
		Fenstergröße 3	80,6 (+0,2)	80,3 (+2,3)	80,5 (+1,4)
		Fenstergröße 4	81,3 (+0,9)	79,7 (+1,7)	80,5 (+1,4)
		Fenstergröße 5	81,1 (+0,7)	80,3 (+2,3)	80,7 (+1,6)
Parsing	„not“	vorläufiger Wirkungsbereich	79,8 (-0,6)	79,0 (+1)	79,4 (+0,3)
		Begrenzer	81,1 (+0,7)	78,7 (+0,7)	79,9 (+0,8)
		Begrenzer und Ausnahmen	80,9 (+0,5)	79,0 (+1)	79,9 (+0,8)
	Tabelle 3.1	vorläufiger Wirkungsbereich	80,5 (+0,1)	80,0 (+2)	80,3 (+1,2)
		Begrenzer	81,1 (+0,7)	80,3 (+2,3)	80,7 (+1,6)
		Begrenzer und Ausnahmen	80,6 (+0,2)	80,3 (+2,3)	80,5 (+1,4)

Tabelle 5.3: Auswirkung der Negationsverarbeitung auf die Dokumentenklassifikation. (Alle Angaben in %)

zu erkennen, die Werte schwanken nur leicht bei unterschiedlicher Fenstergröße. Die Werte zeigen die Verbesserung der Dokumentenklassifikation und nicht wie genau die Negationsverarbeitung arbeitet, deswegen kann es sein, dass die Auswirkungen hier nicht erkennbar sind.

Mit einer deutlich umfassenderen Liste von Negationswörtern kann die Verbesserung der Klassifikation nochmal um etwa die Hälfte verbessert werden. Durch eine längere Liste von Negationswörtern können mehr Negationen erkannt und verarbeitet werden. In dem verwendeten Textkorpus werden insgesamt ungefähr 5.300 Negationen gefunden, die durch „not“ eingeleitet werden. Im Vergleich dazu führt eine Suche mit der gesamten verwendeten Liste zu ungefähr 15.100 Negationen. Die besseren Ergebnisse zeigen, dass die Verarbeitung der Negationen positive Auswirkung auf die Klassifikation hat, obwohl mit einer festen Fenstergröße gearbeitet wird.

An den geringen Unterschieden zwischen den einzelnen Werten kann man erkennen, dass die gewählte Fenstergröße in dem getesteten Bereich keine großen Auswirkungen hat. Außerdem wirkt sich eine längere Liste von Negationswörtern und damit eine umfassendere Negationsverarbeitung positiv aus. Diese Ergebnisse finden sich auch in anderen Untersuchungen wieder, zum Beispiel bei Dadvar u. a. (2011).

5.2.2 Parsing

Mit Informationen über die grammatikalische Struktur eines Satzes soll die Negationsverarbeitung weiter verbessert werden. Die Negationsverarbeitung mit der allgemeinen Regel zur Bestimmung des vorläufigen Wirkungsbereiches zeigt schon eine leichte Verbesserung bei dem Negationswort „not“ und eine Verbesserung von über einem Prozentpunkt bei der Verwendung aller Negationswörter gegenüber dem Test ohne Vorverarbeitung. Die Verwendung von Begrenzern für die vorläufigen Wirkungsbereiche verbessert die Ergebnisse. Die Berücksichtigung der Ausnahmen für Ausdrücke, die das Negationswort enthalten, und rhetorische Fragen verschlechtern das Ergebnis.

Die verwendete Regel ist sehr allgemein. Es wird nicht zwischen verschiedenen Negationswörtern unterschieden oder wirklich mit grammatikalischen Regeln gearbeitet. Trotzdem erreicht diese Art der Verarbeitung schon eine Verbesserung gegenüber einem Test ohne Vorverarbeitung. Im Vergleich zu einer Negationsverarbeitung mit fester Fenstergröße hingegen ist sie deutlich schlechter. Dies liegt daran, dass der vorläufige Wirkungsbereich bei vielen Negationen recht groß ist und oft bis zum Satzende reicht. Deswegen werden viele Features, die nicht zum wirklichen Wirkungsbereich gehören, durch die Negationsverarbeitung verändert.

