

Melanie Siegel
Melpomeni Alexa

Sentiment-Analyse deutschsprachiger Meinungsäußerungen

Grundlagen, Methoden
und praktische Umsetzung



Springer Vieweg

Sentiment-Analyse deutschsprachiger Meinungsäußerungen

Melanie Siegel · Melpomeni Alexa

Sentiment-Analyse deutschsprachiger Meinungsäußerungen

Grundlagen, Methoden und praktische
Umsetzung

Melanie Siegel
Forschungszentrum Angewandte Informatik
Hochschule Darmstadt
Dieburg, Deutschland

Melpomeni Alexa
Institut für Kommunikation und Medien
Hochschule Darmstadt
Dieburg, Deutschland

ISBN 978-3-658-29698-8 ISBN 978-3-658-29699-5 (eBook)
<https://doi.org/10.1007/978-3-658-29699-5>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2020

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung: Sybille Thelen

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Inhaltsverzeichnis

- 1 Einleitung** 1
- 2 Sentiment Retrieval – Meinungsäußerungen identifizieren** 5
 - 2.1 Meinungsäußerungen 5
 - 2.1.1 Meinungsäußerungen vs. Sachinformation 6
 - 2.1.2 Arten von Meinungsäußerungen 8
 - 2.2 Methoden für die Suche nach Meinungsäußerungen 10
 - 2.3 Zusammenfassung 13
 - 2.4 Übungen 14
 - 2.5 Weiterführende Literatur 15
- 3 Polarität: Dokumente klassifizieren** 17
 - 3.1 Die Aufgabe 17
 - 3.2 Vorbereitung der Daten 18
 - 3.3 Wortlistenabgleich 20
 - 3.4 Qualitätssicherung und systematische Evaluation 20
 - 3.5 Klassifikation und Regression 23
 - 3.6 Dokumentklassifikation mit maschinellem Lernen 25
 - 3.6.1 Supervised Learning – Probabilistisches Sprachmodell 25
 - 3.6.2 Supervised Learning mit Features 27
 - 3.6.3 Deep Learning 33
 - 3.7 Zusammenfassung 33
 - 3.8 Übungen 33
 - 3.9 Weiterführende Literatur 35
- 4 Wörter in der Sentiment-Analyse** 37
 - 4.1 Normalisierung der Texte 37
 - 4.2 Einbindung eines existierenden Sentiment-Wörterbuchs 38
 - 4.3 Gewinnung von Sentiment-Wörtern mithilfe von WordNet 40
 - 4.4 Gewinnung von Sentiment-Wörtern aus annotierten Korpora 42
 - 4.5 Gewinnung von Wörtern aus nicht annotierten Korpora 44

4.6	Zusammenfassung	45
4.7	Übungen	46
4.8	Weiterführende Literatur	48
5	Sentiment-Analyse auf Satzebene	49
5.1	Satz-Tokenisierung	50
5.2	Identifikation von Sätzen mit Meinungsäußerungen	51
5.3	Satzanalyse	52
5.4	Zusammenfassung	55
5.5	Übungen	55
5.6	Weiterführende Literatur	57
6	Was bewertet wird: Aspekte identifizieren	59
6.1	Taxonomie der Aspekte	60
6.2	Phrasen-Lexikon der Aspekte	62
6.3	Aspekte im Text identifizieren und interpretieren	64
6.4	Aspektidentifizierung ohne Beschränkung auf eine Domäne	65
6.5	Sentiment-Klassifikation des Aspekts	66
6.6	Zusammenfassung	68
6.7	Übungen	68
6.8	Weiterführende Literatur	69
7	Ironie	71
7.1	Übungen	73
7.2	Weiterführende Literatur	74
8	Analyse politischer Trends	75
8.1	Aufstellung der Datenbasis	76
8.1.1	Tweets mit Meinungen zu Politikerinnen und Politikern	76
8.1.2	ZDF-Politbarometer	77
8.2	Sentiment-Analyse für Tweets	78
8.3	Zusammenfassung	79
8.4	Übungen	79
8.5	Weiterführende Literatur	80
9	Opinion Spam	81
9.1	Gefälschte Bewertungen	82
9.2	Annotierte Korpora für Opinion Spam	83
9.3	Klassifikation von Bewertungen	85
9.3.1	Klassifikation mit Meta-Daten	85
9.3.2	Klassifikation mit linguistischer Information	86
9.4	Beobachtungen über Opinion Spam im deutschsprachigen Amazon-Portal	87
9.5	Maschinelles Lernen für die automatische Klassifikation	89

9.6	Zusammenfassung	91
9.7	Übungen	91
9.8	Weiterführende Literatur	92
10	Erkennung und Klassifikation von Aggression in Meinungsäußerungen ...	93
10.1	Daten, Daten, Daten	95
10.2	Methoden zur automatischen Klassifikation	97
10.3	Zusammenfassung	100
10.4	Übungen	101
10.5	Weiterführende Literatur	101
11	Sentiment-Analyse im Unternehmenskontext und Softwarelösungen im Markt	103
11.1	Markt für kostenpflichtige Sentiment-Analyse-Tools und -Services ...	103
11.1.1	Technische Bereitstellung der Lösung	105
11.1.2	Art der Sentiment-Analyse	107
11.1.3	Sprachenabdeckung	108
11.1.4	Leistungsumfang und Funktionen	108
11.2	Tools für deutschsprachige Texte	109
11.2.1	Amazon Comprehend	109
11.2.2	Cogito Intelligence Plattform von Expert System	110
11.2.3	InMap, Insius	111
11.2.4	Monkey Learn	111
11.2.5	OpenText Magellan Text Mining	112
11.2.6	ParallelDots	112
11.2.7	SAS® Visual Text Analytics	114
11.2.8	Sentiment Intelligence in SAP Hana	114
11.2.9	Sentiment Lab von m-result	115
11.2.10	Übermetrics	116
11.3	Anwendung der Sentiment-Analyse in der Praxis	117
11.4	Zusammenfassung	121
11.5	Übungen	121
11.6	Weiterführende Literatur	122
Literatur.	123
Stichwortverzeichnis.	129

Der Zugang zu Information ist durch das Internet erheblich verändert und erleichtert worden. Gleichzeitig gibt es mit dem Web 2.0 die Möglichkeit für alle Internet-Nutzer, selbst Inhalte beizusteuern, indem sie in Foren schreiben, oder Twitter, Xing, LinkedIn, Facebook oder andere soziale Medien nutzen und auf veröffentlichte Posts z. B. durch Kommentare reagieren. Diese Fülle an Informationen und Meinungen ist ein wertvoller und in der Regel sehr großer Datenschatz, den man nur mit automatischen Verfahren sinnvoll nutzen kann.

Die Kundenmeinung ist für Dienstleistungs- und Verkaufsunternehmen zu einem wichtigen Geschäftsfaktor geworden. Unternehmen können direkt von ihren Kunden erfahren, was diese über ihr Produkt oder ihre Dienstleistung denken. Durch das unmittelbare Kundenfeedback haben sie die Möglichkeit, sich und ihre Produkte möglichst schnell an die Bedürfnisse der Kunden anzupassen. Bestenfalls wollen sie die (positiven) geäußerten Meinungen als Marketinginstrument nutzen. Für Unternehmen kann es existentiell sein, vor einem aufkommenden “Shitstorm” der Kunden gewarnt zu werden. Gleichzeitig müssen schnell neue Trends erkannt werden. Unternehmen möchten ihre Wettbewerber gezielt beobachten, um sich mit ihren Produkten und Dienstleistungen gegenüber dem Wettbewerb abzuheben.

Das hat auch Vorteile für die Kunden. Die Verbraucher informieren sich gegenseitig, und fast niemand bucht heutzutage eine Urlaubsreise, ohne die Bewertungen anderer Gäste gesehen zu haben. Die Verbraucher wollen die Meinungen anderer Verbraucher vor dem Kauf eines Produkts, der Buchung eines Hotels, der Wahl eines Politikers oder einem Kinobesuch lesen.

Verbraucher und Bürger wollen aber auch gehört werden, wenn sie ihre Erfahrungen und Ansichten austauschen. Die Bedeutung des Themas für Unternehmen wird durch die Tatsache unterstrichen, dass Unternehmen komplexe manuelle und automatische Prozesse nutzen wie Presseberichte, Umfragen, Verfolgen von Diskussionen in Newsgroups und Foren sowie die Auswertung von Kunden-E-Mails, um die Meinungen der Kunden zu erfahren.

Die automatische Analyse von Meinungsäußerungen gehört in den Bereich Informationsextraktion. Aus Texten – den Bewertungen der Kunden – wird Information darüber extrahiert, wie die Kunden Produkte oder Aspekte der Produkte bewerten.

Das Ziel dieses Buchs ist eine systematische Einführung in Methoden der automatischen Analyse von Meinungsäußerungen. Die beschriebenen Methoden werden in Programmierübungen umgesetzt, sodass sie vollständig erfasst werden können.

Wir legen dabei den Fokus auf deutschsprachige Daten. Die meisten bisherigen Forschungen beziehen sich auf englischsprachige Daten, wie (Liu 2015). Wir stellen dar, welche linguistischen Ressourcen für die deutsche Sprache zur Verfügung stehen und wie sie genutzt werden können. Dabei beschränken wir uns soweit es geht auf Ressourcen mit Open-Source-Lizenzen. Weiterhin stellen wir dar, welche Methoden zur automatischen Analyse für die deutsche Sprache anwendbar sind.

Im Kap. 2 versuchen wir eine Definition von Meinungsäußerungen und beschäftigen uns mit Methoden zur Identifikation von Meinungsäußerungen. Anschließend geht es um die Klassifikation von Dokumenten als positive, negative oder neutrale Meinungsäußerung im Kap. 3. Eine wichtige Grundlage für die Sentiment-Analyse sind Wörterbücher mit Sentiment-Wörtern und deren Klassifikation. Um die Gewinnung von Wörtern geht es im Kap. 4. Im Kap. 5 analysieren wir Strukturen unterhalb der Dokumentenebene, Sätze. Hier kommt auch die Interpretation von Negationen ins Spiel. Noch eine Ebene tiefer gehen wir im Kap. 6, in dem wir untersuchen, welcher Aspekt eines Produkts (oder einer Dienstleistung o. ä.) wie bewertet wird, denn in einem Satz können verschiedene Aspekte unterschiedlich bewertet werden. Hier kommen neue semantische Verfahren der Satzanalyse hinzu. Mit diesem Kapitel sind die grundlegenden Methoden dargestellt. Kap. 7 behandelt ironische Meinungsäußerungen, die für die automatische Analyse – aber auch schon für Menschen – besonders schwierig sind. Die Kap. 8, 9 und 10 zeigen Beispiele für Anwendungen der Sentiment-Analyse. Dazu gehört die Analyse von Trends, die wir anhand von Meinungen zu Politiker_innen und einem Vergleich mit dem “ZDF-Politbarometer” darstellen. Gefälschte Meinungsäußerungen finden sich mehr und mehr in den Online-Foren, sodass wir uns in Kap. 9 mit der automatischen Erkennung von Fälschungen beschäftigen. Schließlich nutzen wir die erlernten Methoden, um aggressive Meinungsäußerungen wie Beleidigungen, Bedrohungen oder Diskriminierungen automatisch zu klassifizieren. Im letzten Kap. 11 wagen wir einen Blick in den Markt und stellen Software-Lösungen dar, die momentan kommerziell erhältlich sind.

Das Buch ist als Lehrbuch für die Hochschullehre konzipiert. Nach dem theoretischen (aber praktisch orientierten) Teil gibt es Übungsaufgaben, mit denen die Studierenden zunächst ihr Wissen überprüfen können. Anschließend gibt es Programmierbeispiele und -aufgaben, die in der Programmiersprache Python implementiert werden. Das Ziel ist, ein komplexes Analysesystem zu programmieren und dann auch in weiteren Anwendungsbereichen einzusetzen. Die Programmierungen sind dabei als Basis-System gedacht, das durch Studierende in nachfolgenden Projekten optimiert werden kann und sollte. Der letzte Teil

der Übungsaufgaben ist jeweils eine Reflexion in Gruppenarbeit. Für eine weitergehende Vertiefung gibt es jeweils anschließend einen kurzen Abschnitt zu weitergehender Literatur.

Text-Ressourcen und Programmierungen, die im Buch vorgestellt werden, stehen unter <https://github.com/hdaSprachtechnologie/Sentiment-Analysis> zur Verfügung. Ein Foliensatz für den Einsatz in der Hochschullehre wird dort ebenfalls zur Verfügung gestellt.

Sentiment Retrieval – Meinungsäußerungen identifizieren

2

Im ersten Schritt der automatischen Analyse sollen in unstrukturierten Textdaten Meinungsäußerungen automatisch identifiziert werden. Nicht alle Textteile enthalten auch Meinungsäußerungen, bei einer Buch- oder Film-Rezension zum Beispiel wird häufig ein großer Teil des Textes das Buch oder den Film zusammenfassen, ohne ihn auch gleich zu bewerten, in Twitter gibt es viele Tweets, die nicht bewerten und in Nachrichtentexten sind nur wenige Bewertungen. Für die Erkennung und Auswertung von Meinungsäußerungen werden zunächst Textteile oder Sätze gesucht, die subjektive Äußerungen enthalten. Werden solche identifiziert, werden sie im zweiten Schritt zusammen mit den Angaben zur Quelle, zum Datum und falls veröffentlicht zum Autor in eine Struktur gebracht und gespeichert, um anschließend nach Sentiments, d. h. nach Stimmungen, Emotionen und Haltungen, analysiert zu werden.

Beispiele und Testdaten in diesem Buch stammen aus zwei öffentlich verfügbaren Korpora mit deutschen Meinungsäußerungen. Das “Amazon Customer Reviews Dataset” ist eine Sammlung von Bewertungen im Amazon-Portal von 1995 bis 2015. Die deutschen Daten lassen sich unter https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_DE_v1_00.tsv.gz herunterladen. Die Datensammlung der “Germeval Task 2017” (Wojatzki et al. 2017b) enthalten 22.000 Meldungen aus verschiedenen Social-Media-Kanälen zum Thema “Deutsche Bahn”. Sie sind hier verfügbar: <https://sites.google.com/view/germeval2017-absa/data>.

2.1 Meinungsäußerungen

Das Ziel der Sentiment-Analyse ist die automatische Analyse von Meinungstexten, um Stimmungen und Haltungen von Nutzern, Kunden, Bürgern, Fans usw. über Produkte, Personen, Organisationen, Ereignisse usw. zu erkennen. Meinungsäußerungen sind subjektive, oft emotionale Aussagen, sie beinhalten oft Sentiments und können unterschiedlich inter-

pretiert werden. Dabei drückt das Sentiment eine persönliche Bewertung oder Haltung aus, die als positiv, negativ oder neutral klassifiziert werden kann. Diese Klassifikation nennen wir “Polarität”.

2.1.1 Meinungsäußerungen vs. Sachinformation

Um die wesentlichen Eigenschaften der Meinungsäußerungen besser zu verstehen, ist es hilfreich, diese im Vergleich zu den Texten zu betrachten, die Sachinformation beinhalten: Bei Sachinformation handelt es sich um objektive Aussagen über eine Entität, z. B. eine Person, ein Produkt, eine Dienstleistung, ein Ereignis oder einen Prozess. Diese Äußerungen sind belegbar und beinhalten keine Sentiments. Im Kontext von Texten im Social Web ist eine Meinungsäußerung eine Aussage, die die persönliche Meinung, Haltung oder die Emotion des Nutzers kundgibt. Sachinformation im Social-Media-Kontext ist dagegen ein objektiver Nutzer-Bericht über ein bestimmtes Thema. (Benamara et al. 2017) zählen Meinungsäußerungen, Sentiments und Haltungen in Computerlinguistik zu den Phänomenen der wertenden Sprache (englisch “Evaluative Language”), die im Allgemeinen als subjektive Aussagen von Meinungsträgern über ein Meinungsobjekt oder -zieldefiniert wird (S. 209). Für (Liu 2017) dient Meinung als breiter Begriff, der sowohl ein Sentiment, eine Bewertung, eine Beurteilung oder Einschätzung oder Haltung abdeckt als auch die damit verbundene Information über den Meinungsgegenstand (“Opinion Target”) sowie die Person, die die Meinung hat. Der Begriff Sentiment hingegen bedeutet lediglich die zugrundeliegende oder ggf. latente positive oder negative Emotion, die eine Meinung impliziert (Liu 2017, S. 12).

Schauen wir folgendes Beispiel einer Hotel-Bewertung an, die auf Tripadvisor.com veröffentlicht wurde:

- 1. Das Hotel ist sehr gut mit dem Flughafenbus X9 in ca. 10–15 min erreichbar.*
- 2. Die U-Bahnstation befindet sich auch in unmittelbarer Nähe.*
- 3. Angestellte sind sehr nett und hilfsbereit.*

Satz 1 enthält eine Mischung aus Sachinformation, bezogen auf den Flughafenbus und die Erreichbarkeit und positiver Meinungsäußerung hinsichtlich der Erreichbarkeit. Seine Polarität wird durch “sehr gut” ausgedrückt. Satz 2 verfügt lediglich über Sachinformation und hat eine neutrale Polarität. Der dritte Satz hingegen enthält ausschließlich eine subjektive Beurteilung hinsichtlich des Hotelpersonals und hat durch “sehr nett und hilfsbereit” eine positive Polarität. Schon bei den drei Sätzen dieses Bewertungsbeispiels ist gut erkennbar, dass der Anteil der Stimmungs- bzw. Emotionswörter an der Gesamtzahl der Wörter als ein einfacher Indikator für die mögliche Subjektivität eines Textes dienen kann. Eine komplexere Aufgabe hängt mit dem Vorkommen einer Mischung von Meinungsäußerungen und sachlicher Informationen zusammen, wie am Beispiel der Hotelbewertung. Unabhängig davon, ob im Social-Web, in Online-Medien und -Portalen oder in klassischen Medien

wie Zeitungen veröffentlicht, können Texte oft Fakten zusammen mit Meinungsäußerungen aufweisen. Solche Texte können unterschiedlich komplex sein:

- Längere Texte, wie Blogs zu einem bestimmten Thema oder politische Artikel
- Relativ kompakte Texte, wie Bewertungen zu Produkten oder Hotels
- Kurze Texte, wie Kommentare zu einem Facebook-Beitrag oder Tweets

(Petz et al. 2014) haben verschiedene Texttypen von nutzergenerierten Inhalten analysiert und u. a. die Anteile von subjektiven, objektiven und Mischformen je nach Social-Media-Kanal bestimmt (siehe Tab. 2.1). Sie zeigen, dass Twitter-Texte den höchsten Anteil an subjektiven Postings enthalten, während Diskussionsforen ungefähr 50 % subjektive Inhalte und etwas über 14 % Mischformen (subjektive und objektive Inhalte) aufweisen. Produktbewertungen dagegen weisen in knapp 25,5 % der Fälle eine Mischung von subjektiven Äußerungen und Sachinformation auf.

Die Mischung von Meinungsäußerungen und Fakten innerhalb einer einzelnen Textstelle stellt eine Herausforderung für Text- und Web-Mining-Systeme dar, denn je nach Zielsetzung müssen die relevanten Teile gezielt identifiziert und extrahiert werden: Eine Anwendung zur Informationsextraktion sucht nach Sachinformationen im Gegensatz zu einer Anwendung, die eine Sentiment-Analyse durchführt, die auf die Erkennung von Emotions- und Meinungsäußerungen fokussiert ist. Subjectivity Detection (Erkennung von Subjektivität) ist daher eine wesentliche Teilaufgabe der Sentiment-Analyse und in der Praxis keineswegs einfach. Die Genauigkeit bei der Unterscheidung zwischen Fakten und Meinungsäußerungen mithilfe der Erkennung, ob ein Text subjektiv durch die Äußerung von persönlichen Meinungen und Sentiments oder objektiv durch die Beschreibung von Sachinformationen ist, trägt direkt zu der Qualität der Analyse bei. Durch diese Teilaufgabe wird vermieden, dass objektive Texte für die automatische Erkennung von Sentiments fälschlicherweise einbezogen werden, und damit Erkenntnisse hinsichtlich einer neutralen, positiven oder negativen Polarität verfälschen.

Tab. 2.1 Auswertung der Anteile von subjektiven und objektiven Text-Postings (Petz et al. 2014, S. 903)

Social media channel	Subjective (%)	Objective (%)	Subjective and objective (%)
Microblog (Twitter)	82,9	12,8	4,3
Product review	71,7	2,9	25,4
Blog	69,3	19,6	11,1
Social network (Facebook)	67,3	26,1	6,6
Discussion forum	50,2	35,5	14,3

2.1.2 Arten von Meinungsäußerungen

Eine Meinungsäußerung kann eine reguläre Meinung ausdrücken, z. B.:

Ich finde das neue Rafik Shami Buch mega!

Sie kann auch zwei oder mehrere Gegenstände miteinander ins Verhältnis setzten, um eine vergleichende Meinung auszudrücken, z. B. bei dieser Aussage:

Diese Nike-Sneakers sind cooler als meine alten Adidas.

Insbesondere in der Marketing-Praxis ist es üblich, dass Produkte, Services und Brands miteinander verglichen und bewertet werden, sodass diese Art von subjektiven Äußerungen oft auf Online-Portalen vorkommt. Die Art der Meinungsäußerung wirkt sich auf die Komplexität und die Qualität der Subjektivitäts- sowie der Polaritätsanalyse aus. Basierend auf (Jindal und Liu 2006) sowie (Liu 2007) unterscheiden wir zwischen regulären und vergleichenden Meinungsäußerungen und zwischen direkten und indirekten regulären Meinungsäußerungen.

2.1.2.1 Direkte Meinungen

Direkte Meinungen sind subjektive Äußerungen, die Sentiment-Aussagen zu einem bestimmten Objekt (oder ggf. zu mehreren Objekten) enthalten. Zum Beispiel die Äußerung einer Studierenden-Bewertung für einen Studiengang auf dem Portal studycheck.de:

Überall sind mega nette und meistens auch lustige Dozenten!

Diese Art der Meinungsäußerung gilt als die Art, die am einfachsten automatisch zu erkennen ist. In der Regel arbeitet man hier mit Wörtern als Indikatoren für die Zuordnung der positiven oder negativen Polarität.

2.1.2.2 Indirekte Meinungsäußerungen

Indirekte Meinungsäußerungen weisen keine explizite und eindeutige Sentiment-Aussage auf, sondern erst durch die Interpretation und den Zusammenhang der Aussage wird eine Meinungsäußerung deutlich. Ein Beispiel dafür ist die folgende Aussage:

Die Teilnehmer fühlten sich nach dem Seminar deutlich besser qualifiziert, sowie in ihrer Entscheidung bestärkt.

Eine automatische Erkennung von indirekten Meinungsäußerungen ist daher komplizierter als die von direkten Meinungsäußerungen.

2.1.2.3 Vergleichende Meinungsäußerungen

Bei einer vergleichenden Meinungsäußerung werden (mindestens) zwei Objekte miteinander verglichen. Die Sentiment-Aussagen können eine Priorisierung oder Reihenfolge beschreiben, indem sie Ähnlichkeiten oder Unterschiede zwischen den besprochenen Objekten aufzeigen. Der Vergleich kann eine subjektive Meinungsäußerung sein, es kann sich aber auch um Sachinformation und damit um eine objektive Äußerung handeln, was für die Aufgabe der Subjektivitätsanalyse eine Herausforderung darstellen kann. Als Beispiel dient hier folgender Satz aus einer Amazon-Rezension:

Auch die Bedienung über die Lynette funktioniert gut, wenn auch nicht unbedingt besser als bei Apple mit der Krone, aber dennoch gut und macht vor allem Spaß.

Die Erkennung der subjektiven Aussagen von vergleichenden Meinungsäußerungen ist komplizierter als die von direkten einfachen Meinungsäußerungen.

2.1.2.4 Äußerung von Emotionen ohne Meinung

Auch wenn die Äußerung von Emotionen ein deutlicher Indikator für eine Meinung ist, handelt es sich jedoch nicht bei jeder Emotionsäußerung unbedingt um eine Meinungsäußerung. Schauen wir uns folgende Anmerkung an:

Einige von uns waren glücklich, dass sie die Kletterei geschafft haben.

Hier wird eine Beobachtung geäußert und dabei das Emotionswort “glücklich” verwendet. Jedoch handelt es sich hier nicht um eine Meinungsäußerung. Zur automatischen Erkennung und Kategorisierung von subjektiven Meinungsäußerungen werden in der Regel Wörter und Ausdrücke genutzt, die auf Emotionen wie Freude, Glück, Ärger, Wut usw. hinweisen. Die Aufgabe der präzisen Meinungserkennung wird dadurch erschwert, dass Emotionswörter nicht nur in subjektiven Aussagen verwendet werden.

2.1.2.5 Subjektive Meinungsäußerung vs. verwertbare Meinungsäußerung

Eine weitere Schwierigkeit für die Meinungserkennung besteht darin, dass nicht jede subjektive Meinungsäußerung auch eine verwertbare Meinungsäußerung ist. Diese Beispielaussagen von Bewertungen auf Tripadvisor zeigen die Problematik:

1. Nach einer kurzen Sicherheits- und Fahrradtechnik-Unterweisung rasten wir den Berg hinunter.
2. Durch meine sportliche Figur war es unmöglich für mich einen schönen vorgefertigten Anzug zu finden
3. Heyho, vorab: Ich habe weder Ahnung von Mode noch von Stoffen. Jedoch habe ich mich hier überragend informiert gefühlt und kann mich den Vorrednern anschließen. Termine wurden eingehalten und ich bin vom Anzug total überzeugt. Ich habe mich stets wohl gefühlt und würde mir direkt noch einen anfertigen lassen, wenn ich nicht als Backpacker unterwegs wäre. Alles in allem bin ich super zufrieden. Ich hoffe, dass der Anzug mir auch in Deutschland noch passt, da das Essen hier so verdammt lecker ist – hahah

Die Äußerungen in 1 und 2 können als subjektive Meinungsäußerungen gelesen werden, sind aber kaum nützlich für die Meinungsanalyse. Die positive Bewertung des Essens in Beispiel 3 hat nichts mit dem eigentlichen Bewertungsobjekt (Anzugschneider) zu tun. Irrelevante Textteile sollten nicht berücksichtigt werden, damit die Ergebnisse der Sentiment-Analyse nicht verfälscht werden.

Die Betrachtung der verschiedenen Arten von Meinungsäußerungen oben macht die Notwendigkeit deutlich, die konkrete Art einer Meinungsäußerung zu berücksichtigen, um möglichst präzise Analyseergebnisse zu erzielen. Je nach Art des Textes gibt es unterschiedliche Herausforderungen für die Sentiment-Analyse. Zum Beispiel sind Meinungsäußerungen durch vergleichende Meinungen oft länger und komplexer als direkte Meinungsäußerungen. Die Analyse muss mindestens zwei Objekte und die jeweilige Polarität für jedes Objekt richtig erkennen. Dies ist eine Aufgabe, die einer tiefgehenden Textanalyse für die jeweilige Sprache bedarf, in der die Meinung geäußert wurde.

2.2 Methoden für die Suche nach Meinungsäußerungen

Nehmen wir das folgende Beispiel für die Bewertung eines Films aus dem Amazon-Korpus (bei dem die Absätze nummeriert wurden):

Wie krass ist das denn

1. Brillantes SciFi-Thriller-Kammerspiel um einen Cyber Guru, der an künstlicher Intelligenz bastelt, und einen seiner Unterlinge, der als Versuchskaninchen, bzw. Testperson herhalten muss. Außerdem gibt's noch ein paar künstliche Frauen, mit teils atemberaubenden Talenten und Charaktereigenschaften.
2. Was mich völlig faszinierte, ließ meine Frau im Kino einschlafen. Familiendurchschnitt in der Wertung also nur Durchschnitt. Von mir volle 5 Sterne.

3. Die deutsche Tonfassung ist überzeugend flapsig. Aus rein sprachlichem Interesse muss ich nun noch die Originalversion sehen. Oder kann mir jemand sagen, was Oscar Isaacs Figur im Original sagt, als es auf Deutsch heißt: wie krass ist denn das?
 4. Hervorragend und überzeugend alle drei Hauptdarsteller. Auch die Neben-Roboter sind nicht schlecht. Die Haupt-’Frau’ sieht aus wie die junge Natalie Portman, und überzeugt komplett.
- 2015-05-02

Schon die Überschrift ist eine Meinungsäußerung, die jedoch positiv oder negativ sein kann, denn der Ausdruck “krass” ist hier nicht festgelegt und zudem ein Zitat aus dem Film. Im ersten Absatz wird der Film nur beschrieben und nicht bewertet. Im zweiten Absatz stehen Bewertungen des Autors und seiner Frau, die gegensätzlich sind. Hier muss man auch noch den Kontext beachten. Wenn ein Film zum Einschlafen ist, dann ist das klar negativ. Wenn aber z. B. ein Beruhigungstee dazu führt, dass man einschläft, so ist das eine positive Eigenschaft davon. Anders ist das bei der Faszination: Wenn jemand “völlig fasziniert” ist, so ist das in jedem Kontext positiv.

Der dritte Absatz beginnt mit einer klar positiven Bewertung der deutschen Tonfassung, also eines speziellen Aspekts dieses Films. Die Frage in diesem Absatz zeigt, dass Fragen gesondert behandelt werden müssen.

Im vierten Absatz werden weitere Aspekte (die Darsteller) bewertet. Im zweiten Satz kann man sehen, dass eine Negation verwendet wird, sodass sich die Polarität von “schlecht” (negativ) umkehrt in “nicht schlecht” (schwach positiv). Man erkennt auch, dass eine Klassifikation nur nach negativ – neutral – positiv nicht ausreichend ist, denn “hervorragend” ist deutlich positiver als “nicht schlecht”. Um den Vergleich der “Haupt-Frau” mit der jungen Natalie Portman als positive Bewertung zu verstehen, muss man natürlich wissen, wer Natalie Portman ist und wie sie in jungen Jahren aussah. Der Reviewer nutzt im letzten Satz eine Verstärkung, er ist nicht nur überzeugt, sondern “komplett” überzeugt.

Der Zeitstempel dieser Bewertung kann dann relevant sein, wenn man sich z. B. Trends ansehen möchte oder Änderungen in den Bewertungen eines Produkts.

Fassen wir zusammen:

1. Meinungsäußerungen können positiv oder negativ sein. Nicht immer ist das ohne Kontext sofort klar.
2. Es gibt weitere Abstufungen, wie stark positiv oder negativ eine Meinungsäußerung ist.
3. Die Meinung einer Person wird geäußert, wobei das nicht immer die Autorin/der Autor des Beitrags ist, sondern auch über Meinungen anderer Personen geschrieben werden kann.
4. Meinungsäußerungen beziehen sich nicht immer auf die gesamte Entität (das Produkt, den Film etc.), sondern auch auf einzelne Aspekte (wie die deutsche Übersetzung oder die Figuren).
5. Ausdrücke der Meinungsäußerung können kontextunabhängig sein, wie “faszinierend” oder kontextabhängig, wie “einschlafen”. In einigen Fällen wird zur Interpretation Welt-

wissen benötigt, wie beim Vergleich der Hauptdarstellerin mit einer anderen Schauspielerin.

6. Negation und Verstärker müssen gesondert behandelt werden.

7. Auch Fragen benötigen eine besondere Behandlung.

Nach (Liu 2012, S. 19) kann eine Meinung als Quintupel $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ beschrieben werden. Dabei steht e_i für die Entität i , also das Produkt oder die Dienstleistung, die bewertet werden. Eine solche Entität kann z. B. ein Buch sein. Der Aspekt j zur Entität i wird mit a_{ij} bezeichnet. Ein Aspekt der Entität ist ein Teil oder eine Eigenschaft, also z. B. bei einem Buch das Cover. Die eigentliche Meinung darüber, das Sentiment, ist s_{ijkl} . Die meinende Person mit dem Index k ist mit h_k bezeichnet. Der Zeitpunkt der Meinungsäußerung l ist t_l . Die Aufgabe der Sentiment-Analyse ist damit, dieses Quintupel möglichst vollständig aufzustellen.

Für einen Ausschnitt aus unserem Text sieht das so aus:

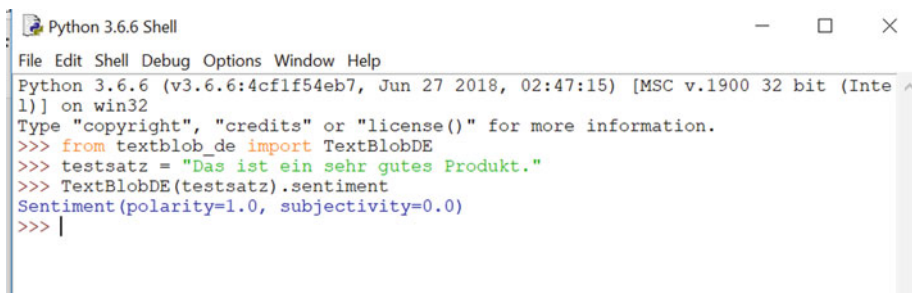
(Film, generell, positiv, Autor, 2015-05-02)	<i>Was mich völlig faszinierte</i>
(Film, generell, negativ, Frau, 2015-05-02)	<i>ließ meine Frau im Kino einschlafen</i>
(Film, generell, neutral, Autor+Frau, 2015-05-02)	<i>Familiendurchschnitt in der Wertung also nur Durchschnitt</i>
(Film, generell, positiv, Autor, 2015-05-02)	<i>Von mir volle 5 Sterne</i>
(Film, übersetzung, positiv, Autor, 2015-05-02)	<i>Die deutsche Tonfassung ist überzeugend flapsig.</i>
(Film, Hauptdarsteller, positiv, Autor, 2015-05-02)	<i>Hervorragend und überzeugend alle drei Hauptdarsteller</i>
(Film, Neben-Roboter, positiv, Autor, 2015-05-02)	<i>Auch die Neben-Roboter sind nicht schlecht.</i>
(Film, Haupt-, 'Frau', positiv, Autor, 2015-05-02)	<i>Die Haupt-, 'Frau' sieht aus wie die junge Natalie Portman, und überzeugt komplett</i>

Dabei sind emotionale oder subjektive Äußerungen nicht immer auch Meinungsäußerungen in unserem Sinne. Hier ist ein Beispiel für eine emotionale Äußerung:¹

*Danke für Cd
Eine Lied von See you again ist schön. Und auch traurig. Finde schon schade das Paul Walker nicht mehr da ist.
2015-04-27*

Die Wörter “traurig” und “schade” deuten auf subjektive emotionale Äußerungen hin, die jedoch keine (direkte) Meinung über das Produkt (hier die CD) oder einen Aspekt davon äußern.

¹Die Beispiele im Text sind wörtliche Zitate, daher sind Rechtschreibfehler und -varianten direkt übernommen worden.



```
Python 3.6.6 Shell
File Edit Shell Debug Options Window Help
Python 3.6.6 (v3.6.6:4cflf54eb7, Jun 27 2018, 02:47:15) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from textblob_de import TextBlobDE
>>> testsatz = "Das ist ein sehr gutes Produkt."
>>> TextBlobDE(testsatz).sentiment
Sentiment(polarity=1.0, subjectivity=0.0)
>>> |
```

Abb. 2.1 Sentiment-Analyse von TextBlob

Man sieht, dass schon die Erkennung von Meinungsäußerungen in großen Dokumentensammlungen kein einfach zu lösendes Problem ist. Sehen wir uns daher zunächst einmal an, mit welchen Methoden die Software-Lösungen TextBlob und NLTK an dieses Problem herangehen.

Das Python-Modul TextBlob DE² enthält eine Reihe von Methoden und Ressourcen, mit denen Texte in deutscher Sprache verarbeitet werden können. Die TextBlob-Sentiment-Analyse nutzt das “German Polarity Lexicon”³ zusammen mit einer kleinen Liste von Negationen, um Wörter in einem Satz nachzuschlagen. Nach der Installation des Moduls findet man die Einträge dieses Lexikons in der Python-Installation unter Lib\site-packages\textblob_de\data.

Das “Natural Language Toolkit NLTK” ist ebenfalls eine Menge von Modulen für die Verarbeitung von Sprache. Auch NLTK hat Sentiment-Module (aktuell nur für die englische Sprache), die Lexika verwenden und die Wörter im Satz zählen, die in diesen Lexika stehen (siehe <https://www.nltk.org/api/nltk.sentiment.html>). Darüber hinaus stellt NLTK weitere Module zur Verfügung, mit denen man auf annotierten Daten auf Grundlage der Wörter, die darin vorkommen, eine Sentiment-Klassifikation trainieren kann.

Die Verwendung der Sentiment-Analyse von TextBlob kann man in der Abb. 2.1 sehen.

2.3 Zusammenfassung

Für das Sentiment-Retrieval soll im ersten Schritt zwischen subjektiven (Meinungsäußerungen) und objektiven Texten (Sachinformationen) unterschieden werden. Wir können die Meinungsäußerungen nach ihrer Subjektivität (objektiv/neutral – subjektiv) und Polarität (sehr positiv, positiv, neutral, negativ, sehr negativ) kategorisieren. Meinungen können wir

²<https://textblob-de.readthedocs.io/en/latest/>

³German Polarity Lexicon: <http://bics.sentimental.li/index.php/downloads/> Authors: Manfred Klenner, Simon Clematide, Martin Wiegand, Ronny Peters Version: 1.1. 2010/08/01.

nach verschiedenen Arten kategorisieren, von regulären, direkten Meinungen, die einfacher erkannt und analysiert werden können über vergleichende Meinungen, die komplexer für die Analyse sind, bis hin zu subjektiven Meinungen, die jedoch für die Analyse irrelevant sind und deswegen aus der Analyse ausgeschlossen werden müssen. Die Qualität der Sentiment-Analyse hängt mit der Unterscheidung der Meinungsäußerung zusammen, denn die Qualität der Ergebnisse hängt direkt davon ab, ob z. B. subjektive Meinungen richtig erkannt werden und ob objektive Aussagen, die nicht relevant für die Sentiment-Analyse sind, auch beim Sentiment-Retrieval nicht berücksichtigt werden.

Meinungsäußerungen können mit Lius Quintupeln beschrieben werden, die aus der Entität, dem Aspekt, der Meinung, dem Meinenden und der Zeit bestehen. Jedoch ist dies nicht immer direkt mit den Wörtern im Satz möglich, wie wir am Beispiel gezeigt haben. Auch gibt es emotionale und subjektive Äußerungen, die keine Meinungsäußerungen sind.

Die direkte Methode, Meinungsäußerungen zu entdecken, ist der Abgleich der Wörter in einem Satz mit Wörtern in einem Lexikon. Diese relativ einfache Methode wird z. B. von TextBlob und NLTK verwendet. Die Qualität der Analyse ist dabei stark abhängig von der Qualität des Lexikons. Auf die Methode, anhand von annotierten Daten automatisch zu lernen, welche Wörter relevant sind, gehen wir zu einem späteren Zeitpunkt in Abschn. 4.4 noch ein.

Schon das Entdecken von Meinungsäußerungen ist ein komplexer Prozess, der auch schon Anwendungen hat. So kann man damit z. B. herausfinden, welche Produkte oder Themen intensiv diskutiert werden (siehe im Kapitel Analyse politischer Trends) oder man kann versuchen, extreme Meinungsäußerungen automatisch zu identifizieren (siehe im Kapitel Erkennung und Klassifikation von Aggression in Meinungsäußerungen).

2.4 Übungen

1. Prüfen Sie Ihr Wissen:

- Welche Teilaufgaben beinhaltet das Sentiment Retrieval und welche Ziele haben sie jeweils?
- Wie erklären Sie, dass die Unterscheidung zwischen Sachinformationen und Meinungsäußerungen relevant für die Genauigkeit und Qualität der Sentiment-Analyse ist?

2. Setzen Sie Ihr neues Wissen ein:

- a) Recherchieren Sie auf amazon.de jeweils zwei Beispiele für die verschiedenen Arten von Meinungsäußerungen unter 2.1.2 (direkt, indirekt, vergleichende Meinungsäußerung, Emotionsäußerung ohne Meinung), deren Erkennung eine Herausforderung für die Sentiment-Analyse ist. Erklären Sie, worin die Herausforderung jeweils besteht.
- b) Suchen Sie nach sprachlichen Merkmalen in Online-Reviews oder Tweets, die Ihrer Meinung nach für die Bestimmung der Subjektivität und Polarität während des Sentiment-Retrievals berücksichtigt werden sollten.

- c) Erstellen Sie eine Sammlung von 10 Meinungsäußerungen und strukturieren Sie sie in einer Tabelle nach den Quintupel-Werten von (Liu 2012, S. 19). Speichern Sie diese Daten zur Nutzung für weitere Übungsaufgaben.
 - d) Installieren Sie NLTK, TextBlob und TextBlob DE.
 - e) Testen Sie die Sentiment-Analyse von TextBlob DE mit den Sätzen, die Sie gesammelt haben und mit Sätzen, die Negationen wie “kein” enthalten. Dokumentieren Sie Ihr Ergebnis.
 - f) Testen Sie die Sentiment-Analyse von NLTK mit englischen Sätzen.
 - g) Schreiben Sie eine Funktion, die den Nutzer um einen Eingabesatz bittet und dann das Ergebnis der Sentiment-Analyse mit TextBlob ausgibt.
 - h) Schreiben Sie eine Funktion, die auf eine Datei mit Sätzen die Sentiment-Analyse mit TextBlob anwendet und ausgibt, ob die Sätze Meinungsäußerungen enthalten oder nicht.
3. Reflexion in Gruppenarbeit:
- Diskutieren Sie in Ihrer Übungsgruppe darüber, wie aufwändig und komplex es für einen Menschen selbst ist, Meinungsäußerungen richtig zu erkennen und eindeutig nach ihrer Polarität zu kategorisieren. Fassen Sie anschließend Ihre Diskussionspunkte zusammen und stellen sie anhand von Beispielen anderen Gruppen vor.