Durch die Verwendung der Begrenzer werden vergleichbare Ergebnisse wie bei der Verarbeitung mit fester Fenstergröße erreicht. Eine Verbesserung, die man aufgrund der Berücksichtigung grammatikalischer Informationen erwartet, ist allerdings hier nicht zu erkennen. Durch die Behandlung von Ausnahmen wird das erreichte Ergebnis etwas verschlechtert. Dies könnte ein Zeichen dafür sein, dass durch die Ausnahmen falsche Sätze aus der Verarbeitung entfernt wurden. Hier wurden nur einige der von Jia u. a. (2009) vorgestellten Regeln zur Eingrenzung des vorläufigen Wirkungsbereiches eingesetzt. Weitere Regeln könnten die Verarbeitungsgenauigkeit erhöhen und die Ergebnisse weiter verbessern.

Der verwendete Parser schafft es nicht, alle Sätze mit Negationen zu verarbeiten. Diese Sätze werden bei diesem Test durch die Negationsverarbeitung ignoriert und ergeben vollstän-

dige, unveränderte Features. Bei dem verwendeten Textkorpus ist die Anzahl der Sätze, die nicht verarbeitet werden können, sehr gering und liegt unter 1%. Es muss beachtet werden, dass die Sätze im Textkorpus Filmbewertungen von Benutzern sind. Der Satzbau ist hier oft fehlerhaft und einige Wörter sind Abkürzungen oder Umgangssprache. Die Auswirkungen dieser fehlenden Sätze sollten sehr gering sein. Bei diesen Sätzen könnte auf eine Verarbeitung mit fester Fenstergröße zurückgegriffen oder ein anderer Parser verwendet werden.

Ein Nachteil bei dieser Art der Verarbeitung ist die Geschwindigkeit. Das Parsen eines Satzes dauert im Vergleich zu den anderen Verfahren sehr lange.

Die Genauigkeit der Negationsverarbeitung lässt sich eindeutig auf diese Art und Weise verbessern. Dies zeigen auch Untersuchungen wie von Apostolova u. a. (2011) und Jia u. a. (2009). Allerdings sind die Auswirkungen auf die Dokumentenklassifikation durch überwachtes Lernen sehr gering und die Verarbeitung dauert verhältnismäßig sehr viel länger. Deswegen sollte hier wohl vorerst mit einer Näherung der Verarbeitung ohne Parsing-Informationen gearbeitet werden.

5.3 Auswirkung der Vorverarbeitungsschritte auf die Dokumentenklassifikation

Art des Tests	**Precision**	**Recall**	**F-Score**
ohne Vorverarbeitung	80,4	78,0	79,1
mit Subjektivitätsanalyse (Naive-Bayes, Unigramme)	83,8 (+3,4)	81,0 (+3)	82,4 (+3,3)
mit Negationsverarbeitung (Tabelle 3.1, Fenstergröße 2)	81,1 (+0,7)	80,3 (+2,3)	80,7 (+1,6)
mit Subjektivitätsanalyse und Negationsverarbeitung	85,8 (+5,4)	82,7 (+4,7)	84,2 (+5,1)

Tabelle 5.4: Ergebnisse der Dokumentenklassifikation mit unterschiedlichen Vorverarbeitungsschritten. (Alle Angaben in %)

Die Tabelle 5.4 zeigt die Auswirkungen der Subjektivitätsanalyse und der Negationsverarbeitung auf die Dokumentenklassifikation im Vergleich. Die Subjektivitätsanalyse erzielt dabei eine stärkere Verbesserung. Die stärkere Auswirkung der Subjektivitätsanalyse gegenüber der Negationsverarbeitung lässt sich unter anderem mit dem stärkeren Eingreifen in die Klassifikation erklären. Durch die Subjektivitätsanalyse werden alle Features aus objektiven Sätzen vor der Klassifikation entfernt. Dies sind deutlich mehr Features als die, die von der Ne-

gationsverarbeitung verändert werden, weil sie in dem Wirkungsbereich einer Negation liegen.