2.5 Weiterführende Literatur

Die englischsprachige Literatur zur Subjektivitätserkennung bzw. -analyse (englischsprachig Subjectivity Detection) ist ein aktives Forschungsfeld. Grundlagenliteratur dazu bilden die Arbeiten von (Yu und Hatzivassiloglou 2003; Pang und Lee 2004; Wiebe et al. 2004) sowie (Pang und Lee 2008). Eine sehr gute, aktuelle und umfassende Einführung in die Sentiment-Analyse gibt (Liu 2015). In (Sidarenka 2019), Kap. 1, findet sich ein umfassender Überblick der Historie der Sentiment-Analyse.

Methoden und die damit einhergehenden Herausforderungen werden ausführlich in (Charurvedi et al. 2018) behandelt. Verschiedene Methoden der Sentiment-Analyse findet man in den Tagungsbänden von SemEval (Nakov et al. 2016) für die englische Sprache und GermEval (Wojatzki et al. 2017b) für die deutsche Sprache. Einen Überblick über die Analyse der deutschen Sprache gibt (Wolfgruber 2015). (Atalla et al. 2011) haben eine Studie durchgeführt, in der sie verschiedene Ansätze zur Subjektivitätserkennung implementiert und miteinander verglichen haben.

Das Buch zur Einführung in NLTK ist (Bird et al. 2009). TextBlob wird in (Loria et al. 2014) eingeführt.

Bei der Dokumentklassifikation geht es darum, für ein Dokument zu entscheiden, ob es insgesamt eine positive, negative oder neutrale Meinungsäußerung ist. Diese Klassifikation nennt man “Polarität”. Bei der GermEval Shared Task 2017 (Wojatzki et al. 2017b) war die Dokumentklassifikation die Aufgabe, an der sich alle teilnehmenden Gruppen beteiligten.

3.1 Die Aufgabe

Der Ausdruck “Dokument” lässt an längere Textdokumente denken. In der Sentiment-Analyse geht es jedoch meist um kurze Texte wie Bewertungen in Konsumenten-Portalen, in Twitter oder Facebook. Nachdem ein solches Dokument als Meinungsäußerung identifiziert worden ist, muss die Polarität bestimmt werden. Nehmen wir als Beispiel diese Twitter-Meldung aus dem Korpus der GermEval Shared Task 2017:

Re: DB Bahn Pünktlich zu meiner Reise :) perfekt.

Die Aufgabe der Dokumentklassifikation ist es herauszufinden, dass diese Äußerung zur deutschen Bahn (Entität) positiv (Polarität) ist. Nicht relevant sind zu diesem Zeitpunkt der Aspekt (die Pünktlichkeit), die meinende Person (der Autor der Bewertung) und der Zeitpunkt der Bewertung. Die Entität ist durch Metadaten oder den Kontext bekannt. In diesem Beispiel beziehen sich alle Texte auf die deutsche Bahn, bzw. die anderen Texte sind als “nicht relevant” markiert.¹

¹Die Unterscheidung von relevanten (zur deutschen Bahn) und nicht-relevanten Texten war eine Aufgabe in GermEval 2017, die wir hier aber außer Acht lassen.

Die positive Polarität in diesem Beispiel kann man an den Wörtern “pünktlich” und “perfekt” sowie am Emoji “:)” erkennen. Die Relevanz einzelner Wörter lässt sich auch an folgendem Beispiel mit negativer Polarität feststellen:

Deutsche Bahn – Eine Horrorgeschichte Darf jetzt erstmal zum nächsten Bahnhof laufen, weil der Zug auf der Strecke stehengeblieben ist

Hier sind es die Wörter “Horrorgeschichte” und “stehengeblieben”, die für die Bestimmung der Polarität relevant sind. Aber Achtung: In einem anderen Kontext können diese Wörter eine ganz andere Polarität haben. Im Kontext einer Buchrezension bezeichnet “Horrorgeschichte” vielleicht eine Entität und ist gar keine Bewertung. Im Zusammenhang mit der Bewertung von Bremsen könnte “stehengeblieben” positiv sein. Eine Anpassung der Wortlisten an das Themengebiet – wir nennen Themengebiete im Kontext der Sentiment-Analyse “Domänen” – ist daher immer notwendig.

Um Wörter aus dem Dokument mit Wortlisten abgleichen zu können, muss aber zunächst erkannt werden, was eigentlich die Wörter im Text sind – die sogenannte “Tokenisierung”. Das ist auch für Menschen nicht immer ganz einfach, aber für Computerprogramme eine nicht triviale Aufgabe.

3.2 Vorbereitung der Daten

Zunächst müssen die Textdaten normalisiert werden, damit die Verarbeitung einfacher wird und bei der Ausführung von Programmcode keine Fehler entstehen. So kann man entscheiden, alle Twitter-Nutzernamen (“@user”) und alle Hyperlinks durch eine standardisierte Zeichenkette zu ersetzen und für Emojis z. B. das Wort “happysmile” einzuführen. (Hövelmann und Friedrich 2017) berichten, dass durch eine solche Vorverarbeitung die Ergebnisse der Sentiment-Analyse signifikant verbessert werden konnten. (Räbiger et al. 2016) gehen weiter und ersetzen URLs, Datumsangaben, numerische Angaben, Slang und unkorrekte Schreibweisen und normalisieren damit den Text weiter.

Der nächste Schritt ist die Tokenisierung. Bei der Tokenisierung geht es darum, einen Text in Sätze und diese Sätze in sogenannte “Tokens” aufzuteilen. Sehen wir uns diesen Text aus dem Bahnkorpus der GermEval 2017 an:

Morgens um halb sieben, dem vollen Pendler-RE einen ganzen Wagen 1. Klasse anhängen. Der ist natürlich leer. Fehlleistungen der @DB_Bahn.

Wir sprechen dabei bewusst nicht von Wörtern, sondern von Tokens, weil es sich bei Tokens auch um Zahlen (“1.”), Abkürzungen (“DB”) oder Sonderzeichen (“@”) handeln kann.

Eine erste Hypothese für die Tokenisierung ist, dass Leerzeichen zentral für die Tokenisierung sind:

HYPOTHESE 1 Ein Token ist eine Kette von Zeichen, wobei davor und dahinter ein Leerzeichen steht.

Damit können viele Tokens identifiziert werden. Allerdings haben wir bei Satzzeichen das Problem, dass davor kein Leerzeichen steht, wie im Beispiel bei “sieben,”. Dazu kommt, dass Leerzeichen auch Teil eines Tokens sein können, wie bei “1. Klasse”.

Eine zweite Hypothese ist die Übertragbarkeit auf andere Sprachen:

HYPOTHESE 2 Tokenisierung ist sprachunabhängig

Das gilt für einen Teil der Tokenisierung und einen Teil der Sprachen. Aber schon beim Apostroph im Englischen (z. B. “don’t”) wird deutlich, dass für das Deutsche andere Regeln gelten. Noch komplexer wird es bei Sprachen mit anderen Schriftsystemen, wie dem Japanischen:

花子か本を読んだ。

Hier gibt es gar keine Leerzeichen zwischen den Tokens, außerdem ist der Satzendeppunkt ein anderer als im Deutschen oder Englischen.

HYPOTHESE 3 Ein Punkt beendet den Satz

Auch das stimmt nur zum Teil. In unserem Beispiel gibt es drei Satzendeppunkte, aber auch einen Punkt mitten in einem Token: “1. Klasse”. Punkte können auch Teil eines Tokens in Zahlen sein (“100.000”). Es gibt auch Abkürzungen, die mit einem Punkt enden und nach denen der Satz weitergeht: “ca.”. Dabei gibt es auch Abkürzungen mit zwei Punkten, wie “z.B.”. Das gilt übrigens auch für andere Satzzeichen wie Kommas: “1,0%”, “20,5 Mio. EUR”. Tokens können auch sehr komplex sein und verschiedene Zeichen enthalten, wie im Fall von E-Mail-Adressen oder URLs.

TextBlob, das wir schon im vorherigen Kapitel eingeführt haben, hat einen integrierten Tokenizer, der diese Ergebnisse für den Beispieltext bringt:

- SENTENCE TOKENIZER
 - [Sentence (“Morgens um halb sieben, dem vollen Pendler-RE einen ganzen Wagen 1.”)
 - Sentence (“Klasse anhängen.”)
 - Sentence (“Der ist natürlich leer.”)
 - Sentence (“Fehlleistungen der @DB_Bahn.”)]
- WORD TOKENIZER
 - [‘Morgens’, ‘um’, ‘halb’, ‘sieben’, ‘dem’, ‘vollen’, ‘Pendler-RE’, ‘einen’, ‘ganzen’, ‘Wagen’, ‘1’, ‘Klasse’, ‘anhängen’, ‘Der’, ‘ist’, ‘natürlich’, ‘leer’, ‘Fehlleistungen’, ‘der’, ‘DB_Bahn’]

Beim Satz-Tokenizer sehen wir, dass die Ordinalzahl mit Punkt “1.” nicht erkannt wurde. Typischerweise gibt es eine Liste von Abkürzungen, die mit einem Punkt auftreten, in der Ordinalzahlen offensichtlich nicht enthalten sind. Auch der Wort-Tokenizer trennt den Punkt von der Ordinalzahl ab. Ein zusätzliches Problem ist, dass nach einem Abkürzungspunkt kein Satzende folgt, wenn die Abkürzung am Satzende steht. Zusätzlich ist es also notwendig, die auf den Punkt folgenden Wörter anzusehen und z. B. großgeschriebene Artikel wie “Der” oder “Das” als Hinweis für den Satzanfang zu beachten (siehe Heyer et al. 2006).

3.3 Wortlistenabgleich

Die einfachste – aber dennoch sehr effektive – Methode der Dokumentklassifikation ist der Abgleich eines Dokuments mit einer Wortliste, in der positive und negative Wörter enthalten und als solche klassifiziert sind. Wenn im Dokument mehr negative als positive Wörter sind, dann ist das Dokument eine negative Meinungsäußerung und wenn mehr positive als negative Wörter enthalten sind, dann ist das Dokument eine positive Meinungsäußerung. Sehen wir uns noch mal das Beispiel für eine Filmrezension aus Kap. 2 an:

Hervorragend und überzeugend alle drei Hauptdarsteller. Auch die Neben-Roboter sind nicht schlecht. Die Haupt- 'Frau' sieht aus wie die junge Natalie Portman, und überzeugt komplett.

Die positiven Wörter sind “hervorragend”, “überzeugend” und “überzeugt”. Das einzige negative Wort darin ist “schlecht”. Zwei Probleme stellen sich dabei: Das erste Problem ist die Aufstellung von Wortlisten, die zum Themengebiet, der sogenannten “Domäne”, passen. Damit beschäftigen wir uns im Kap. 4. Das zweite Problem sind die Negationen, denn “nicht schlecht” ist ja das Gegenteil von “schlecht”. Diese untersuchen wir im Kap. 5.

3.4 Qualitätssicherung und systematische Evaluation

Nachdem wir mit der Programmierung begonnen haben, müssen wir uns mit der Qualitätssicherung unserer Lösungen beschäftigen. Die Idee dabei ist, dass wir einen Gold-Standard aufbauen, d. h. eine Referenz mit richtig annotierten Sätzen. Mit dem Gold-Standard können wir unsere Ergebnisse stetig vergleichen. Dieser Gold-Standard kann eine einfache Liste von annotierten Sätzen sein, z. B. in dieser Form:

Das ist ein gutes Produkt	Positiv
Das ist ein schlechtes Produkt	Negativ
Das ist ein Produkt	Neutral

Es gibt dabei natürlich die Möglichkeit, sich die Sätze selbst auszudenken. Der Vorteil davon ist, dass man systematisch Phänomene einbauen kann, die man testen möchte, also z. B. Negationen oder implizite und explizite Meinungsäußerungen. Eine andere Möglichkeit ist, einen Gold-Standard aus einem externen Textkorpus zu nehmen. Man könnte z. B. Amazon-Bewertungen und die damit verbundene Sterne-Klassifikation in die Tabelle aufnehmen. Eine weitere Möglichkeit sind die annotierten Trainingsdaten einer Shared Task, also z. B. die Daten der GermEval 2017 Shared Task (Wojatzki et al. 2017b).

Wenn der Gold-Standard aufgestellt ist, werden die Sätze mit dem Programm klassifiziert, und dann wird diese Klassifikation mit der Annotation im Gold-Standard verglichen. Für die Weiterentwicklung des Sentiment-Analyse-Programms benötigt man die falsch klassifizierten Sätze und einen Messwert für die Genauigkeit der Klassifikation. Mit diesem Messwert kann man später schnell feststellen, ob die Weiterentwicklung des Programms zu einer besseren Klassifikation geführt hat.

Der einfachste Messwert ist “Accuracy”. Hier dividiert man die Anzahl der richtig klassifizierten Sätze durch die Anzahl aller klassifizierten Sätze:

$$\text{Accuracy} = \frac{|\text{richtig_klassifizierte_Saetze}|}{|\text{alle_Saetze}|}$$

Der Accuracy-Wert für Klassifikationen hat aber einen großen Nachteil, wenn die Testdaten unausgewogen sind. Wenn nämlich z. B. viel mehr negative als positive Sätze in den Testdaten sind und ein Klassifikator einfach annimmt, dass alle Sätze negativ sind, dann hat dieser Klassifikator einen hohen Accuracy-Wert. Nehmen wir z. B. einen Datensatz mit 10 positiven und 30 negativen Sätzen. Der Klassifikator, der einfach immer “negativ” annimmt, ist dann in 30 von 40 Fällen korrekt. Das wäre ein recht hoher Accuracy-Wert von 75 %.

Wir wollen daher genauer hinsehen und für jeden Klassifikationswert der Sentiment-Analyse (positiv, negativ, neutral) einzeln untersuchen, wie gut die automatische Klassifikation ist. Hier arbeiten wir mit “Precision” und “Recall”. Nehmen wir die Klasse “positiv” als Beispiel: Precision ist der Wert für die Sätze, die als “positiv” klassifiziert wurden und die wirklich positiv sind, im Verhältnis zu allen Sätzen, die als “positiv” klassifiziert wurden und unter denen auch solche sind, die nicht positiv sind.

$$\text{Precision}_{\text{positiv}} = \frac{|\text{positiv_klassifizierte_Saetze, die_wirklich_positiv_sind}|}{|\text{alle_positiv_klassifizierten_Saetze}|}$$

Bei Recall handelt es sich um den Wert für die positiven Sätze, die vom Programm auch gefunden wurden.

$$\text{Recall}_{\text{positiv}} = \frac{|\text{positiv_klassifizierte_Saetze, die_wirklich_positiv_sind}|}{|\text{alle_positiven_Saetze}|}$$

Weder Precision noch Recall reichen allein aus, um die Qualität einer automatischen Klassifikation zu bestimmen. Häufig möchte man aber einen einzelnen Wert haben, um z. B. verschiedene Systeme miteinander zu vergleichen. Für diesen Zweck wurde das F-Maß

erfunden. Die beiden Werte Precision und Recall werden mit dem F-Maß kombiniert, das das harmonische Mittel aus den beiden Werten errechnet. Der Wert für das F-Maß liegt immer zwischen den Werten von Precision und Recall. Wenn aber einer der Werte Null ist, dann ist auch der Wert für das F-Maß Null.² Das F-Maß definiert sich folgendermaßen:

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

Sehen wir uns das mal an einem kleinen Beispiel an. Der Gold-Standard in diesem Beispiel enthält 5 positive, 5 negative und 5 neutrale Sätze. In der zweiten Spalte der Tabelle stehen die Ergebnisse unseres fiktiven Klassifikators.

Beispielsatz	Gold-Standard	Klassifikator
Das X ist gut	Positiv	Positiv
Das X ist toll	Positiv	Positiv
Super ist das Y	Positiv	Positiv
Ach wie schön ist das Y	Positiv	Positiv
X ist ganz und gar nicht schlecht	Positiv	Negativ
Das Y ist schlecht	Negativ	Negativ
Das X ist totaler Mist	Negativ	Negativ
A ist ja überhaupt nicht gut	Negativ	Positiv
Ich hasse B	Negativ	Negativ
X ist sehr schlecht	Negativ	Negativ
Ich habe ein X	Neutral	Neutral
Ein Y gehört zu mir	Neutral	Positiv
Dort ist ein B	Neutral	Negativ
Wo ist das X?	Neutral	Neutral
Ich gebe Dir ein Y	Neutral	Neutral

Unser fiktiver Klassifikator hat elf von fünfzehn Sätzen richtig klassifiziert. Der Accuracy-Wert liegt daher bei 0,73. Der Klassifikator hat sechs Sätze als positiv klassifiziert, von denen vier wirklich positiv sind. Daher ist der Precision-Wert für positiv 0,67. Insgesamt gibt es 5 positive Sätze, sodass der Recall-Wert für positiv bei 0,8 liegt. Das ergibt einen F-Wert für die positiven Sätze von 0,73. Bei den negativen Sätzen sieht es so aus, dass unser Klassifikator ebenfalls sechs Sätze als negativ klassifiziert hat, von denen vier wirklich negativ sind. Daher haben wir für negativ dieselben Werte. Unser Klassifikator hat nur drei Sätze als neutral klassifiziert, die allerdings auch alle neutral sind. Das bedeutet, dass der Precision-Wert für neutral bei 1 liegt, der Recall-Wert bei 0,6 und der F-Wert bei 0,75. Die beste Klassifikation hat unser fiktiver Klassifikator also für die neutralen Sätze erreicht.

²Vielen Dank an unseren Kollegen Reginald Ferber für die Erklärung.

3.5 Klassifikation und Regression

Im Sentiment Retrieval handelt es sich um eine binäre Klassifikation mit den Klassen “enthält Sentiment” und “enthält kein Sentiment”. Bei der Kategorisierung, die wir im letzten Abschnitt untersucht haben, haben wir drei mögliche Klassen: “positiv”, “neutral” und “negativ”. Nun haben wir aber schon gesehen, dass weitere Abstufungen denkbar sind. Schließlich ist “ganz gut” weniger stark als “total toll”.

Produktbewertungen haben oft eine Anzahl von vergebenen Sternen, meistens 1–5. Die Sentiment-Analysetools, die wir uns angesehen haben, geben oft einen Zahlenwert für die Polarität aus, der zwischen 1 und -1 liegen kann. Es handelt sich hier nicht mehr um feste Klassen, also eine Klassifikation, sondern um Regression. Die “SemEval-2016 Task 4: Sentiment Analysis in Twitter” (Nakov et al. 2016) führte die Regression in den Wettbewerb ein.

Die Vorgehensweise von TextBlob und anderen Tools ist, ein Lexikon (Python-Dictionary) zu haben, in dem den Sentiment-Wörtern ein Wert zugeteilt wird, der den Polaritätswert quantifiziert, z. B.:

- gut: 0.5
- toll: 1.0
- schlecht: $-0,7$
- doof: $-1,0$

Wie aber kommt man zu diesen Polaritätswerten? Eine Möglichkeit ist, die Wörter in Textkorpora mit Sternchen-Annotationen (wie bei Amazon) automatisch zu klassifizieren und die Polarität nach der Wahrscheinlichkeit zu berechnen, mit der sie in positiven oder negativen Bewertungen vorkommen. Man kann auch für wenige Wörter von Hand Polaritäten eintragen und dann in einem Synonymwörterbuch diese Werte auf ihre Synonyme automatisch übertragen. Es ist weiterhin möglich, Texte von mehreren AnnotatorInnen mit Polaritätswerten annotieren zu lassen, dann die Annotationen zu vergleichen und schließlich die Polaritäten wie im Fall der Sternchen-Annotationen zu berechnen. Näheres dazu steht im Kap. 4.

Es gibt mehrere Möglichkeiten, die Polaritätswerte der einzelnen Wörter im Text zu aggregieren: Man kann die Werte addieren, den Durchschnitt berechnen, den höchsten oder niedrigsten Wert nehmen oder auch die Maximalwerte miteinander addieren. Wenn man die Polaritätswerte einfach addiert, dann bekommen längere Texte einen höheren Polaritätswert als kurze Texte, denn da kommen mehr bewertende Wörter vor, auch wenn ein kurzer Text z. B. stark negativ sein kann. Dazu kommt, dass man auf diese Weise den möglichen Wertebereich für Polaritätswerte nicht kennt. Daher wollen wir den Ergebniswert mit der Anzahl der Tokens im Text normalisieren und auf einen definierten Wertebereich beschränken. Dazu nutzen wir die “Min-Max-Skalierung” (siehe Raschka 2017, S. 120). Der Polaritätswert p wird skaliert in einen Polaritätswert p' . Die allgemeine Formel für diese Normalisierung sieht so aus:

$$p' = \frac{p - p_{min}}{text_length - p_{min}}$$

Bei p_{min} handelt es sich um den kleinsten Wert, den wir für p vergeben wollen. Wir ziehen also von p den kleinsten Wert ab und teilen das Ergebnis durch die Textlänge minus den kleinsten Wert. In unserem Fall ist der kleinste Wert ein negativer Wert, -1 . Die Formel für die Min-Max-Skalierung geht jedoch von einem Wert für p aus, der zwischen 0 und 1 liegt. Daher müssen wir die Formel etwas anpassen. Wenn $p = 0$ ist, also der Text neutral und ohne Meinungswörter ist, dann ist auch $p' = 0$ und wir müssen nicht weiter rechnen. Wenn $p > 0$ ist, es sich also um eine positive Meinung handelt, dann wird 1 zu p addiert und dann durch die Textlänge +1 geteilt:

$$p' = \frac{p + 1}{text_length + 1}$$

Bei negativen p -Werten müssen wir die Formel für die Skalierung etwas abändern, denn, wenn wir zu einem negativen Wert 1 addieren, schwächen wir den Wert ab, was wir nicht wollen. Also ist die Formel im Fall von negativen Werten:

$$p' = \frac{p - 1}{text_length + 1}$$

Sehen wir uns das an dem Beispiel aus dem GermEval-Korpus an, mit dem wir schon gearbeitet haben:

Re: DB Bahn Pünktlich zu meiner Reise :) perfekt.

Dieser Text besteht aus elf Tokens, von denen drei (“pünktlich”, “:”) und “perfekt”) sehr positiv sind. Wir geben allen den Wert +1 im Sentiment-Wörterbuch. Die Summe der Polaritätswerte für diesen Text (p) ist damit +3. Für die Skalierung berechnen wir also:

$$P' = \frac{3 + 1}{11 + 1} = 0,33$$

Ein kürzerer Text wäre:

Die Bahn ist perfekt.

Hier haben wir fünf Tokens, von denen einer (“perfekt”) sehr positiv (+1) ist. Die Summe der Polaritätswerte für diesen Text (p) ist also +1. Wir skalieren und kommen auf denselben Wert:

$$P' = \frac{1 + 1}{5 + 1} = 0,33$$

3.6 Dokumentklassifikation mit maschinellem Lernen

In der Sprachverarbeitung werden aktuell vor allem Methoden des maschinellen Lernens eingesetzt. Beim maschinellen Lernen von Klassifikationen geht es darum, aus Textdaten Modelle abzuleiten, mit denen neue Textdaten klassifiziert werden. Man unterscheidet dabei das “Supervised Learning” und das “Unsupervised Learning” (siehe Liu 2012, auch Raschka und Mirjalili 2019). Beim Supervised Learning stehen Dokumente zum Training zur Verfügung, die manuell klassifiziert sind. Da das häufig nicht der Fall ist, versucht man, auch aus nicht klassifizierten Daten Modelle abzuleiten. Diese Vorgehensweise nennt man “Unsupervised Learning”. Die Hauptmethode beim Unsupervised Learning ist der Wortlistenabgleich, wie wir ihn in einer einfachen Form auch schon getestet haben. Dabei liegt der Schwerpunkt der wissenschaftlichen Arbeit dann im Aufbau der Wortlisten. Das sehen wir uns im Kap. 4 näher an und schauen hier erst einmal auf Sprachmodelle des Supervised Learning.

3.6.1 Supervised Learning – Probabilistisches Sprachmodell

Das Supervised Learning (siehe auch Liu 2012, S. 24 ff.) basiert darauf, dass annotierte Daten zur Verfügung stehen, also in unserem Fall Dokumente, die bereits als positive, negative oder neutrale Bewertungen vorklassifiziert sind. Für Produktbewertungen können wir dabei z. B. auf Amazon-Reviews zurückgreifen, die ja von Sternen begleitet sind, die man als Annotation nutzen kann. Bei der GermEval Shared Task 2017 wurden von Hand klassifizierte Texte als Trainingsdaten zur Verfügung gestellt. Andere Datensätze aus Shared Tasks oder Bewertungsportalen sind ebenfalls verfügbar.

Das Ziel der Dokumentklassifikation ist ja, ein Dokument als positiv, neutral oder negativ zu bewerten. Um ein Modell aus den Daten abzuleiten, benötigt man außer der Klassifikation die Texteigenschaften der Dokumente. Diese Texteigenschaften können die Wörter im Text mit der Häufigkeit des Auftretens evtl. im Vergleich mit anderen Texten, die syntaktischen Kategorien der Wörter, die Sentiment-Wörter (die man mit einem Sentiment-Wörterbuch vergleicht), die Negationen oder die Abhängigkeiten (die Abhängigkeit der Wörter im Satz untereinander) sein. Das probabilistische Sprachmodell, das wir hier vorstellen, orientiert sich an der Beschreibung von (Heyer et al. 2006).

Beginnen wir damit, Wörter in Bewertungen als positiv, negativ oder neutral zu klassifizieren und damit das Dokument zu klassifizieren. Wir wollen für jedes Wort in den Dokumenten berechnen, wie oft es in negativen und positiven Dokumenten auftritt. Damit berechnen wir die Wahrscheinlichkeit für das Wort, positiv oder negativ zu sein. Diese Wahrscheinlichkeit benennen wir $PR(word|positive)$, bzw. $PR(word|negative)$. Die Wahrscheinlichkeit für ein Dokument d , positiv zu sein, ist dann $PR(pos|d)$.

Ein Mini-Textkorpus mit zwei Dokumenten, unser Trainingskorpus, soll das Vorgehen verdeutlichen:

1. *Das Handy ist super. (positiv)*
2. *Ich habe es gekauft und muss sagen, das Handy ist toll. (positiv)*

Das Testdokument, das klassifiziert werden soll, ist:

- *Das Handy ist toll.*

Zunächst wird die Wahrscheinlichkeit des Auftretens von “toll” im positiven Kontext berechnet. Dabei dividieren wir die Anzahl des Auftretens von “toll” im positiven Kontext durch die Anzahl aller Wortformen im Korpus:³

$$\text{PR}(w_i) = \frac{|w_i|}{\sum |wk|}$$

mit:

$|w_i|$: Anzahl der Vorkommen von w_i

$\sum |wk|$: Summe aller Wortformen im Korpus

Also für “toll”, das einmal im Korpus von 15 Wörtern auftritt:

$$\text{PR}(\text{toll}|\text{positive}) = \frac{1}{15}$$

Die Wahrscheinlichkeiten aller Wörter im Testsatz dafür, positiv zu sein, werden addiert. Da die Wörter “das”, “Handy” und “ist” jeweils zweimal im Trainingskorpus auftreten, ist der Wert für den gesamten Satz daher:

$$\text{PR}(\text{Das_Handy_ist_toll}|\text{positive}) = \frac{2 + 2 + 2 + 1}{15} = 0,47$$

Das Problem bei dieser Herangehensweise ist, dass der Kontext, also die Wörter rechts und links vom untersuchten Wort, verloren geht. Daher sehen wir uns Trigramme an, also Wortketten aus drei Wörtern, die aufeinander folgen. Um in unserem kurzen Testkorpus auch den Satzanfang mit einbeziehen zu können, erweitern wir die Dokumente um Markierungen für den Satzanfang⁴:

1. *S. 1 S. 2 Das Handy ist super.*
2. *S. 1 S. 2 Ich habe es gekauft und muss sagen, das Handy ist toll.*

Das Satzgewicht für den Testsatz in Bezug auf den Trainingskorpus ist dann ein Produkt aus den Wahrscheinlichkeiten für den Satzanfangsmarker S. 1, den Satzanfangsmarker S. 2, der auf S. 1 folgt, die Sequenz “S. 1, S. 2, das”, die Sequenz “S. 2, das, handy”, die Sequenz “das, handy, ist” und die Sequenz “handy, ist, toll”:

³(Heyer et al. 2006, S. 102)

⁴vgl. (Heyer et al. 2006, S. 106)

$$\begin{aligned}
& G(\text{Das Handy ist toll}) \\
&= PR(S.1) \cdot PR(S.1|S.2) \cdot PR(das|S.1, S.2) \cdot PR(handy|S.2, das) \cdot PR(ist|das, \\
& \quad handy) \cdot PR(toll|handy, ist)
\end{aligned}$$

Diese Wahrscheinlichkeit für ein Trigramm wie “das handy ist” wird auf dem Textkorpus so berechnet, dass die Anzahl der Vorkommen des Trigramms im Korpus durch die Anzahl der Vorkommen des Bigrams “das handy” geteilt wird, also $2/2$ in unserem Fall. Am Satzanfang teilt man durch die Anzahl aller Wortformen plus der Zahl der Wortanfänge. So berechnen wir dann das Satzgewicht:

$$\begin{aligned}
&= \frac{|S.1|}{|S.1| + |S.2| + \sum |wk|} \cdot \frac{|S.1, S.2|}{|S.1|} \cdot \frac{|S.1, S.2, das|}{|S.1, S.2|} \cdot \\
& \quad \frac{|S.2, das, handy|}{|S.2, das|} \cdot \frac{|das, handy, ist|}{|das, handy|} \cdot \frac{|handy, ist, toll|}{|handy, ist|} \\
&= \frac{2}{19} \cdot \frac{2}{2} \cdot \frac{1}{2} \cdot \frac{1}{1} \cdot \frac{2}{2} \cdot \frac{1}{2} = \frac{1}{38}
\end{aligned}$$

Jetzt muss man noch dafür sorgen, dass bei Wörtern im Testsatz, die im Trainingskorpus nicht vorkommen, nicht “0” im Produkt steht, wobei der ganze Wert dann 0 wäre. Das erreicht man durch eine Technik mit dem Namen “Smoothing”⁵: Von jeder Auftretenswahrscheinlichkeit im Text wird eine ganz kleine Summe abgezogen. Diese kleine Summe wird experimentell mit dem Trainingskorpus bestimmt.

3.6.2 Supervised Learning mit Features

Beim Supervised Learning auf Feature-Basis trainiert man ein Modell auf einer Liste von Merkmalen, die zuvor für den Text berechnet werden. So muss man zunächst entscheiden, welche Merkmale für das Training eines Modells interessant sein könnten, basierend auf der Zielsetzung des maschinellen Lernens. Hier ist ein Beispiel für ein Dokument im GermEval-Trainingskorpus:

```

<Document id="http://twitter.com/DOMKEYTV/statuses/733302306079379456">
<Opinions>
  <Opinion category="Zugfahrt#Pünktlichkeit"
    from="46" to="55" target="pünktlich"
    polarity="negative"/>
</Opinions>
<relevance>true</relevance>

```

⁵(Heyer et al. 2006, S. 108)


```
<sentiment>negative</sentiment>
<text>Waere ja mal ein Wunder wenn die deutsche Bahn pünktlich faehrt</text>
</Document>
```

Wir brauchen “relevance”, um nur die relevanten Tweets für das Training auszuwählen. Den Wert in “sentiment” brauchen wir, weil er die Klassifikation enthält, die wir trainieren wollen. Den Text brauchen wir, weil wir daraus weitere Merkmale berechnen wollen. Ein solches Merkmal kann unser Wortlistenvergleich sein, der einen Polaritätswert als Ausgabe herausgibt. Weiterhin könnte die Anzahl der Negationen und Verstärker im Satz interessant sein. Der wichtigste Schritt ist, die Merkmale auszusuchen und dann zu berechnen.

Hier ist eine Liste mit Negationen:

```
negations = ('NIE', 'nicht', 'nich', 'kein', 'keine', 'Keine', 'ohne', 'nie', 'nein', 'keiner',
'nichts', 'weder', 'Weder', 'garnicht', 'statt', 'Nix', 'nix', 'wäre', 'Wäre', ':-)', 'Gegen-
satz', 'kaum', 'Niemand')
```

Hier ist eine Liste mit Verstärkern:

```
verstaerker = ('sehr', 'total', 'enorm', 'häufig', 'wirklich', 'völlig', 'voellig', 'absolut',
'rein', 'endlich', 'vollstes', 'viel', 'hoffen', 'genug', 'Ziemlich', 'scharf', 'ziemlich',
'kolossalen', 'kolossale', 'kolossales', 'stark', 'hohes', 'hohe', 'zusätzlichen', 'lupen-
reinen', 'absolut', 'schleichend', 'definitiv')
```

Mit dieser Definition lassen sich die Negationen und Verstärker in einem tokenisierten Satz zählen:

```
def neg_emp_in_sentence(sent):
    negs = 0
    emps = 0
    for tok in sent:
        if tok in negations:
            negs = negs +1
        elif tok in verstaerker:
            emps = emps +1
    return(negs, emps)
```

Die Dokumente aus dem GermEval-Trainingskorpus im XML-Format werden nun so vorbe-reitet, dass sie im TSV-Format mit dem Polaritätswert aus unserem Wortlistenvergleich, der

Anzahl an Negationen und Verstärkern und der Annotation als positiv, negativ oder neutral stehen. Die nicht relevanten Dokumente werden aussortiert.

```

from xml.etree import ElementTree as ET
from wortlistenvergleich import *
dev_data = open(r"germeval_2017_dev.xml", "r")
out = open("out.txt", "w", encoding="utf-8")
tree = ET.parse(dev_data)
root = tree.getroot()

def convert_data():
    for document in root.iter('Document'):
        relevance = document.find('relevance').text
        sent = document.find('text').text
        sentiment = document.find('sentiment').text
        polarity = wortlistenvergleich(sent)
        (negs, emps) = neg_emp_in_sentence(sent)
        if relevance == 'true':
            out.write(str(sent) + '\t' + str(polarity)
                    + '\t' + str(negs) + '\t' + str(emps) + '\t'
                    + str(sentiment) + '\n')
    dev_data.close()
    out.close()

```

Ein Beispiel für eine Umwandlung in das neue Format ist:

```

<Document id="http://twitter.com/Ariantoser/statuses/
687328181582475265">
  <Opinions>
    <Opinion category=Allgemein\# Haupt" from="81" to="101"
      target=" ueber die Bahn aergern" polarity="negative"/>
  </Opinions>
  <relevance>true</relevance>
  <sentiment>negative</sentiment>
  <text>RT @Tryli: Wie schoen es ist wenn man
    sich nach nem nervigen Arbeitstag auch noch
    ueber die Bahn aergern muss.</text>
</Document>

```

RT @Tryli: Wie schön es ist wenn man sich nach nem nervigen Arbeitstag auch noch über die Bahn ärgern muss. -1.0 0 0 negative

Wir benennen die Datei um in “germeval_trainingdata.txt”.

Mit den so vorbereiteten Texten können wir jetzt ein Modell trainieren. Das hier vorgestellte Vorgehen orientiert sich an Jason Brownlee⁶.

Wir müssen einige Python-Module importieren, bevor wir beginnen: Pandas⁷ ist eine Bibliothek mit Werkzeugen für den Umgang mit Daten. Scikit-learn⁸, das wir als “sklearn” importieren, stellt Werkzeuge für das maschinelle Lernen bereit. Pickle⁹ benötigen wir, um das gelernte Modell abzuspeichern.

```
import pandas

import sklearn

import pickle

from sklearn import model_selection

from sklearn.tree import DecisionTreeClassifier
```

Zunächst lokalisieren wir die vorbereitete Eingabedatei:

```
input_data = (r"germeval_trainingdata.txt")
```

Dann verweisen wir auf die Bedeutung der Spalten:

```
names = ['text', 'polarity', 'neg', 'emp', 'cat']
```

⁶<https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>

⁷<https://pandas.pydata.org/about/>

⁸<https://scikit-learn.org/stable/>

⁹<https://docs.python.org/3/library/pickle.html>

Hier lesen wir jetzt die Datei mit ihren Spalten ein:

```
names = ['text', 'polarity', 'neg', 'emp', 'cat']
```

Im nächsten Schritt “erklären” wir die Features in der Datei: Die erste Spalte ist der Text (den wir ignorieren), danach kommen drei Zahlenwerte, die für die Berechnung genutzt werden, in der letzten Spalte steht die Kategorie, die wir klassifizieren wollen:

```
array = dataset.values
X = array[:,1:4]
X = X.astype('int')
Y = array[:,4]
Y = Y.astype('str')
```

Dann teilen wir den Korpus in 80 % Trainings- und 20 % Testdaten zufällig auf:

```
validation_size = 0.20 seed = 7 X_train, X_validation, Y_train,
Y_validation =
    model_selection.train_test_split(X, Y,
        test_size=validation_size, random_state=seed)
```

Es gibt verschiedene Klassifikatoren, die man jetzt ausprobieren könnte. Wir entscheiden uns der Einfachheit halber für einen, den Decision Tree Classifier, und trainieren unser Modell damit:

```
def train_model():
    classifier = DecisionTreeClassifier()
    classifier.fit(X_train,Y_train)
    model_file = 'finalized_model.sav'
    pickle.dump(classifier, open(model_file, 'wb'))
```

Das Modell ist nun in einer Datei mit dem Namen 'finalized_model.sav' gespeichert und wir können es anwenden. Wenn wir es auf einen Text anwenden wollen, dann müssen wir diesem Text dieselben Werte zuweisen wie den Trainingstexten, also Polarität nach Wortlistenvergleich, Anzahl der Negationen und Anzahl der Verstärker.

```
def pred_senti_sentence(sent):
    (neg, emp) = neg_emp_in_sentence(sent)
    polarity = sentiment_analysis(sent)
    sent_array = [[polarity,neg,emp]]
    model = pickle.load(open('finalized_model.sav', 'rb'))
    result = model.predict(sent_array)[0]
    return(result)
```

Jetzt kann man Texte prüfen und ein Gefühl dafür bekommen, wie gut das Modell ist:

```
>>> pred_senti_sentence("Wir leiden unter Bahnlärm")
'negative'
>>> pred_senti_sentence("Meine Strecke war stundenlang gesperrt")
'negative'
>>> pred_senti_sentence("Bei der Bahn ist WLAN kostenlos!")
'neutral'
```

Die Qualität der Klassifikation hängt einerseits von der Qualität der Merkmale und andererseits von der Größe und der Ausgewogenheit des Trainingskorpus ab.

Mit den Daten der GermEval stellen wir fest, dass die Klassifikation von positiven Texten nicht gelingt. Egal, wie das Modell trainiert ist, kein Satz wird als positiv klassifiziert. Ein näherer Blick in die Daten zeigt, warum das so ist: 590 negative und 1199 neutrale Texte stehen lediglich 151 positiven gegenüber. Wir haben also einen nicht ausgewogenen Datensatz. Die Gesamt-Accuracy ist damit einfach am höchsten, wenn das Modell annimmt, dass es keine positiven Meinungsäußerungen im Datensatz gibt. Wenn man nun mit einem ausgewogenen Datensatz trainiert, also mit je ca. 150 negativen, neutralen und positiven Texten, dann gelingt auch die Klassifikation mit dem trainierten Modell. Allerdings ist der Datensatz dann letztlich zu klein, um ein wirklich gutes Modell trainieren zu können. Man könnte sich auch vorstellen, die 151 positiven Texte im Datensatz zu erweitern, indem man sie kopiert und leicht modifiziert, also Varianten davon erzeugt. Das Problem des maschinellen Lernens auf nicht ausgewogenen Datensätzen und verschiedene Methoden, damit umzugehen, werden bei (Haixiang et al. 2017) beschrieben.

Forschungsgruppen haben mit weiteren Merkmalen experimentiert, wie der Zahl der Sentiment-Wörter im Text, dem maximalen Polaritätswert im Text, den negativen und den positiven Polaritätswerten, dem Polaritätswert des letzten Sentiment-Wortes im Satz oder den Tf-idf-Werten für alle Wörter im Text.¹⁰

¹⁰Beim Tf-idf-Maß handelt es sich um ein Maß aus dem Information Retrieval, bei dem die Häufigkeit des Auftretens eines Wortes in einem Dokument mit der Häufigkeit des Auftretens in der gesamten Dokumentmenge in Beziehung gesetzt wird. Für nähere Informationen dazu siehe (Ferber 2003).