Die Ergebnisse der Subjektivitätsanalyse, also die Einteilung der Sätze in subjektiv und objektiv, lassen sich sehr gut auf die Features abbilden, indem nur Features aus subjektiven Sätzen gebildet werden. Dabei gehen keine Informationen über möglicherweise wichtige Merkmale in den subjektiven Sätzen verloren. Bei der Negationsverarbeitung hingegen ist die Repräsentation der Ergebnisse der Vorverarbeitung in den Features ein Problem. Liegt ein Wort in dem Wirkungsbereich einer Negation, wird seine semantische Orientierung negiert. Dies lässt sich nicht in den Features abbilden. Es werden lediglich alle Features von negierten Wörtern verändert. Bei der Klassifikation werden diese dann wie vollkommen andere Features behandelt. Die Information, dass dieses Feature wahrscheinlich ein gleich starker Indikator für eine Klassifikation in eine der Klassen ist, geht verloren. Dazu müsste in die Klassifikation selbiger eingegriffen werden und es müsste ermöglicht werden, diese Beziehung zwischen entgegengesetzten Features abzubilden.

Der letzte Test in Tabelle 5.4 zeigt die Ergebnisse der Klassifikation, wenn sowohl die Subjektivitätsanalyse als auch die Negationsverarbeitung verwendet werden. Obwohl die Auswirkung der Negationsverarbeitung durch die herausgefilterten objektiven Sätze der Subjektivitätsanalyse abgeschwächt werden muss, ist die Verbesserung etwas größer als die Verbesserung durch die einzelnen Vorverarbeitungsschritte. Daran lässt sich erkennen, dass sich die beiden Vorverarbeitungsschritte gut zusammen einsetzen lassen und sich gegenseitig nicht behindern. Durch Hinzufügen weiterer Vorverarbeitungsschritte könnten die Ergebnisse noch weiter verbessert werden.

6 Schlussbetrachtung

6.1 Zusammenfassung

In dieser Arbeit wurden viele verschiedene Ansätze von zwei Vorverarbeitungskonzepten für *Sentiment Analysis* durch überwachtes Lernen zusammengetragen und vorgestellt. Die Ansätze wurden in einer selbsterstellten auf *UIMA* basierenden Testumgebung getestet und somit wurde die Möglichkeit geschaffen, die erzielten Ergebnisse miteinander zu vergleichen und auszuwerten. Ein Vergleich der Ansätze war bisher nicht möglich, weil sie mit unterschiedlichen Daten und Verfahren bewertet wurden.

Die hier vorgestellten Konzepte können dazu eingesetzt werden, die Ergebnisse der *Sentiment Analysis* in der Klassifikation von Meinungen und auch in anderen Bereichen deutlich zu verbessern. Die entwickelte Testumgebung kann frei verwendet und bei Bedarf angepasst werden, um die Auswirkungen der Verfahren bei anderen Texten oder bei anderen Anwendungsfällen zu testen.

Bei der *Sentiment Analysis* geht es darum, die Meinungen in Texten zu analysieren. Diese Arbeit konzentriert sich auf den Anwendungsfall der Klassifizierung von Dokumenten in die Klassen Positiv und Negativ aufgrund der enthaltenen Meinung.

Im Rahmen der Arbeit werden zwei Konzepte für die Vorverarbeitung vorgestellt und verschiedene Ansätze für die Umsetzung im Detail erläutert. Das erste Konzept ist die Subjektivitätsanalyse. Hier wird versucht, in einem Dokument für alle Sätze zu bestimmen, ob sie subjektiv oder objektiv sind. Für eine Klassifikation nach Meinungen sind eigentlich nur die Textteile des Dokuments relevant, die auch Meinungen enthalten. Durch die Subjektivitätsanalyse wird es möglich, die objektiven Sätze bei der Klassifikation zu ignorieren. Ein guter Indikator für Subjektivität sind Adjektive. Es gibt verschiedene Arten von Adjektiven, wie zum Beispiel positive, negative, steigerbare und nicht steigerbare. Mengen von Adjektiven mit verschiedenen Eigenschaften können unterschiedlich stark auf Subjektivität hindeuten. Ein

weiterer Ansatz zur Klassifikation von Subjektivität ist der Einsatz von überwachtem Lernen. Für diese Klassifikation ist es wichtig, die richtigen Features auszuwählen. Dafür wurden in der Testumgebung verschiedene Featurekombinationen aus n-Grammen, der Präsenz bestimmter Wortarten im Satz und Informationen über die semantische Orientierung von Wörtern und Wortgruppen getestet.