3.6.3 Deep Learning

Ein großer Teil der aktuellen Veröffentlichungen im Bereich Sentiment-Analyse nutzt das sogenannte “Deep Learning”, wie man an den Tagungsbänden der SemEval Shared Tasks (z. B. Nakov et al. 2016; Rosenthal et al. 2019) sehen kann. Deep Learning ist eine Form des maschinellen Lernens mit neuronalen Netzen, das zunächst für die Bilderkennung entwickelt wurde und seit einiger Zeit auch für Aufgaben der Sprachverarbeitung eingesetzt wird. Voraussetzung dafür ist, dass der Text in ein Zahlenformat konvertiert wird. In diesem Zahlenformat erkennt der Lernalgorithmus Muster, die Grundlage der Modellbildung sind. In der einfachsten Form bekommt jedes Wort im Textkorpus einen Zahlenwert, durch den es ersetzt wird. Das nennt sich “one-hot encoding”. Interessanter sind jedoch die Word Embeddings, bei denen jedes Wort als Vektor repräsentiert ist, in dem kodiert ist, in welchem Kontext es auftritt. Aus den Word Embeddings lassen sich semantische Ähnlichkeiten zwischen Wörtern herauslesen. Wie Word Embeddings entstehen, erklären wir im Abschn. 4.5.

3.7 Zusammenfassung

Die Aufgabe bei der Dokumentklassifikation ist, zu einem Dokument (also z. B. einem Tweet) die Polarität zu bestimmen. Der erste Schritt dazu ist, die Daten zu bereinigen und die Texte in Sätze und Wörter zu tokenisieren. Die einfachste Methode ist ein Abgleich mit Wortlisten. Diese Wortlisten enthalten Sentiment-Wörter mit Informationen zu ihrer Polarität. Da wir nun mit der eigentlichen Programmierung angefangen haben, haben wir begonnen, einen Gold-Standard aufzubauen, um ständig über die Fortschritte unserer Programmierung informiert zu sein. Wir haben die Ergebnisse unserer Implementierungen mithilfe der Metriken Accuracy, Precision, Recall und F-Maß evaluiert. Anstelle einer einfachen Klassifikation (positiv – negativ – neutral) haben wir die Regression gesetzt und Polaritätswerte eingefügt. Anschließend haben wir Methoden zur Sentiment-Analyse auf Dokumentenebene mit maschinellern Lernen ausprobiert.

3.8 Übungen

1. Prüfen Sie Ihr Wissen:
 - Was versteht man in der Sentiment-Analyse unter “Polarität”?
 - Was sind die Aufgaben eines Tokenizers?
 - Wovon ist der Erfolg des Einsatzes von maschinellern Lernen abhängig?
2. Setzen Sie Ihr neues Wissen ein:
 - a) Wir werden jetzt unser erstes eigenes Mini-Programm für eine Sentiment-Analyse der deutschen Sprache erstellen.

- i. Stellen Sie eine Wortliste mit positiven Wörtern und eine mit negativen Wörtern auf.
 - ii. Schreiben Sie eine Funktion, die einen Eingabesatz tokenisiert. Nutzen Sie dafür entweder die Python-Funktion `split()` oder einen Tokenizer von `spaCy`¹¹ oder `TextBlob`.
 - iii. Schreiben Sie eine Funktion, die die Tokens im Eingabesatz mit den Wörtern in der Wortliste vergleicht.
 - iv. Testen Sie Ihre Funktion mit verschiedenen Sätzen und dokumentieren Sie das Ergebnis.
- b) Stellen Sie einen Gold-Standard zum Testen auf, bei dem es sich um eine einfache Tabelle mit zwei Spalten (Satz und Polarität) handelt.
- i. Verändern Sie Ihr Programm zum Wortlistenvergleich so, dass eine Testdatei eingelesen wird und das Ergebnis in eine Ausgabedatei ausgegeben wird. Dabei soll das Ergebnis eine zweite Spalte sein:

Satz	Polarität	Kategorisiert
Das ist ein gutes Produkt	Positiv	Positiv
Das ist ein Produkt	Neutral	Neutral
Das ist kein gutes Produkt	Negativ	Positiv
Das ist ein schlechtes Produkt	Negativ	Negativ

- ii. Schreiben Sie ein Programm, das die Ausgabe mit dem Gold-Standard vergleicht. Geben Sie dabei die falsch klassifizierten Sätze, den Accuracy-Wert und den F-Wert für jede Kategorie aus.
- c) Arbeiten Sie mit Polaritätswerten
- i. Machen Sie aus den Listen der positiven und der negativen Wörter ein Python-Dictionary und geben Sie jedem Wort einen Polaritätswert.
 - ii. Verändern Sie Ihren Wortlistenvergleich so, dass für jedes Wort der Polaritätswert ermittelt und im Text addiert wird, sodass das Ergebnis ein Polaritätswert für den Text ist.
 - iii. Normalisieren Sie diesen Polaritätswert, sodass er zwischen -1 und $+1$ liegt.
 - iv. Verändern Sie das Programm zur Evaluation so, dass die Zahlenwerte interpretiert werden (positive Werte: Klassifikation positiv, negative Werte: Klassifikation negativ, 0: Klassifikation neutral).
- d) Berechnen Sie die Wahrscheinlichkeiten, dass der Satz “Ich liebe die Bahn” positiv oder negativ ist, anhand der Trainingsdaten der GermEval 2017.
- e) Schreiben Sie ein Programm, das aus dem GermEval-2017-Datensatz eine Tabelle (csv) der relevanten Dokumente generiert, mit folgenden Spalten:

¹¹<https://spacy.io/>

- Text
- Addierte Polarität für Wörter aus dem Sentiment-Wörterbuch
- Zahl der Negationen
- Sentiment-Klassifikation aus dem Datensatz

Versuchen Sie, ein Modell für einen ausgewogenen Ausschnitt aus den GermEval-Daten zu trainieren und dann auch zu testen, so wie in diesem Kapitel beschrieben. Experimentieren Sie mit anderen Features.

Weitere Informationen dazu bekommen Sie unter

<https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>

3. Reflexion in Gruppenarbeit:

Vergleichen Sie die Ergebnisse Ihrer Implementierungen. Fassen Sie die Gemeinsamkeiten und die Unterschiede zusammen und besprechen, welche Vorteile erkennbar sind. Fassen Sie Ihre Gold-Standards zusammen und machen Sie daraus eine gemeinsame Gold-Standard-Liste.

3.9 Weiterführende Literatur

Eine grundlegende Einführung in Methoden des maschinellen Lernens finden Sie bei (Heyer et al. 2006). Die Dokumentklassifikation für deutschsprachige Daten war die Aufgabe B der “GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback” (Wojatzki et al. 2017b). An dieser Aufgabe nahmen alle Gruppen teil, sie kann also als die Basis-Aufgabe der Shared Task bezeichnet werden. Alle Gruppen nutzten existierende oder selbst für diese Aufgabe programmierte Tokenizer und es stellte sich heraus, dass sich die Ergebnisse mit einer Anpassung der Tokenisierung verbesserten.

Die Dokumentklassifikation für englischsprachige Twitter-Daten war eine Aufgabe in der “SemEval-2016 Task 4: Sentiment Analysis in Twitter” (Nakov et al. 2016). Der Tagungsband dieses Wettbewerbs für die englische Sprache zeigt, wie mit verschiedenen Methoden Tweets kategorisiert wurden. In den meisten Fällen wurde maschinelles Lernen verwendet, häufig auch “Deep Learning”. (Räbiger et al. 2016) zeigen in einer Tabelle auf, welche Features für das Maschinelle Lernsystem verwendet wurden: unter anderem Emoticons, Hashtags, negative und positive Wörter, Negationen. (Vilares et al. 2016) sind ein Beispiel für ein Deep-Learning-System auf der Basis von Word Embeddings.

Wir haben schon gesehen, dass in der Sentiment-Analyse Wörter und Wortlisten eine große Rolle spielen. Ein wichtiger Teil der Arbeiten ist es daher, die Abdeckung zu erweitern, also mehr bewertende Wörter zu erkennen. Dies sind vor allem Adjektive wie “schlecht”, “schön”, “schnell” oder “robust”. Aber auch Mehrwortlexeme spielen eine Rolle, wie “geht schnell kaputt” oder “bringt mich um den Verstand”. Der Kontext der Wörter kann dabei sehr wichtig sein, wie man bei “schnell” und “geht schnell kaputt” sehen kann. Um Wörter abgleichen zu können, müssen die Texte zunächst normalisiert werden (Abschn. 4.1). Der Rest des Kapitels beschäftigt sich mit der Gewinnung von Wörtern für Wortlisten in der Sentiment-Analyse. Wir zeigen die Option auf, existierende Wortlisten einzubinden (Abschn. 4.2), WordNet als Quelle zu nutzen (Abschn. 4.3), Wörter aus annotierten Textkorpora zu extrahieren (Abschn. 4.4) und Wörter aus nicht-annotierten Textkorpora zu extrahieren (Abschn. 4.5).

4.1 Normalisierung der Texte

Eine bessere Erkennung der meiningausdrückenden Wörter lässt sich zunächst damit erreichen, dass die Texte vereinheitlicht werden. Zwei wichtige Schritte sind hier zu nennen: Textnormalisierung und Lemmatisierung. Bei der Textnormalisierung schreibt man alle Wörter im Text (und in den Wortlisten) mit Kleinbuchstaben und löscht alle Bindestriche und Sonderzeichen, um sie besser vergleichen zu können.

Hier ist noch einmal das Beispiel aus dem GermEval 2017 Korpus:

Deutsche Bahn – Eine Horrorgeschichte Darf jetzt erstmal zum nächsten Bahnhof laufen, weil der Zug auf der Strecke stehengeblieben ist

Die normalisierte Version davon ist:

*deutsche bahn eine horrorgeschichte darf jetzt erstmal zum nächsten bahnhof laufen,
weil der zug auf der strecke stehengeblieben ist*

Die Ersetzung von Sonderzeichen wie Emojis, die zum Absturz von Python-Programmen führen können, kann z. B. mit diesem Code-Schnipsel realisiert werden:¹

```
def bmp(s):  
    return "".join((i if ord(i) < 10000 else '\ufffd' for i in s))
```

In einigen Fällen kann es jedoch sinnvoll sein, Emojis zu untersuchen und nicht zu ignorieren. Dafür gibt es z. B. das Python-Modul “emoji”.²

Bei der Lemmatisierung führen wir alle Wörter auf ihre Grundform (Lemma) zurück und müssen dann nicht mehr “gutes”, “guter”, “gute” in die Wortlisten aufnehmen, sondern nur noch “gut”. Damit geht Information über die Satzsyntax allerdings verloren, sodass man genau überlegen muss, an welcher Stelle dieser Schritt eingesetzt wird.

Für die Lemmatisierung haben Programme wie TextBlob und spaCy³ eigene Module, die in den meisten Fällen auf einfachen Wortlisten basieren, die den Wörtern ihre Lemmata zuordnen.

Die lemmatisierte Version ist nun:

deutschen bahn einen horrorgeschichte dürfen jetzt erstmal zum nächst bahnhof laufen,
weil der zug auf der strecke stehenbleiben sein

4.2 Einbindung eines existierenden Sentiment-Wörterbuchs

Eine weitere gute Möglichkeit, mehr lexikalische Einheiten zu erkennen, ist, existierende Wortlisten einzubinden. In Forschungsprojekten der letzten Jahre sind Listen von Wörtern und Phrasen entstanden, die frei zur Verfügung stehen und in neue Systeme eingebunden werden können. Auf der IGGSA-Webseite⁴ sind einige dieser Ressourcen gelistet. Hier sollen nur ein paar Beispiele genannt werden, denn es kommen ständig neue Ressourcen hinzu:

¹<https://frageit.de/questions/45715280/ucs2-codec-cant-encode-characters-in-position-6161>

²<https://pypi.org/project/emoji/>

³<https://spacy.io/>

⁴<https://sites.google.com/site/iggsahome/downloads>

Tab. 4.1 Beispiel für Einträge in SePL

Phrase	Opinion value	Standard deviation	Standard error	Phrase type	Manual correction
<i>Abartig</i>	−0,540	0,811	0,186	a	
<i>Abermals bestens gelungen</i>	0,800	0,000	0,000	a	m
<i>Abgedreht</i>	0,000	0,000	0,000	a	m

Die “SePL (Sentiment Phrase List)” der Universität Hof wird bei (Rill et al. 2012) beschrieben. Die Liste kann über die Projektwebseite⁵ angefordert werden. Sie enthält über 14.000 Einträge im csv-Format, die aus Produktbewertungen mit ihrer Sternewertung automatisch erzeugt und zum Teil von Hand korrigiert wurden.

Ein paar Beispiele aus dieser Liste sind in Tab. 4.1. Das “Polarity Lexicon” der Universität Zürich (Clematide und Klenner 2010) enthält ca. 8400 Einträge. Diese Einträge sind – mit ihren Polaritätswerten – manuell erzeugt worden. Zwei Beispiele aus dieser Liste:

pittoresk NEG=1 polemisch NEG=0.7

Die Basis dieser Wortliste sind literarische Texte, also eine ganz andere Textsorte als die Produktbewertungen der SePL.

Das “Multi-Domain Sentiment Lexicon for German”⁶ der Hochschule Darmstadt ist aus einem studentischen Projekt⁷ entstanden, in dem die lexikalischen Daten aus drei verschiedenen Wortlisten kombiniert wurden. Es enthält ca. 2900 Einträge im XML-Format. Hier ist ein Auszug:

<pre><entry> <term>menschlich</term> <opinion source="pressrelease dataset" polarity="1.0" /> <opinion source="MLSA" polarity="0.0" /> <opinion source="SentiWS" polarity="0.3324" /> </entry></pre>
--

⁵<http://www.opinion-mining.org/SePL-Sentiment-Phrase-List>
⁶<https://sites.google.com/site/iggsahome/downloads/OPM.zip?attredirects=0>
⁷Hauptverantwortlich für dieses Projekt war Kerstin Diwisch.

```

<entry>
  <term>schlecht</term>
  <opinion source="pressrelease dataset" polarity="-0.8333333" />
  <opinion source="MLSA" polarity="-0.11111111" />
  <opinion source="SentiWS" polarity="-0.7706" />
</entry>

```

Bei der Zusammenführung fiel auf, dass die Polaritätswerte der drei Ressourcen sich teilweise ganz wesentlich unterscheiden. Das ist einerseits darauf zurückzuführen, dass bei manuellen Annotationen unterschiedliche Standards angenommen werden, so wie “+”, “−”, “0” oder Fließkommazahlen, und andererseits darauf, dass für ganz unterschiedliche Domänen gearbeitet wurde. So ist z. B. der Terminus “gerecht” im politischen Kontext als stark positiv und im wirtschaftsorientierten Kontext als nur leicht positiv annotiert.

Bei der Auswahl einer Wortliste für die Integration ist es wichtig, die thematische Domäne zu beachten, denn literarische Texte können ganz andere bewertende Wörter enthalten als z. B. Social-Media-Bewertungen. Darüber hinaus müssen verschiedene Formate (CSV, Tabellen, XML, ...) in das Format überführt werden, das in der eigenen Implementierung gebraucht wird. (Schulz et al. 2017) beschreiben, wie sie in künftigen Implementierungen die Wörterbücher für spezielle Domänen aufbauen wollen: “A major improvement for the future would be to create a domain-dependent sentiment lexicon in order to capture specific words and phrases which in this particular context have a stronger polarity than in others.”⁸

4.3 Gewinnung von Sentiment-Wörtern mithilfe von WordNet

WordNet ist eine linguistische Ressource, die seit den 90er Jahren entwickelt wird (Fellbaum 1998). Die lexikalischen Einträge sind in Synonym-Gruppen – sogenannten “Synsets” – organisiert. Zwischen den Synsets sind Beziehungen wie Hyponymie und Antonymie definiert. Dazu kommen weitere Informationen, wie Definitionen und Sprachbeispiele.

Zunächst für die englische Sprache entwickelt, kamen weitere WordNets für viele andere Sprachen hinzu (Bond und Paik 2012), die in einer globalen Initiative miteinander verknüpft wurden (Bond et al. 2016). Die Ressource wurde für viele sprachtechnologische Anwendungen genutzt, so auch für die Entwicklung und Erweiterung von Wörterbüchern zur Sentiment-Analyse.

(Hu und Liu 2004) beschreiben, dass sie eine kleine Menge englischer Adjektive von Hand als positiv oder negativ bewertet haben und dann für diese Adjektive zunächst die Synonyme aus WordNet gesucht haben. Im nächsten Schritt haben sie die Antonyme mit

⁸Deutsch: Eine wesentliche Verbesserung für die Zukunft wäre die Schaffung eines domänenabhängigen Sentiment-Lexikons, um bestimmte Wörter und Phrasen zu erfassen, die in diesem speziellen Kontext eine stärkere Polarität aufweisen als in anderen. (eigene Übersetzung).

dem jeweils gegensätzlichen Wert bewertet und davon wiederum Synonyme gesucht. Das Opinion Lexicon⁹ enthält damit 6800 englische Adjektive, die als positiv und negativ klassifiziert sind.

(Baccianella et al. 2010) sind ähnlich vorgegangen, haben dann aber die Klassifikation direkt im WordNet aufgenommen, das WordNet also erweitert. Dabei haben sie auch die Adjektive in Definitionen für die Synsets mit derselben Polarität annotiert wie das Synset selbst.

(Naderalvojud et al. 2017, S. 20) testen drei verschiedene Sentiment-Lexika im Zusammenhang mit einer Analyse auf der Basis von “Recurrent Neural Networks” (RNN) und stellen fest, dass das auf WordNet basierte Lexikon SWN am besten funktioniert:

Furthermore, while the German SentiSpin lexicon does not improve the performance of the RNN model in the positive class, the proposed German SWN lexicons significantly improve its performance.¹⁰

Da auch WordNets für andere Sprachen als das Englische existieren, wie z. B. OdeNet¹¹ für die deutsche Sprache, sind diese Methoden übertragbar. Der Vorteil davon ist, dass man recht schnell zu einer großen Liste klassifizierter Wörter kommt. Es fehlt aber die Anpassung an die Domäne, die Gegenstand der Implementierung ist.

Hier sind Synonyme zu “gut” aus OdeNet:

[‘1a’, ‘O. K.’, ‘Seele von Mensch’, ‘abgemacht’, ‘akzeptiert’, ‘alles klar’, ‘alles paletti’, ‘angenehm’, ‘charmant’, ‘„d’accord“’, ‘da sage ich nicht nein’, ‘das ist ein Wort’, ‘dein Wille geschehe’, ‘dienlich’, ‘eins a’, ‘einverstanden’, ‘erbaulich’, ‘erfreulich’, ‘ergötzlich’, ‘erhebend’, ‘erquicklich’, ‘ersprießlich’, ‘es geschehe nach deinen Worten’, ‘es sei’, ‘fein’, ‘fruchtbar’, ‘förderlich’, ‘gebongt’, ‘gedeihlich’, ‘gefreut’, ‘geht in Ordnung’, ‘geht klar’, ‘gemacht’, ‘genehmigt’, ‘gewinnbringend’, ‘glücklich’, ‘gutmütig’, ‘günstig’, ‘gütig’, ‘herzensgut’, ‘herzerfrischend’, ‘herzerquicklich’, ‘hilfreich’, ‘ich nehme dich beim Wort’, ‘ist recht’, ‘lohnend’, ‘machen wir’, ‘manierlich’, ‘menschlich’, ‘nutzbringend’, ‘nutzwertig’, ‘nützlich’, ‘o. k.’, ‘okay’, ‘okey-dokey’, ‘opportun’, ‘pläsiertlich’, ‘positiv’, ‘roger’, ‘sachdienlich’, ‘schon überredet’, ‘schön’, ‘„so machen wir’s“’, ‘so sei es’, ‘sympathisch’, ‘tadellos’, ‘trefflich’, ‘von Nutzen’, ‘von Vorteil’, ‘von guter Qualität’, ‘vorteilhaft’, ‘warum nicht’, ‘wertvoll’, ‘wohl’, ‘wohltuend’, ‘zuträglich’]

⁹<http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon>

¹⁰Deutsch: Während das deutsche SentiSpin-Lexikon die Leistung des RNN-Modells in der positiven Klasse nicht verbessert, verbessern die vorgeschlagenen deutschen SWN-Lexika ihre Leistung deutlich. (eigene Übersetzung).

¹¹<https://github.com/hdaSprachtechnologie/odenet>

Synonyme zu “schlecht” aus OdeNet:

[‘(jemand) hätte mehr erwartet’, ‘Billig...’, ‘Hinterhof-...’, ‘Hintertreppen-...’, ‘Hobby-...’, ‘Küchen-...’, ‘Möchtegern-’, ‘Provinz-...’, ‘Wald- und Wiesen-...’, ‘am Boden’, ‘am Tiefpunkt’, ‘amateurhaft’, ‘arg’, ‘ausbaufähig’, ‘bedenklich’, ‘bescheiden’, ‘beschissen’, ‘billig’, ‘bitter’, ‘blöd’, ‘böse’, ‘böseartig’, ‘böse’, ‘böswillig’, ‘derb’, ‘dilettantenhaft’, ‘dilettantisch’, ‘drittklassig’, ‘dumm’, ‘dürftig’, ‘eher nicht’, ‘eher weniger’, ‘ernstlich’, ‘es gibt Entwicklungsbedarf’, ‘etwas Dummes’, ‘fadenscheinig’, ‘fies’, ‘flach’, ‘ganz unten’, ‘gemein’, ‘geringwertig’, ‘gut gemeint, aber schlecht gemacht’, ‘gut gewollt, aber schlecht gekonnt’, ‘halbwertig’, ‘hapern mit’, ‘hart’, ‘hobbyhaft’, ‘im Keller’, ‘insuffizient’, ‘kaum’, ‘laienhaft’, ‘lausig’, ‘leidig’, ‘lästig’, ‘mangelhaft’, ‘mau’, ‘medioker’, ‘mies’, ‘minderer Güte’, ‘minderwertig’, ‘misslich’, ‘mäßig’, ‘nachteilig’, ‘negativ’, ‘nicht (besonders) ambitioniert’, ‘nicht ausreichend’, ‘nicht besonders einfallreich’, ‘nicht den Erwartungen entsprechen’, ‘nicht ernst zu nehmen’, ‘nicht erwünscht’, ‘nicht genug’, ‘nicht in Ordnung’, ‘nicht rosig’, ‘nicht so gut’, ‘nicht so richtig’, ‘nicht wirklich’, ‘nicht wünschen’, ‘nicht wünschenswert’, ‘niveaulos’, ‘ohne Ambition’, ‘ohne Anspruch’, ‘ohne Niveau’, ‘platt’, ‘prekär’, ‘primitiv’, ‘schlechter Qualität’, ‘schlimm’, ‘schmerzlich’, ‘schmerzvoll’, ‘schrecklich’, ‘schwach’, ‘schwer’, ‘schwer zu ertragen’, ‘schwerlich’, ‘schädlich’, ‘seicht’, ‘steigerungsfähig’, ‘störend’, ‘stümperhaft’, ‘unambitioniert’, ‘unangenehm’, ‘unbequem’, ‘unerfreulich’, ‘unerquicklich’, ‘unerwünscht’, ‘ungenügend’, ‘ungut’, ‘ungünstig’, ‘unliebsam’, ‘unmöglich’, ‘unprofessionell’, ‘unqualifiziert’, ‘unschön’, ‘unter Soll’, ‘unwillkommen’, ‘unzulänglich’, ‘unzureichend’, ‘vermutlich kaum’, ‘vermutlich nicht’, ‘von Nachteil’, ‘wahrscheinlich kaum’, ‘wahrscheinlich nicht’, ‘wenig beneidenswert’, ‘widrig’, ‘wie in einem schlechten Film’, ‘wohl kaum’, ‘wohl nicht’, ‘zu wenig’, ‘zu wünschen übrig lassen’, ‘zweiten Ranges’, ‘zweitklassig’, ‘ärgerlich’, ‘übel’]

Es wird deutlich, dass diese Methode geeignet ist, um die Wortlisten der allgemeinen Sentiment-Wörter zu erweitern.

4.4 Gewinnung von Sentiment-Wörtern aus annotierten Korpora

Eine gute Methode, um Sentiment-Wörter zu gewinnen, die in der Domäne relevant sind, ist, sie aus einem annotierten Textkorpus automatisch zu extrahieren. Wenn man z.B. ein Sentiment-Analyse-Tool entwickeln möchte, das Texte aus sozialen Medien zum Thema “Bahn” analysieren soll, könnte man die annotierten Daten der GermEval Shared Task 2017¹² zugrunde legen. Diese Daten sind frei verfügbar, als tsv- und als xml-Datei. Jeder Text ist u.a. mit der Polarität der Meinungsäußerung annotiert. Im ersten Schritt werden die positiven, negativen und neutralen Texte in getrennten Dateien gesammelt. Um dafür das XML-Format zu verarbeiten, empfiehlt sich die “ElementTree XML API”¹³. Für das tsv-Format kann man das “CSV”-Modul von Python verwenden.¹⁴

¹²<https://sites.google.com/view/germeval2017-absa/data>

¹³<https://docs.python.org/3/library/xml.etree.elementtree.html>

¹⁴<https://docs.python.org/3/library/csv.html>

Nach Tokenisierung und Lemmatisierung extrahiert man alle Wörter aus den negativen, alle Wörter aus den positiven und alle Wörter aus den neutralen Sätzen und subtrahiert die Listen voneinander. Das Ergebnis davon ist allerdings noch nicht zufriedenstellend. So beginnt die Liste der positiven Wörter, die damit aus dem GermEval-Korpus extrahiert wurden, folgendermaßen:

['chauffiert', 'putzig', 't.co/IPJOlehVTB', 'Stratenschulte', 'Rückmeldung', 'ndr', 'Erleichterung', 'einwandfrei', 'Kompliment', 'abfragen', 'dramatisch', 'Entertainmentsystem', '1:40', 'HobbyIngenieur', '09:37:00', 'businessstraveller.de', 't.co/ZO9QnFqUYF', 'Idealfall', 'Jaaa', 'Storno', 'Ticket-Funktionalität', 'QR-Lesegeräte', 'denk', 't.co/Hbms3Ya9xa', 'Lokführer-Streiks', 'Handyticket', 'Heimweg', 'Mdr.de', 'kürzen', 'Niederschelderhütte', 'flyingdutchy04', 'Göleli', 'Einsatzplaner', 'weiere', 'Bahn-Streiks', 'gemach', 'Frühjahr', 'deinstallieren', 'schluden', 'feierlich', 'Tarifkompromiss', 'punkt', 'Halim', '10:18:00', 'Herrlich', ...]

Es scheint also sinnvoll zu sein, in erster Linie Adjektive zu extrahieren. Die bereits erwähnten Module TextBlob und spaCy haben einen "Part-of-Speech-Tagger", der die syntaktischen Kategorien der Wörter im Text bestimmt. Für die GermEval-Daten sind einige der auf diese Weise extrahierten Adjektive:

Positive: ['fair', 'feierlich', 'Herrlich', 'wirksam', 'Hoch', 'dramatisch', 'einwandfrei', 'unverzüglich', 'rheinisch', 'klimafreundlichen', 'strahlend', 'Herchen', 'kundenfreundlich', 'sanieren', 'machbar', 'Mitteldeutsche', 'skeptisch', 'legitim', 'hiesig', 'Rheinische', 'Mittelbahnsteig', 'einigen', 'wier', 'prompt', 'Zeig', 'einfach/Langweilig', 'Hausbahnsteig', 'hilfreich', 'mollig', 'Windeck-Herchen', 'idyllisch', 'perfekt', 'Weltweit', 'merkwürdig', 'widersprechen', 'kölnisch']

Negative: ['Steig', 'spezial', 'rücksichtslos', 'Umweltschonend', 'misstrauisch', 'unterhaltsam', 'selbstverständlich', 'aggressiv', 'nachfolgend', 'heilig', 'anwesend', 'Munchen', 'Islam', 'lächerlich', 'entgegengesetzt', 'Dulig', 'monatelang', 'blechen', 'Einzig', 'endgültig', 'unsinnig', 'Menschenfeindlich', 'konkret', 'geschützt', 'bundesdeutsch', 'unverschämt', 'Mutmaßliche', 'wildrotierend', 'S-Bahnsteig', 'ernsthaft', 'Tatsächlich', 'fähig', 'Groß', 'mutig', 'heldenhaft', 'Weitere', 'exklusiv', 'Möglich', 'Völlig', 'verscheuchen', 'Fremdenfreundlich', 'vorübergehen', 'leid', 'voraussichtlich', 'unbeständig', 'erkennbar', 'stündlich', 'belgisch', 'ärgerlich', 'dreiteilig', 'stressig', 'Regional', 'verbunden']

Einerseits wird deutlich, dass diese Listen überarbeitet werden müssen, bevor man die Wörter ins Sentiment-Wörterbuch übernehmen kann. Andererseits wird aber auch deutlich, dass auf diese Weise bewusste oder unbewusste Falschschreibungen der Social-Media-AutorInnen auftreten, die speziell für die Textsorte "Soziale Medien" sind und in Standard-Wortlisten nicht enthalten sind, wie "unzufärläsig" oder "verhungertambahnsteig". Diese sind für eine gute Erkennung äußerst relevant.

4.5 Gewinnung von Wörtern aus nicht annotierten Korpora

Im Internet sind sehr große Mengen an Text in sehr vielen Sprachen verfügbar. Dies ist eine enorme Chance, um Wortlisten zu gewinnen. Da diese Texte jedoch nicht annotiert sind, versucht man, andere Methoden zu finden, um Wörter zu klassifizieren.

Eine Methode ist die “Pointwise Mutual Information Measure (PMI)”. Sie basiert darauf zu erkennen, welche Wörter häufig mit bestimmten vorklassifizierten Wörtern (wie “gut” oder “schlecht”) auftreten und nutzt eine Suchmaschine dafür als Basis.

PMI berechnet sich aus der Wahrscheinlichkeit des gemeinsamen Auftretens zweier Terme:

$$\text{PMI}(\text{term}_1, \text{term}_2) = \log_2 \left(\frac{\Pr(\text{term}_1 \wedge \text{term}_2)}{\Pr(\text{term}_1)\Pr(\text{term}_2)} \right)$$

Dabei ist $\Pr(\text{term}_1 \wedge \text{term}_2)$ die Wahrscheinlichkeit, dass term_1 und term_2 zusammen im Satz auftreten und $\Pr(\text{term}_1)\Pr(\text{term}_2)$ die Wahrscheinlichkeit, dass term_1 und term_2 (unabhängig voneinander) auftreten. Der Logarithmus in der Formel dient dazu, besser darstellbare Zahlen zu bekommen, die weder extrem niedrig noch extrem hoch sind.

Mit der Suchmaschine google ergibt sich so ein PMI-Wert für die Wörter “gut” und “super” von 8,61, für die Wörter “gut” und “kalt” von 1,22. Es wäre also denkbar, aus Texten in der Domäne alle Adjektive zu filtern und mit den PMI-Werten für ihre Kombination mit “gut” und “schlecht” ihre Polarität zu bestimmen.

Eine andere Methode ist die der sogenannten “Word Embeddings” (Mikolov et al. 2013). Die grundlegende Idee ist, dass semantisch ähnliche Wörter in ähnlichen Kontexten auftreten. Um diese Kontexte mit maschinellen Lernverfahren trainieren zu können, werden Vektoren aufgestellt. Dabei wird jedes Wort durch einen Vektor repräsentiert, der für jedes andere Wort im Korpus einen Zahlenwert enthält. Dieser Zahlenwert ist die Wahrscheinlichkeit, mit der das aktuelle Wort in einem definierten Kontext mit dem Wort aus dem Korpus zusammen auftritt. Nehmen wir zur Illustration des Verfahrens diesen Mini-Textkorpus mit vier Sätzen:

- Die schwarze Katze miaut.
- Die schwarz-weiße Katze frisst.
- Der schwarze Hund miaut.
- Der schwarz-weiße Hund frisst.

Es gibt im Textkorpus 8 verschiedene Wörter:

[die, schwarze, katze, miaut, schwarz-weiße, frisst, der, hund]

Daher bekommt jedes Wort einen Indexwert, der zwischen 1 und 8 liegt:

- die: 1
- schwarze: 2

- katze: 3
- miaut: 4
- schwarz-weiße: 5
- frisst: 6
- der: 7
- hund: 8

Der Satz “Der schwarze Hund miaut” wird als ein Vektor repräsentiert, der eine 4×8 -Matrix ist:

der	00000010
schwarze	01000000
hund	00000001
miaut	00010000

Nun gilt es zu beachten, dass man diese Methode auf große Textkorpora mit mehreren tausend Wörtern anwendet, sodass die Matrix extrem groß wird. Mit einem Algorithmus mit dem Namen “word2vec” (Mikolov et al. 2013) wird der Vektor auf weniger Dimensionen reduziert und dabei dennoch der Kontext erhalten. Für jedes Wort wird berechnet, welche Wörter am wahrscheinlichsten davor und dahinter stehen. Das Kontextfenster kann dabei verschoben werden. Meistens ist dieses Fenster fünf Wörter groß, also zwei davor und zwei dahinter. Für ein Wort bekommen dann die Wörter ein höheres Gewicht, die häufig in diesem Fenster zusammen mit dem Wort auftreten.

Mit dieser Methode kann dann eine Wortliste mit Wörtern aus der Domäne erweitert werden, wenn ausreichend Textmaterial vorhanden ist, das aber nicht annotiert sein muss. Die Vektoren können aber im Zusammenhang mit annotierten Daten auch direkt für die Sentiment-Analyse mit Deep-Learning-Verfahren genutzt werden (siehe Abschn. 3.6.3).

4.6 Zusammenfassung

Wortlisten sind zentral für die Sentiment-Analyse, egal ob diese statistisch oder regelbasiert ist. Dieses Kapitel hat sich damit befasset, wie Wortlisten genutzt oder aufgestellt werden können. Eine Textnormalisierung ist dafür die Grundvoraussetzung, vor allem wenn wir mit Social-Media-Daten arbeiten. Eine Option, um an Wortlisten zu kommen, ist die Übernahme und ggf. Anpassung eines existierenden Sentiment-Lexikons. Dabei muss beachtet werden, dass die Polaritätswerte der einzelnen Lexika zum Teil erheblich voneinander abweichen und dass die Lexika auf sehr unterschiedlichen Daten basieren. Eine weitere Option ist die Nutzung von WordNet-Einträgen, die in Synonym-Mengen organisiert sind. Domänenspezifische Sentiment-Wörter bekommt man, indem man sie aus Korpora der jeweiligen Domäne extrahiert. Es gibt Methoden, die annotierte Korpora verwenden, aber auch solche, die mit nicht annotierten Korpora arbeiten.

4.7 Übungen

1. Prüfen Sie Ihr Wissen:

- Welche Möglichkeiten der Textnormalisierung gibt es?
- Was sind die Herausforderungen bei der Einbindung existierender Wörterbücher?
- Was ist WordNet und wie kann man die Ressource für die Sentiment-Analyse nutzen?
- Welche Möglichkeiten zur Gewinnung von Sentiment-Wörtern aus annotierten Korpora gibt es?
- Welche Möglichkeiten zur Gewinnung von Sentiment-Wörtern aus nicht annotierten Korpora gibt es?

2. Setzen Sie Ihr neues Wissen ein:

- a) Vergrößern Sie die Abdeckung Ihres Sentiment-Wörterbuchs durch Normalisierung. Reduzieren Sie Ihr Sentiment-Wörterbuch so, dass nur noch Lemmata darin stehen. Verändern Sie den Wortlistenvergleich, indem Sie die kleingeschriebenen Lemmata der Wörter im Text und im Wörterbuch miteinander vergleichen. Die Lemmatisierung kann z. B. mit spaCy oder TextBlob geschehen.
- b) Recherchieren Sie Sentiment-Wortlisten für die deutsche Sprache und integrieren Sie die gefundenen Wörter in Ihr Wörterbuch.
- c) Vergrößern Sie Ihr Sentiment-Wörterbuch durch WordNet-Ressourcen. Recherchieren Sie die multilingualen WordNets und OdeNet. Sehen Sie sich die Daten an und diskutieren Sie, wie man diese in der Sentiment-Analyse nutzen kann.
Fügen Sie die Synonyme zu “gut” und “schlecht” aus OdeNet in Ihr Wörterbuch der Sentiment-Wörter ein und testen Sie dann die Sätze aus Ihrem Gold-Standard. Die Liste enthält auch Mehrwortlexeme. Fügen Sie Ihrem Gold-Standard Sätze mit solchen Mehrwortlexemen hinzu. Sehen Sie Möglichkeiten, mit Mehrwortlexemen umzugehen?
- d) Nehmen Sie den Korpus der GermEval 2017 zur Hand und extrahieren Sie daraus negative und positive Adjektive. Erweitern Sie damit Ihr Lexikon.
- e) Vergrößern Sie Ihr Sentiment-Wörterbuch mit Word Embeddings
Für ein kleines Trainingsexperiment nehmen Sie die Installation von Nathan Rooy und passen Sie sie für unser Spielzeugbeispiel von oben an: <https://nathanrooy.github.io/posts/2018-03-22/word2vec-from-scratch-with-python-and-numpy/>
Aufgrund der großen Datenmengen, die benötigt werden und der riesigen Vektoren, die dabei entstehen, benutzen wir ein bereits vortrainiertes Modell für unsere Experimente. Die Universität Heidelberg stellt unter https://www.cl.uni-heidelberg.de/english/research/downloads/resource_pages/GermanTwitterEmbeddings/GermanTwitterEmbeddings_data.shtml ein Modell von Word

Embeddings zur Verfügung, das auf deutschsprachigen Twitterdaten aus den Jahren 2013 bis 2017 trainiert worden ist. Laden Sie das Modell herunter, entpacken Sie die “gz”-Datei und importieren Sie sie:¹⁵

```
from gensim.models import KeyedVectors

model = 'twitter-de_d100_w5_min10.bin' model =
KeyedVectors.load_word2vec_format(model, binary=True, limit=50000)
```

Die Ähnlichkeit von Wörtern wird folgendermaßen definiert:

```
def similar(w, top=10):
    try:
        for w, confidence in model.similar_by_word(w, topn=top):
            yield w, round(confidence, 2)
    except:
        pass
```

Jetzt können Sie schon Wörter suchen, die in ähnlichen Kontexten wie z. B. das Wort “blöd” auftreten. Testen Sie das mit verschiedenen Wörtern und nehmen Sie die Ergebnisse in Ihre Wortlisten auf:

```
for w, v in similar("blöd", 100):
    if v >= 0.7:
        print (v, w)
```

Spannend ist es aber auch, dass wir Werte für die Ähnlichkeit von Sätzen bekommen. Das geht folgendermaßen:

```
similarity_pos = model.wmdistance("Das ist total toll", "Ich bin froh und stolz darauf,
ein Mensch geworden zu sein, der Leuten die Bahn aufhält, Senioren beim Aussteigen hilft")

print("Positive Ähnlichkeit: "+"{:4f}".format(similarity_pos))

similarity_neg = model.wmdistance("Das ist absoluter Mist", "Ich bin froh und stolz darauf,
ein Mensch geworden zu sein, der Leuten die Bahn aufhält, Senioren beim Aussteigen hilft")

print("Negative Ähnlichkeit: "+"{:4f}".format(similarity_neg))
```

¹⁵Zunächst importieren Sie gensim, mit dem man auf das Modell zugreifen kann. Das Limit beim Import des Modells ist notwendig, um Ihren Rechner nicht zu überfordern.

Versuchen Sie, so einen Wert als Feature in Ihr System zum maschinellen Lernen einzubauen und testen Sie, ob sich die Accuracy dadurch verbessert.

3. Reflexion in Gruppenarbeit:

Diskutieren Sie in der Gruppe, welche Methoden der Wortlisten-Erweiterung am besten funktioniert haben und wo es Schwierigkeiten gab. Fügen Sie dann ihre entstandenen Wortlisten zusammen.

4.8 Weiterführende Literatur

Existierende Sentiment-Wörterbücher für die deutsche Sprache, die in eine Implementierung eingebunden werden können, sind auf der IGGSA-Webseite¹⁶ zu finden. Die Beiträge im Tagungsband der GermEval 2017 (Wojatzki et al. 2017b) zeigen, wie diese in verschiedene Systeme eingebunden worden sind. (Naderalvojud et al. 2017) sind einen anderen Weg gegangen, indem sie existierende englischsprachige Sentimentwörterbücher automatisch ins Deutsche übersetzt haben. (Hu und Liu 2004) und (Baccianella et al. 2010) sind Beispiele für den WordNet-Ansatz. Das Standardwerk für Word Embeddings ist (Mikolov et al. 2013).

¹⁶<https://sites.google.com/site/iggsahome/downloads>

Bisher haben wir uns ganze Dokumente angesehen und eine Entscheidung getroffen, ob dieses Dokument eine positive, negative oder neutrale Meinungsäußerung enthält. Diese Entscheidung beruht auf dem Vorhandensein von Sentiment-Wörtern, Negations- und Verstärkungswörtern im Text, ohne dass die syntaktische Struktur beachtet wird. Für kurze Texte wie Tweets ist das eine sinnvolle Herangehensweise. Manchmal wird man aber komplexere Texte betrachten wollen. Bewertungen von Büchern sind oft wesentlich ausführlicher und bestehen aus mehreren Sätzen. Andere Anwendungen von Sentiment-Analyse wollen z. B. die Bewertungen von politischen Ereignissen durch Politiker analysieren, indem Politikerreden analysiert werden. Ein anderes Anwendungsbeispiel ist die Analyse der Bewertungen einer Partei in Zeitungsartikeln. Dies sind komplexe Fälle, bei denen in einem Satz etwas Positives und im nächsten Satz etwas Negatives stehen kann.