Die Klassifikation durch überwachtes Lernen erzielte die besten Ergebnisse für die Subjektivitätsanalyse. Durch Verwendung von einfachen n-Grammen lässt sich ein F-Score-Wert von über 90% erreichen. Dies ist ein sehr gutes Ergebnis, weil hier natürliche Sprache verarbeitet wird. Selbst ein Mensch würde bei der Klassifikation keinen Wert von 100% erreichen, weil die Sprache mehrdeutig ist und ein Satz von verschiedenen Menschen unterschiedlich interpretiert werden kann. Die durchgeführten Tests haben gezeigt, dass sich durch die Subjektivitätsanalyse, die Ergebnisse der Dokumentenklassifikation spürbar verbessern lassen. In diesem Test ergab sich eine Verbesserung von bis zu 3 Prozentpunkten im Vergleich zu einem Durchlauf ohne Vorverarbeitung. Im Bereich eines Ergebnisses, welches schon bei etwa 80% liegt, stellt dies eine starke Verbesserung dar.

Das zweite vorgestellte Konzept für die Vorverarbeitung ist die Negationsverarbeitung. Negationen kehren die semantische Orientierung eines oder mehrere Wörter um. Bei der Negationsverarbeitung wird zuerst das Wort gesucht, welches die Negation bildet. So ein Wort wird Negationswort genannt und kann zum Beispiel durch eine Liste von Negationswörtern im Text identifiziert werden. Nach der Identifikation des Negationswortes wird versucht herauszufinden, welche Wörter des Satzes von der Negation beeinflusst werden. Diese Menge von Wörtern liegt im Wirkungsbereich der Negation und wird für spätere Verarbeitungsschritte markiert. Die verschiedenen Ansätze für die Negationsverarbeitung unterscheiden sich vor allem in der Abgrenzung des Wirkungsbereiches. Ein einfacher, aber durchaus erfolgversprechender Ansatz zur Abgrenzung des Wirkungsbereiches einer Negation ist es, eine feste Anzahl n von Wörtern hinter dem Negationswort zu markieren. Im Rahmen dieser Arbeit wurden Fenstergrößen von 1 bis 5 getestet und gute Ergebnisse mit einer Verbesserung von etwa 1,5 Prozentpunkten erreicht.

Ein weiterer Ansatz ist die Nutzung der grammatikalischen Struktur des Satzes. Dazu wird ein Parser für natürliche Sprache eingesetzt. Auf diese Weise ist es möglich, den Wirkungsbereich genauer abzugrenzen. In dem für den Test in dieser Arbeit verwendeten Ansatz wird versucht, mit einer allgemeinen Regel aus dem Parsebaum einen vorläufigen Wirkungsbereich abzugren-

zen. Dieser wird dann im zweiten Schritt durch weitere Regeln verkleinert. Die Ergebnisse dieses Ansatzes erreichten ähnliche Werte wie der Test mit fester Fenstergröße. Weitere und verbesserte Regeln könnten hier zu besseren Ergebnissen führen.

Für die Untersuchungen der Auswirkungen der einzelnen Vorverarbeitungsschritte wurde im Laufe der Anfertigung dieser Arbeit eine Testumgebung erstellt. Die Architektur der Testumgebung basiert auf *UIMA*. Dies ermöglicht eine hohe Austauschbarkeit der einzelnen Verarbeitungskomponenten, da die Kommunikation und die Ablaufsteuerung von *UIMA* übernommen werden. Dadurch ist es möglich, mit wenig Aufwand weitere Untersuchungen von anderen Vorverarbeitungsschritten, Klassifikationsalgorithmen oder Tests mit anderen Texten durchzuführen. Die Testumgebung und alle verwendeten externen Bibliotheken können für den Einsatz in der Forschung frei verwendet werden. Da alle Ansätze mit derselben Testumgebung und denselben Daten getestet wurden, ist es erstmals möglich, die Ergebnisse der einzelnen Ansätze direkt zu vergleichen.

Durch die zwei vorgestellten Konzepte ist es gelungen, eine Verbesserung der *Sentiment Analysis* von Dokumenten bei den verwendeten Filmbewertungen um insgesamt 5 Prozentpunkte zu erreichen. Die Konzepte können für die Klassifikation von Meinungen und andere *Sentiment Analysis*-Anwendungen verwendet werden. Aufgrund der Ergebnisse sind diese in der Praxis gut einsetzbar. Für den Praxiseinsatz sollte eine kurze Untersuchung mit einem Textkorpus aus der Domäne der Anwendung gemacht werden. Hierfür kann die Testumgebung, die im Rahmen dieser Arbeit erstellt wurde, verwendet werden.

6.2 Ausblick

An den hier vorgestellten Konzepten und einzelnen Ansätzen können an vielen Stellen weitere Untersuchungen durchgeführt werden. So besteht die Möglichkeit, die einzelnen Ansätze weiter zu verbessern und damit die Ergebnisse der *Sentiment Analysis* positiv zu beeinflussen. Außerdem könnten weitere Konzepte im Zusammenhang mit den hier beschriebenen untersucht werden. Im Folgenden werden einige Ideen für weitere Untersuchungen vorgestellt.

Viele der vorgestellten Ansätze für die Vorverarbeitungskonzepte könnten durch die Unterscheidung zwischen unterschiedlichen Wortbedeutungen verbessert werden. Weil verschiedene Wortbedeutungen unterschiedlich stark auf eine Klasse hindeuten können, könnten hier die Ergebnisse der Subjektivitätsanalyse wahrscheinlich weiter verbessert werden.

Eine Fortsetzung der Untersuchungen könnte darin bestehen, die Auswirkung der Vorverarbeitungsschritte auf andere Klassifikationsalgorithmen zu testen. Dies könnten andere überwachte Lernverfahren oder andere Klassifikationsverfahren sein. Es wäre auch möglich, die Vorverarbeitungsschritte, die ja beide schon auf Satzebene arbeiten, für die Verbesserung einer Klassifikation von Meinungen auf Satzebene einzusetzen.

Diese Arbeit untersucht nur die Auswirkungen auf die Klassifikation von Filmbewertungen. Andere Textkorpora könnten weitere Erkenntnisse über die vorgestellten Vorverarbeitungsschritte erbringen.

Literaturverzeichnis

[Akkaya u. a. 2009] Akkaya, Cem ; ; Wiebe, Janyce M. ; Mihalcea, Rada: Subjectivity word sense disambiguation. In: *EMNLP '09 Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1* Bd. Volume 1, 2009

[Alpaydin 2008] Alpaydin, Ethem: *Maschinelles Lernen.* Oldenbourg, 2008

[Apostolova u. a. 2011] Apostolova, Emilia ; Tomuro, Noriko ; Demner-Fushman, Dina: Automatic extraction of lexico-syntactic patterns for detection of negation and speculation scopes. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, Association for Computational Linguistics, 2011, S. 283–287

[Banea u. a. 2011] Banea, Carmen ; Mihalcea, Rada ; Wiebe, Janyce M.: Multilingual Sentiment and Subjectivity Analysis, 2011

[Bruce und Wiebe 1999] Bruce, Rebecca F. ; Wiebe, Janyce M.: Recognizing Subjectivity: A Case Study of Manual Tagging. In: *Natural Language Engineering* 5 (1999), S. 187–205

[Carvalho u. a. 2009] Carvalho, Paula ; Sarmento, Luís ; Silva, Mário J. ; Oliveira, Eugénio de: Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In: *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, 2009, S. 53–56

[Councill u. a. 2010] Councill, Isaac G. ; McDonald, Ryan ; Velikovich, Leonid: What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In: *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, Association for Computational Linguistics, 2010, S. 51–59

[Dadvar u. a. 2011] Dadvar, Maral ; Hauff, Claudia ; de, Franciska J.: Scope of negation detection in sentiment analysis. In: *DIR 2011: Dutch-Belgian Information Retrieval Workshop Amsterdam*, University of Amsterdam, 2011, S. 16–20