Sehen wir uns den Anfang einer Buchrezension auf Amazon an¹:

Grundsätzlich ist Sebastian Löbners Einführung in die Semantik empfehlenswert. In 10 Kapiteln, die noch weiter unterteilt sind, werden die grundlegenden Aspekte der Semantik angerissen und vertieft. Am Ende der Kapitel gibt es dann noch weiterführende bzw. erläuternde Literaturempfehlungen und Übungsaufgaben.

Löbner ist Professor für Sprachwissenschaft und das merkt man beim Lesen des Buches: er weiß, wovon er spricht, das Thema wird gut vertieft, jedoch merkt man es auch an seiner oft umständlichen und komplizierten Sprache. Da hilft es dann auch nicht mehr, dass das ganze Buch sehr übersichtlich gestaltet ist und man oft veranschaulichende Grafiken und Tabellen vorgesetzt bekommt. Wenn der erläuternde Text dazu in verschachtelten Sätzen serviert wird und man sich das Ganze mehrere Male durchlesen muss, ist es trotzdem schwer zu verstehen. Jedoch sind die Grafiken und Tabellen sehr gut und dienen sicherlich dazu, das Thema besser zu verstehen.

Löbner bringt häufig Beispiele, anhand derer er seine Theorien erläutert. Das ist jedoch auch ein Manko des Buches. Oft stehen Beispiele für sich selbst und werden nicht explizit erläutert.

¹<https://www.amazon.de/product-reviews/B07G4PDB9R>

Es gibt darin einige Sätze, die nicht bewertend sind, wie z. B. „*In 10 Kapiteln, die noch weiter unterteilt sind, werden die grundlegenden Aspekte der Semantik angerissen und vertieft.*“ In diesem einen Dokument sind positive und negative Meinungsäußerungen wie „*Grundsätzlich ist Sebastian Löbners Einführung in die Semantik empfehlenswert.*“ und „*Das ist jedoch auch ein Manko des Buches.*“ Problematisch dabei ist, dass in einem Satz sowohl Positives als auch Negatives geäußert werden kann, wie „*Löbner ist Professor für Sprachwissenschaft und das merkt man beim Lesen des Buches: er weiß, wovon er spricht, das Thema wird gut vertieft, jedoch merkt man es auch an seiner oft umständlichen und komplizierten Sprache.*“ Das zeigt, dass auch die Satzebene nicht ausreicht, um die Meinung zu analysieren. Dennoch nehmen wir im Moment an, dass ein Satz nur eine Meinungsäußerung enthalten kann.

Um die Aufgabe der Sentiment-Analyse auf Satzebene zu lösen, muss der Text zunächst in Sätze unterteilt werden – die Satz-Tokenisierung. Für jeden Satz muss dann entschieden werden, ob er eine Meinungsäußerung enthält. Schließlich muss analysiert werden, ob diese Meinungsäußerung positiv, negativ oder neutral ist. Dabei werden Negationen und Gradpartikeln in die Analyse einbezogen.

5.1 Satz-Tokenisierung

Bei der Satz-Tokenisierung geht es darum, den Text in Sätze aufzuteilen, die dann einzeln analysiert werden können. Man könnte jetzt denken, dass dieses Problem ganz einfach zu lösen ist, indem man immer am Punkt das Ende eines Satzes annimmt. Allerdings gibt es auch in unserem Beispiel Abkürzungen wie „bzw.“ oder „z. B.“, in denen der Punkt Teil des Tokens ist und kein Satzende markiert. Es ist daher erforderlich, Listen von Tokens anzulegen, die einen Punkt in der Mitte (wie „o.ä.“) oder am Ende (wie „ca.“) haben, um diese als Ausnahmen beachten zu können.

(Heyer et al. 2006) geben darüber hinaus Regeln für die Satz-Tokenisierung:

Regeln für den Satzanfang:

- Sätze beginnen niemals mit Kleinbuchstaben.
- Nach einer Überschrift beginnt ein neuer Satz.
- Am Anfang eines Absatzes beginnt ein neuer Satz.
- Groß geschriebene Artikel (wie „Der“, „Die“, „Den“, ...) sprechen für den Satzanfang.
- Beginnt kein neuer Absatz, so steht vor dem neuen Satz ein Satzendezeichen.

Regeln für das Satzende:

- Sätze enden mit einem Satzendezeichen. Solche Satzendezeichen sind Punkt, Fragezeichen und Ausrufezeichen. Nach dem Satzendezeichen muss zusätzlich ein white space (meist ein Leerzeichen, s. u.) stehen. Achtung, Punkte können auch an anderer Stelle stehen, z. B. nach Abkürzungen oder Zahlen.
- Vor einer Überschrift endet ein Satz.
- Am Ende eines Absatzes endet ein Satz.
- Überschriften sollen wie Sätze behandelt werden.

Sprachverarbeitende Module wie spaCy haben oft eine eingebaute Satz-Tokenisierung, die auf ähnliche Weise funktioniert. Testen wir doch einmal den Satz-Tokenisierer von spaCy:

```
>>> document = 'Grundsätzlich ist Sebastian Löbners Einführung in die Semantik  
empfehlenswert. In 10 Kapiteln, die noch weiter unterteilt sind, werden  
\dots zu verstehen.'  
>>> ana = nlp(document)  
>>> for sent in ana.sents:  
    print (sent)  
Grundsätzlich ist Sebastian Löbners Einführung in die Semantik empfehlenswert.  
In 10 Kapiteln, die noch weiter unterteilt sind, werden die grundlegenden Aspekte  
der Semantik angerissen und vertieft.  
Am Ende der Kapitel gibt es dann noch weiterführende bzw. erläuternde  
Literaturempfehlungen und Übungsaufgaben.  
Löbner ist Professor für Sprachwissenschaft und das merkt man beim Lesen des  
Buches:  
er weiß, wovon er spricht, das Thema wird gut vertieft, jedoch merkt man es auch  
an seiner oft umständlichen und komplizierten Sprache.  
Da hilft es dann auch nicht mehr, dass das ganze Buch sehr übersichtlich gestaltet  
ist und man oft veranschaulichende Grafiken und Tabellen vorgesetzt bekommt.  
Wenn der erläuternde Text dazu in verschachtelten Sätzen serviert wird und man  
sich das Ganze mehrere Male durchlesen muss, ist es trotzdem schwer zu verstehen.  
Jedoch sind die Grafiken und Tabellen sehr gut und dienen sicherlich dazu, das  
Thema besser zu verstehen.
```

Der Punkt in der Abkürzung “bzw.” wird richtig als Teil des Tokens und nicht als Satzendeypunkt verstanden. Tatsächlich gibt es im Quellcode von spaCy eine Datei “tokenizer_exceptions.py” mit Abkürzungen, in der diese Zeile steht:

```
{ORTH: 'bzw.', LEMMA: 'beziehungsweise', NORM: 'beziehungsweise'},
```

Es ist möglich, diese Liste zu erweitern.

5.2 Identifikation von Sätzen mit Meinungsäußerungen

Auf der Dokumentenebene ist oft schon durch den Kontext gegeben, dass das Dokument eine Bewertung enthält. Bei Produktrezensionen ist es sehr unwahrscheinlich, objektive (nur beschreibende) Dokumente zu haben. Auf der Satzebene ist das jedoch anders, denn nicht jeder Satz in einer Rezension enthält auch eine Bewertung. Daher steht am Anfang eine Klassifikation von Sätzen als subjektiv oder objektiv.

Annotierte Daten auf der Satzebene sind sehr viel seltener als die auf der Dokumentenebene. Wenn man welche hat, kann man die Supervised Learning-Verfahren darauf anwenden. Features dafür sind die Anzahl der Pronomen, Adjektive und Modalverben oder auch die

Anzahl der Wörter, die in einem Sentiment-Wörterbuch gesammelt sind; genau wie beim Umgang mit Dokumenten. Eine andere Möglichkeit ist die, Sätze auf ihre Ähnlichkeit mit bereits annotierten Sätzen zu prüfen. Wenn im Trainingskorpus z. B. steht “Das Auto finde ich gut.” und jetzt der Satz “Dieses Telefon finde ich gut.” analysiert werden muss, kann der neue Satz als ähnlich dem Trainingssatz identifiziert und damit gleich klassifiziert werden.

Ohne annotierte Daten (“unsupervised”) sucht man nach bewertenden Phrasen im Text. Diese Phrasen werden in einem Lexikon gesammelt. Weiterhin untersucht man die Adjektive: Besonders graduierbare Adjektive (solche mit Steigerungsformen) deuten auf subjektive Sätze hin. Eine Auswertung von Hashtags und Emoticons kann hier ebenfalls weiterhelfen. Schließlich werden auch die Satzstrukturen untersucht: Konditionalsätze (die z. B. mit “wenn” beginnen) oder Fragesätze beinhalten eher keine Meinungsäußerung:

Wenn ich ein gutes Buch darüber kennen würde, würde ich es sofort kaufen.

Kennst Du ein gutes Buch zu diesem Thema?

Diese Klassifikation ist aber abhängig von der Domäne der Dokumente. Gerade in Social Media-Daten können Fragesätze durchaus subjektiv sein:

RT @Banane0711: @danintown Warum durfte die Bahn am Nordbahnhof ungehindert mehrere Tausend Eidechsen töten?

Kann man diesen ganzen Scheiß noch glauben..?

S21 # Bahn # Rückbau - nein, doch, oh - ist nicht barrierefrei. War der Zensor “beantworte die Frage nicht!” pinkeln?

5.3 Satzanalyse

Wenn die subjektiven Sätze im Dokument identifiziert worden sind, sind die Verfahren der Klassifikation zunächst dieselben wie die Verfahren zur Klassifikation von Dokumenten. Auf Satzebene wird jedoch der Kontext interessant, in dem sich die Sentiment-Wörter befinden. Dabei gibt es zwei Analysebereiche, die die Genauigkeit der Sentiment-Analyse beeinflussen: Gradpartikeln und Negationen. Da Gradpartikeln und Negationen im Zusammenhang mit weiteren Wörtern im Satz interpretiert werden müssen, spricht man hier von semantischer Kompositionalität.

Gradpartikeln stehen zusammen mit Adjektiven und Adverbien und geben die Intensität der Bewertung an. Zum Beispiel:

Ich finde das Produkt sehr schlecht.

Meistens verstärken sie das Adjektiv oder Adverb, das sie modifizieren. Abschwächende Gradpartikeln sind jedoch auch möglich:

Das ist ein bisschen ungünstig.

Für die Erkennung günstig ist, dass sie meistens direkt vor dem bewertenden Adjektiv oder Adverb stehen.

Bei Negationen ist der Normalfall, dass die Polarität umgedreht wird. So ist “nicht gut” eben das Gegenteil von “gut”. Aber Achtung: “nicht so gut” ist nur eine abgeschwächte Form von “gut”, nicht das Gegenteil.

Negation wird nicht nur durch das Wort “nicht” ausgedrückt, sondern kann auch in anderen Formen auftreten. Hier sind einige Beispiele:

- Mir hat es keinen Spaß gemacht.
- Mir macht es nie Spaß.
- Niemandem macht das Spaß.
- Ich denke nicht, dass mir das Spaß macht.

Christopher Potts schlägt eine einfache Methode vor, um mit Negationen umzugehen, die man auf Gradpartikeln übertragen kann²: Sobald eine Negation im Satz auftritt, wird jedem Wort zwischen der Negation und dem nächsten Satzzeichen ein _NEG angehängt. Dadurch entstehen zwei ganz unterschiedliche Tokens “gut” und “gut_NEG”, die dann auch unterschiedlich analysiert werden können.

Wir sehen aber schon an den Beispielen mit Negationen, dass diese Methode vor allem für die deutsche Sprache nicht immer funktioniert: Im letzten Beispiel ist das subjektive Wort “Spaß” von der Negation durch ein Komma getrennt. Gerade für Sätze mit “dass” müssen für das Deutsche andere Regeln aufgestellt werden. Ein anderes Problem ist die Wortstellung im Deutschen: Die Negation kann durchaus hinter dem bewertenden Wort stehen: “Spaß hat dabei wirklich niemand”. Auch Gradpartikeln können an anderer Stelle als direkt vor dem bewertenden Wort stehen: “Das interessiert mich wirklich sehr.” Auch (Schulz et al. 2017) stellen fest, dass für Negationen komplexere Verfahren notwendig sind:

Also, instead of just switching the polarity of a word based on the existence of negation words, a more fine-grained approach would be meaningful.³

Eine Grammatikanalyse der Sätze gibt die notwendige Information, um den Skopus (den Wirkungsbereich) von Negationen und Gradpartikeln zu bestimmen. Eine vollständige Grammatikanalyse mit HPSG⁴ für den Satz “Das Handy ist nicht gut.” sieht z. B. so aus wie in Abb. 5.1⁵.

²<http://sentiment.christopherpotts.net/lingstruc.html#negation>

³dt.: Anstatt nur die Polarität eines Wortes auf Basis der Existenz von Negationswörtern zu wechseln, wäre ein feinkörnigerer Ansatz sinnvoll. (eigene Übersetzung)

⁴Head-Driven Phrase Structure Grammar (HPSG) ist ein Grammatikformalismus, mit dem Texte analysiert und semantische, maschinenlesbare Repräsentationen für diese Texte erzeugt werden. Siehe (Pollard und Sag 1994)

⁵<http://gg.delph-in.net/logon>

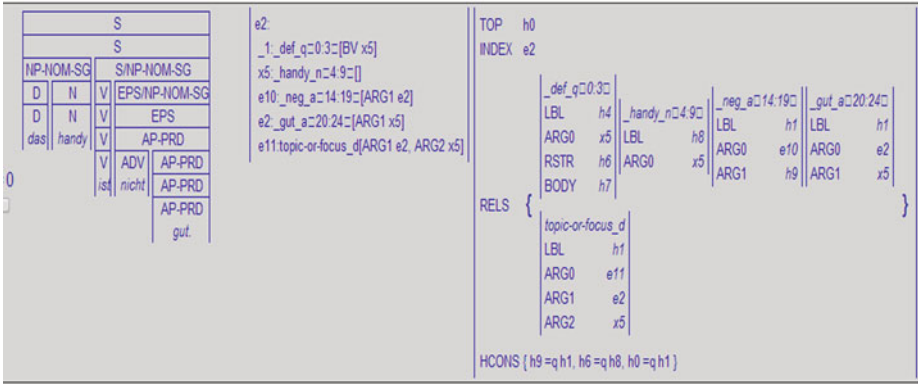


Abb. 5.1 HPSG-Analyse von “Das Handy ist nicht gut.”

Tab. 5.1 spaCy-Analyse von “Das Handy ist nicht gut” (nk: noun kernel element; sb: subject; ng: negation; pd: predicate)

Text	POS	Dependency	Head	Lemma
Das	DET	nk	Handy	Das
Handy	NOUN	sb	ist	Handy
ist	AUX	ROOT	ist	sein
nicht	PART	ng	gut	nicht
gut	ADJ	pd	ist	gut

Daraus, dass der Wert in “LBL” der Negation “_neg_a” derselbe wie der Wert in “LBL” des Adjektivs “_gut_a” ist, nämlich “h1”, kann man schließen, dass die Negation Skopus über das Adjektiv hat und den Polaritätswert daher umdreht.

SpaCy bietet eine einfachere Dependenzanalyse⁶, mit der die Abhängigkeiten der Wörter untereinander im Text analysiert werden. Für den Satz “Das Handy ist nicht gut” bekommen wir eine Analyse mit den Informationen in Tab. 5.17:

Die Negation hat hier unter “Head” das Adjektiv “gut” vermerkt, sodass wir auch hier herauslesen können, was ihr Skopus ist.

Wir sind ja bisher so vorgegangen, dass negative Polarität durch negative Werte und positive Polarität durch positive Werte dargestellt werden. Wir berechnen die Polarität eines Satzes daher so, dass wir im Fall von Negationen den Polaritätswert des dazugehörigen Sentimentworts mit -1 multiplizieren und im Fall von verstärkenden Gradpartikeln mit $1,5$.

⁶Eine Dependenzanalyse geht vom Verb im Satz aus und stellt die Abhängigkeiten der Wörter zueinander in einer Baumstruktur dar. Siehe dazu auch (Carstensen et al. 2009, S.281 f.)

⁷<https://spacy.io/api/annotation#section-dependency-parsing>

$$\text{Satzpolarität} = \sum_{i=1}^n KG \times Pol(sw_i)$$

Dabei ist:

- n: Zahl der Sentimentwörter im Satz
- KG: Kontextgewicht (−1 für Negationen und 1,5 für verstärkende Gradpartikeln)
- Pol(sw_i): Polaritätswert für das Sentimentwort, der im Sentimentwörterbuch steht

Für unseren Satz “Das Handy ist nicht gut” haben wir das Sentimentwort “gut” mit einer Polarität von 0,7 und die Negation “nicht”, sodass wir auf eine Polarität für den Satz von −0,7 kommen. Im Satz “Das Handy ist sehr gut” haben wir die Gradpartikel “sehr” und multiplizieren 0,7 mit 1,5, sodass wir auf einen Wert von 1,05 kommen.

5.4 Zusammenfassung

Sobald Dokumente, die Meinungsäußerungen enthalten wie z. B. Rezensionen, komplexer werden als kurze Social-Media-Bemerkungen, muss die Analyse auf die Satzebene gehen und die Sätze eines Dokuments einzeln analysieren. Zunächst müssen dafür die Sätze identifiziert werden, der Text muss in Sätze unterteilt werden – die Satz-Tokenisierung. Nicht jeder Satz in einer komplexen Rezension enthält eine Bewertung. Daher werden die Sätze im nächsten Schritt als subjektive und objektive Sätze klassifiziert, häufig anhand von Schlüsselwörtern und Emojis, aber auch mit der Satzstruktur. Konditionalsätze und Fragesätze müssen gesondert behandelt werden. Bei der Analyse subjektiver Sätze sind Lösungen für Gradpartikeln und Negationen notwendig. Für die deutsche Sprache werden komplexere Lösungen notwendig als für die englische, vor allem wegen der möglichen Wortstellungsvariationen, denn vor allem Negationen können sowohl vor als auch hinter dem Wort stehen, das negiert wird.

5.5 Übungen

1. Prüfen Sie Ihr Wissen:

- Worum geht es bei der Satz-Tokenisierung und was sind die Herausforderungen für die Arbeit mit der deutschen Sprache?
- Welche Möglichkeiten gibt es, Sätze auf Subjektivität und Objektivität zu prüfen?
- Was sind Gradpartikeln und Negation und wie beeinflussen sie die Sentiment-Analyse?

2. Setzen Sie Ihr neues Wissen ein:

- a) Erweitern Sie Ihre Sentiment-Analyse so, dass ein Text zunächst in Sätze segmentiert wird. Die Polarität der Sätze wird dann einzeln ausgegeben und am Ende zusammengezählt. Dies ist eine mögliche Ausgabe der Software:

```
>>> sentiment_analysis_p("Grundsätzlich ist Sebastian Löbners Einführung in  
die Semantik empfehlenswert. In 10 Kapiteln, die noch weiter  
unterteilt sind, werden die ...")
```

Grundsätzlich ist Sebastian Löbners Einführung in die Semantik empfehlenswert.
POL: 1.0

In 10 Kapiteln, die noch weiter unterteilt sind, werden die grundlegenden
Aspekte der Semantik angerissen und vertieft. POL: 0.0

Am Ende der Kapitel gibt es dann noch weiterführende bzw. erläuternde
Literaturempfehlungen und Übungsaufgaben. POL: -0.3

Löbner ist Professor für Sprachwissenschaft und das merkt man beim Lesen
des Buches: POL: 0.0

er weiß, wovon er spricht, das Thema wird gut vertieft, jedoch merkt man
es auch an seiner oft umständlichen und komplizierten Sprache. POL: 0.3

Da hilft es dann auch nicht mehr, dass das ganze Buch sehr übersichtlich
gestaltet ist und man oft veranschaulichende Grafiken und Tabellen
vorgesetzt bekommt. POL: 0.7

Wenn der erläuternde Text dazu in verschachtelten Sätzen serviert wird
und man sich das Ganze mehrere Male durchlesen muss, ist es trotzdem
schwer zu verstehen. POL: -0.7

Jedoch sind die Grafiken und Tabellen sehr gut und dienen sicherlich
dazu, das Thema besser zu verstehen. POL: 3.25

4.25

- b) Erweitern Sie Ihre Sentiment-Analyse so, dass ein Satz zunächst als subjektiv oder objektiv klassifiziert wird, bevor die Polarität bestimmt wird.
- c) Verändern Sie Ihr Programm zur Sentiment-Analyse so, dass Negationen und Gradpartikeln beachtet werden.
- Verändern Sie Ihr Programm zur Sentiment-Analyse so, dass beim Auftreten einer Negation die Polarität des nächsten Sentiment-Worts umgedreht wird.
 - Finden Sie eine Liste deutscher Negationen im Internet, die Sie verwenden können?