[Dalal und Zaveri 2011] Dalal, Mita K. ; Zaveri, Mukesh A.: Automatic Text Classification: A Technical Review. In: *International Journal of Computer Applications (0975 – 8887)*, 2011

[Ertel 2009] Ertel, Wolfgang: *Grundkurs Künstliche Intelligenz*. Vieweg+Tuebner, 2009

[Esuli und Sebastiani 2006] Esuli, Andrea ; Sebastiani, Fabrizio: SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In: *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06*, 2006, S. 417–422

[Fellbaum 1998] Fellbaum, Christiane (Hrsg.): *WordNet An Electronic Lexical Database*. The MIT Press, 1998

[Görz u. a. 2003] Görz, Günther ; Rollinger, Claus-Rainer ; Schneeberger, Josef: *Handbuch der Künstlichen Intelligenz*. Oldenbourg, 2003

[Hatzivassiloglou und McKeown 1997] Hatzivassiloglou, Vasileios ; McKeown, Kathleen R.: Predicting the Semantic Orientation of Adjectives. In: *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 1997, S. 174–181

[Hatzivassiloglou und Wiebe 2000] Hatzivassiloglou, Vasileios ; Wiebe, Janyce M.: Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In: *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2000

[Jia u. a. 2009] Jia, Lifeng ; Yu, Clement ; Meng, Weiyi: The effect of negation on sentiment analysis and retrieval effectiveness. In: *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, S. 1827–1830. – ISBN 978-1-60558-512-3

[Liu 2010] Liu, Bing: Sentiment Analysis and Subjectivity. In: Indurkhya, Nitin (Hrsg.) ; Damerau, Fred J. (Hrsg.): *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, 2010

[Pang und Lee 2008] Pang, Bo ; Lee, Lillian: *Opinion mining and sentiment analysis*. Now Publishers Inc, 2008

[Pang u. a. 2002] Pang, Bo ; Lee, Lillian ; Vaithyanathan, Shivakumar: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002, S. 79–86

[Princeton University 2010] Princeton University: *About WordNet*. http://wordnet.princeton.edu. 2010

[Quirk u. a. 1985] Quirk, Randolph ; Greenbaum, Sidney ; Leech, Geoffry ; Svartvik, Jan: *A Comprehensive Grammar of the English Language*. Longman, 1985

[Szarvas u. a. 2008] Szarvas, György ; Vincze, Veronika ; Farkas, Richárd ; Csirik, János: The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, Association for Computational Linguistics, 2008, S. 38–45

[The Apache UIMA Development Community 2011] The Apache UIMA Development Community: *UIMA Documentation*. http://uima.apache.org/d/uimaj-2.4.0/index.html. December 2011

[Wiebe u. a. 1999] Wiebe, Janyce M. ; Bruce, Rebecca F. ; O'Hara, Thomas P.: Development and use of a gold standard data set for subjectivity classifications. In: *Proceedings of the Association for Computational Linguistics (ACL)*, 1999, S. 246–253

[Wiebe und Mihalcea 2006] Wiebe, Janyce M. ; Mihalcea, Rada: Word sense and subjectivity. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2006

[Wiegand u. a. 2010] Wiegand, Michael ; Balahur, Alexandra ; Roth, Benjamin ; Klakow, Dietrich ; Montoyo, Andrés: A survey on the role of negation in sentiment analysis. In: *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, Association for Computational Linguistics, 2010, S. 60–68

[Wilson u. a. 2005] Wilson, Theresa A. ; Hoffmann, Paul ; Somasundaran, Swapna ; Kessler, Jason ; Wiebe, Janyce M. ; Choi, Yejin ; Cardie, Claire ; Riloff, Ellen ; Patwardhan, Siddharth: OpinionFinder : A system for subjectivity analysis. In: *October* (2005), S. 34–35

[Yessenov und Misailovic 2009] Yessenov, Kuat ; Misailovic, Sasa: Sentiment Analysis of Movie Review Comments. In: *Report on Spring 2009 final project*, 2009

[Yu und Hatzivassiloglou 2003] Yu, Hong ; Hatzivassiloglou, Vasileios: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, 2003, S. 129–136

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 21. August 2012 Tobias Eichler