- Fügen Sie dann noch Gradpartikeln wie “sehr” oder “total” hinzu, die die Polarität erhöhen.
 - Testen Sie Ihr Programm mit dem GermEval-Korpus. Was funktioniert und was funktioniert nicht?
- d) Verändern Sie Ihr Programm so, dass eine Dependenzanalyse mit SpaCy durchgeführt wird und die Negation die Polarität seiner Head-Konstituente umdreht.
- Machen Sie dasselbe für Gradpartikeln, nur, dass diese die Polarität verstärken.
 - Vergleichen Sie das Ergebnis mit dem Ergebnis der Negation ohne Grammatik aus der vorangehenden Übung auf dem GermEval-Korpus.
3. Reflexion in Gruppenarbeit:
- Fügen Sie Sätze mit Negationen und Gradpartikeln in Ihren gemeinsamen Gold-Standard ein und prüfen Sie, ob Ihre Software diese richtig analysiert.

5.6 Weiterführende Literatur

Umfassende Informationen zur Satz-Tokenisierung finden Sie bei (Heyer et al. 2006). Der skizzierte Ansatz für Negation, der für englischsprachige Sätze gut funktioniert, findet sich auf der Webseite von Christopher Potts: <http://sentiment.christopherpotts.net/lingstruc.html#negation>. In (Pollard und Sag 1994) wird die HPSG-Analyse vorgestellt. Zu diesem Grammatikformalismus und seiner Umsetzung gibt es seitdem unzählige Publikationen. Die Webseite der internationalen Delph-In-Kooperation gibt hier weitere Hinweise: <http://www.delph-in.net>. Der Dependenz-Parser von spaCy wird hier beschrieben: <https://spacy.io/usage/linguistic-features#dependency-parse>.

Was bewertet wird: Aspekte identifizieren

6

In einigen Anwendungen reicht es nicht aus zu wissen, ob ein Dokument oder ein Satz eine positive oder negative Meinungsäußerung enthält. Wenn man sich z. B. die Meinungsäußerungen zur Deutschen Bahn aus der GermEval 2017 ansieht, dann werden unterschiedliche Aspekte der Bahn bewertet, wie Pünktlichkeit, Sauberkeit, Freundlichkeit des Personals usw. Es ist gut vorstellbar, dass das Management der Bahn nicht nur wissen möchte, wie die Bahn insgesamt von ihren Kunden bewertet wird, sondern auch, welche Aspekte schon sehr gut ankommen und welche noch verbessert werden müssen. Die “Subtask C” der GermEval 2017 wird daher so beschrieben¹:

Subtask C) Polarität auf Aspekt-Ebene

Identifizieren Sie alle Aspekte, die im Rahmen der Rezension positiv und negativ bewertet werden. Um die Vergleichbarkeit zu erhöhen, werden die Aspekte vorher in Kategorien eingeteilt (siehe Daten). Ziel der Teilaufgaben ist es daher, alle enthaltenen Kategorien und die damit verbundene Polarität zu identifizieren.

Beispiel:

alle so “Yeah, Streik beendet” Bahn so “okay, dafür werden dann natürlich die Tickets teuer”
Alle so “Können wir wieder Streik haben?” <tab>relevant<tab>neutral <tab>Ticketkauf#
Haupt:negativ Allgemein# Haupt:positive

In diesem Post besteht die Aufgabe darin, die Aspekte (und ihre Polarität) zu identifizieren:
Ticketkauf# Haupt:negativ Allgemein# Haupt:positive

In einem Satz können mehrere Aspekte einer Domäne unterschiedlich bewertet werden. Schauen wir z. B. noch mal in unsere Buch-Rezension aus dem Kap. 5:

er weiß, wovon er spricht, das Thema wird gut vertieft, jedoch merkt man es auch an seiner oft umständlichen und komplizierten Sprache.

¹übersetzt von: <https://sites.google.com/view/germeval2017-absa/home>.

In einem Satz werden gegensätzliche Meinungen (positiv und negativ) zu unterschiedlichen Aspekten (Vertiefung, Sprache) des Buches geäußert.

Aufgabe der aspektbasierten Sentiment-Analyse ist also, Aspekte zu identifizieren, über die eine Meinung geäußert wird, die Meinungsäußerungen zu identifizieren und zu klassifizieren und beides miteinander zu verknüpfen. Der nächste Abschnitt beschäftigt sich mit einer Taxonomie der Aspekte in einer Domäne. Anschließend sehen wir uns die sprachlichen Formen an, mit denen diese Aspekte realisiert werden können. Danach geht es darum, wie diese Aspekte im Text gefunden und interpretiert werden und schließlich um die Sentiment-Klassifikation der Aspekte.

6.1 Taxonomie der Aspekte

Wenn die aspektbasierte Sentiment-Analyse auf eine Domäne beschränkt ist, dann wird zunächst ein Datenmodell der Domäne mit ihren Entitäten und Aspekten aufgebaut. In einer Shared Task wie GermEval 2017 ist das geschehen, bevor die Texte annotiert worden sind. Dort gibt es zum Beispiel die Entität “Service/Kundenbetreuung” mit den Aspekten “Zugbetreuung”, “Am-Platz-Service/1. Klasse-Service” und “Sonstiges”. Die erste Aufgabe der Aspektklassifikation ist also die Aufstellung einer Taxonomie der Entitäten und Aspekte, deren Bewertungen analysiert werden sollen. In den meisten Fällen wird dies manuell und abhängig von der Zielsetzung der Analyse gemacht, denn die Aspekte, die für einen Nutzer interessant sind, sind für einen anderen Nutzer einer Sentiment-Analyse eventuell uninteressant. Außerdem gehören zur Domäne “Bahn” andere Aspekte als beispielsweise zur Domäne “Fahrrad”. Der Prozess des Taxonomie-Aufbaus kann durch eine automatische Terminologie-Extraktion unterstützt werden, wie er in (Siegel und Drewer 2012) skizziert ist. Aus der Liste der Terminologie, die in den Texten verwendet wird, lässt sich eine Taxonomie extrahieren. Eine Ressource wie OdeNet (Siegel 2020) kann Hinweise für die hierarchische Strukturierung geben. Eine andere Möglichkeit ist, die Meinungsausdrücke in den Texten zu identifizieren und die Ziele dieser Ausdrücke in die Aspekt-Taxonomie aufzunehmen.

Die Taxonomie der GermEval 2017 ist die folgende:

- Allgemein
- Atmosphäre
 - Lautstärke
 - Beleuchtung
 - Fahrgefühl
 - Temperatur
 - Sauberkeit allgemein
 - Geruch
 - Sonstiges

- Connectivity
 - WLAN/Internet
 - Telefonie/Handyempfang
 - ICE Portal
 - Sonstiges
- Design
- Gastronomisches Angebot
 - Verfügbarkeit Bordbistro/-restaurant
 - Verfügbarkeit angebotener Produkte
 - Vielfalt/Auswahl
 - Preise
 - Gastronomiebetreuung
 - Sonstiges
- Informationen
- DB App und Website
 - Informationen DB App und Website
 - Störungen DB App und Website
- Service/Kundenbetreuung
 - Zugbetreuung
 - Am-Platz-Service/1. Klasse- Service
 - Sonstiges
- Komfort/Ausstattung
 - Sitzkomfort
 - Funktionsfähigkeit Sitz und Sitzverstellbarkeit
 - Reservierung
 - Steckdosen
 - Kleiderhaken
 - Sauberkeit Sitzplatz
 - Sonstiges
- Gepäck
- Auslastung und Platzangebot
- Ticketkauf
- Toiletten
 - Funktionsfähigkeit Toiletten
 - Sauberkeit Toilette
 - Geruch Toilette
 - Verfügbarkeit Verbrauchsmaterial
 - Sonstiges
- Zugfahrt
 - Pünktlichkeit
 - Anslusserreichung

- Technische Schäden/Störungen am Zug
- Wagenreihung
- Fahrtzeit/Schnelligkeit
- Streckennetz
- Sonstige Unregelmäßigkeiten
- Reisen mit Kindern
- Image
 - Sponsoring
 - Marketing
- QR-Code
- Barrierefreiheit
- Sicherheit

Nicht alle Aspekte der Taxonomie sind jedoch in den Entwicklungsdaten annotiert. Z. B. gibt es keine Beispiele für Aussagen zum Aspekt “Gastronomiebetreuung”.

Bei der Arbeit mit annotierten Daten kann die Aspekt-Taxonomie aus der Annotation übernommen werden. In den GermEval-2017-Daten stehen diese im XML-Tag “category”:

```
<Opinions>
  <Opinion category="Sonstige_Unregelmässigkeiten\# Haupt"
    from="80" to="88" target="entfällt" polarity="negative"/>
</Opinions>
```

6.2 Phrasen-Lexikon der Aspekte

Im nächsten Schritt müssen Phrasen aus dem Text Aspekten dieser Taxonomie zugeordnet werden, um ein Lexikon der Aspekte und der dazugehörigen Phrasen aufzustellen. Wenn die Textphrasen als “Targets” in den Trainingsdaten annotiert sind wie bei GermEval 2017, können diese direkt extrahiert werden. Hier ist ein Beispiel, mit dem Target “Störung”:

```
<Document id="http://twitter.com/LaVieVagabonde/statuses/670623192583708672">
  <Opinions>
    <Opinion category="Sonstige_Unregelmässigkeiten\#Haupt" from="41" to="48"
      target="Störung" polarity="negative"/>
  </Opinions>
  <relevance>true</relevance>
  <sentiment>negative</sentiment>
  <text>@RMVdialog hey, wann fährt denn nach der Störung jetzt die nächste
    Bahn von Glauberg nach Ffm?</text>
</Document>
```

Eine Möglichkeit, die Phrasenlisten zu erweitern, besteht darin, Synonyme der extrahierten Phrasen hinzuzufügen. Diese Synonyme bekommt man aus Synonymlisten, Synonymwörterbüchern wie [openthesaurus.de](https://www.openthesaurus.de)² oder WordNets wie OdeNet (Siegel 2020). Bei einer Erweiterung der Liste mit OdeNet, das auf dem [openthesaurus.de](https://www.openthesaurus.de) beruht, bekommt man z. B. für das Wort “Abzocke” im Aspekt “Allgemein# Haupt” diese Synonymliste:

[‘Abzocke’, ‘Abzockerei’, ‘Bauernfängerei’, ‘Beschmu’, ‘Betrug’, ‘Beutelschneiderei’, ‘Fraud’, ‘Ganerei’, ‘Geldmacherei’, ‘Geldschneiderei’, ‘Manipulation’, ‘Nepp’, ‘Profitmacherei’, ‘Rosstäuscherei’, ‘Schmu’, ‘Schummelei’, ‘Schwindel’, ‘Trickserei’, ‘Täuschung’, ‘Wucher’, ‘krumme Tour’]

Aus diesem Beispiel ist schnell ersichtlich, wie sinnvoll eine Erweiterung mit Synonymen sein kann. Allerdings können Probleme auftreten, die darauf beruhen, dass Synonyme in unterschiedlichen Kontexten zu finden sind: Ebenfalls für den Aspekt “Allgemein# Haupt” ist “Bau” annotiert. Eine Suche nach Synonymen dafür ergibt Folgendes:

[‘Aushöhlung’, ‘Bau’, ‘Bauwerk’, ‘Bunker’, ‘Errichtung’, ‘Gebäude’, ‘Gefängnis’, ‘Gemäuer’, ‘Haftanstalt’, ‘Hafthaus’, ‘Haftort’, ‘Hohlraum’, ‘Häfen’, ‘Höhle’, ‘Höhlung’, ‘JVA’, ‘Justizvollzugsanstalt’, ‘Kahn’, ‘Kerker’, ‘Kiste’, ‘Kittchen’, ‘Knast’, ‘Konstruktion’, ‘Loch’, ‘Strafanstalt’, ‘Strafvollzugsanstalt’, ‘Vollzugsanstalt’, ‘Zuchthaus’, ‘schwedische Gardinen’]

Sicher ist im Kontext der Bahn das Auftreten des Nomens “Knast” sehr selten und nicht ein Hinweis auf den gemeinten Aspekt. Bei einer automatischen Erweiterung der möglichen Targets sollte daher immer manuell nachbearbeitet werden.

Wenn in den Trainingsdaten die Targets nicht markiert sind, gibt es die Möglichkeit, mit dem Wortschatz zu arbeiten. Man nimmt dann alle Wörter, die in Sätzen vorkommen, die mit einem Aspekt annotiert sind, und vergleicht sie mit den Wörtern aller anderen Sätzen. Dies ist dasselbe Verfahren wie bei der Gewinnung von Sentiment-Wörtern aus annotierten Korpora (Abschn. 4.4).

Sehen wir uns zum Beispiel die Wortliste für den Aspekt “Atmosphäre# Geruch” an:

[‘10’, ‘35’, ‘?’, ‘Alkoholunst’, ‘Alle’, ‘Alleine’, ‘Arbeit’, ‘Arschloch’, ‘Bahn’, ‘Bahnhof’, ‘BauerJaM’, ‘Berlin’, ‘Bett’, ‘Bild’, ‘Damit’, ‘Das’, ‘Der’, ‘Dermassen’, ‘Die’, ‘Diese’, ‘Es’, ‘Fürze’, ‘Genau’, ‘Grad’, ‘Hannover’, ‘Haus’, ‘Herunter’, ‘Hier’, ‘Immer’, ‘In’, ‘Irgendwo’, ‘Jetzt’, ‘Klo’, ‘Knoblauch’, ‘Koeln’, ‘Kommen’, ‘Leute’, ‘Lust’, ‘Man’, ‘Mann’, ‘Max’, ‘Mehr’, ‘Millionenhöhe’, ‘Mischung’, ‘Mit’, ‘Modernste’, ‘Monat’, ‘Nach’, ‘Neben’, ‘Nehmen’, ‘Nicht’, ‘Nichts’, ‘Noch’, ‘Nun’, ‘Ob’, ‘Pommes’, ‘Pommesflusterer’, ‘Punkt’, ‘Raucherabteile’, ‘Raus’, ‘Rt’, ‘S’, ‘S-Bahn’, ‘Sitz-

²<https://www.openthesaurus.de/>.

plätze', 'Slums', 'Sowieso', 'Spruch', 'Stadt', 'Station', 'Stinkepenner', 'Tiket', 'Tourist', 'UBahnen', 'Unbeschadet', 'Und', 'Verbrecher', 'Warum', 'Weiss', 'Wenn', 'Zeit', 'Zigarettenrauch', 'Zu', 'Zug', 'Zur', 'alle', 'ander', 'auf', 'aus', 'außen', 'bei', 'besetzen', 'bewerten', 'bezahlen', 'bleiben', 'cologneisnotberlin', 'das', 'dass', 'dazu', 'den', 'der', 'diesen', 'doch', 'du', 'dumm', 'eigentlich', 'ein', 'einen', 'einpferchen', 'erst', 'ertragen', 'es', 'finally', 'fordern', 'fragen', 'ganz', 'geben', 'grad', 'haben', 'hier', 'ich', 'ihre', 'ihren', 'im', 'in', 'kein', 'kommen', 'kotzen', 'können', 'letzt', 'löffeln', 'malen', 'man', 'mehr', 'me...', 'mir', 'mit', 'müssen', 'nach', 'nicht', 'nur', 'oe24.at', 'persönlich', 'pissen', 'renovieren', 'riechen', 'scheißen', 'schlimm', 'schön', 'sehen', 'sein', 'selbe', 'sich', 'sollen', 'sparen', 'stinken', 'stinkend', 'und', 'verreisen', 'versumpfen', 'viel', 'welch', 'werden', 'wie', 'win', 'wissen', 'wo', 'zu', 'zum', 'überfüllt']

Die Wörter, die nur in Sätzen zu diesem Aspekt und nicht in Sätzen zu anderen Aspekten vorkommen, sind:

['Mischung', 'Pommesflusterer', 'Klo', 'finally', 'pissen', 'stinken', 'win', 'cologneisnotberlin', 'Dermassen', 'Stinkepenner', 'Raucherabteile', 'Zigarettenrauch', 'Pommes', 'me...']

Die Qualität der mit dieser Methode gewonnenen Wortlisten ist stark abhängig davon, wie viele annotierte Texte verfügbar sind. Manuelle Nachbearbeitung ist auch hier notwendig.

Eine weitere Möglichkeit, das Lexikon der Phrasen für Aspekte zu erweitern, ist die Nutzung von Word Embeddings, wie im Abschn. 4.5 beschrieben. Mit dieser Methode ist es möglich, Wörter in einem sehr großen Textkorpus zu finden, die den bereits gefundenen semantisch ähnlich sind.

6.3 Aspekte im Text identifizieren und interpretieren

Mit dem Phrasenlexikon können jetzt Aspekte im Text identifiziert werden. Wenn im Text z. B. das Wort "Zigarettenrauch" auftritt, kann mit der Liste der Targets und Aspekte herausgefunden werden, dass es um den Aspekt "Atmosphäre# Geruch" geht. Problematischer als dieses Beispiel sind die Mehrwortlexeme. In der Liste der Targets steht z. B. "über die Bahn ärgern" als Target für den Aspekt "Allgemein# Haupt". Im Text steht das auch in derselben Form:

RT @Tryli: Wie schön es ist wenn man sich nach nem nervigen Arbeitstag auch noch über die Bahn ärgern muss.

Was aber, wenn jemand schreibt: "Ich ärgere mich über die Bahn" oder "über die Bahn muss ich mich immer ärgern"? In diesem Fall sollte man vielleicht das Target auf "Bahn" reduzieren und "ärgern" (mit allen flektierten Formen) als Sentiment-Wort aufnehmen. Ein anderes

Target aus der Liste ist “Ticket * online buchen” für den Aspekt “Ticketkauf#Haupt”. Der dazu passende Text, bei dem die automatische Erkennung der Polarität durch die enthaltene Ironie besonders komplex ist:

RT @pinokju: In 231 einfachen Schritten ein Ticket der Deutschen Bahn online buchen.

Dabei steht das Sternchen (*) für ein oder mehrere Wörter. Hier würde man vielleicht auch reduzieren auf “online buchen” und alle Online-Buchungen im Kontext der Bahn auf den Aspekt “Ticketkauf#Haupt” abbilden, auch wenn z. B. ein Sitzplatz gebucht wird.

Ein weiteres Problem sind Anaphern. Wenn ein Aspekt in einem Folgesatz wiederaufgenommen und dann erst bewertet wird, kann das mit einem Pronomen geschehen, wie in diesem Beispiel:

Die Bahn wird schneller...Wenn Sie mal kommt!

Eine Methode dafür ist, den Aspekt aus dem vorangehenden Satz so lange beizubehalten, bis ein neuer Aspekt explizit genannt wird. In (Sukthanker et al. 2018) werden weitere Methoden der Anaphern-Auflösung beschrieben.

Nicht alle Aspekte sind explizit mit eindeutigen Target-Wörtern identifizierbar. Im modellierten und recht eingeschränkten Kontext wie der Bahn ist das Target-Wort “teuer” immer ein Hinweis auf den Aspekt “Ticketkauf#Haupt”. Das könnte aber in anderen Kontexten anders sein: Nehmen wir als Beispiel die Domäne “Autokauf”. Dort kann natürlich das Auto teuer sein, aber auch die Extras, die Wartung oder die Garantieverlängerung. “Teuer” steht für den impliziten Aspekt “Preis”.

Noch schwieriger wird es, wenn der Aspekt umschrieben ist, wie in diesem Beispiel zum Aspekt “Atmosphäre#Temperatur”:

@JensK1002 @DB_Bahn mußten sie für den Sauna-Besuch zuzahlen ???

6.4 Aspektidentifizierung ohne Beschränkung auf eine Domäne

Bisher sind wir davon ausgegangen, dass wir die Domäne - das Themengebiet - genau kennen, in dem Aspekte bewertet werden. Nun könnte man sich vorstellen, dass eine aspektbasierte Sentiment-Analyse für unterschiedliche Domänen implementiert werden soll oder dass nicht genügend Zeit ist, um eine Taxonomie der Aspekte aufzubauen. Hier kommt die klassische Terminologie-Extraktion ins Spiel: Die “Fachwörter” werden aus dem Text als mögliche (explizite) Aspekte extrahiert. Dafür stehen grundsätzlich drei Methoden zur Verfügung, die auch kombiniert werden können: Die erste Möglichkeit ist, alle Wörter, die mit Großbuchstaben beginnen oder die aus Großbuchstaben bestehen, zu extrahieren. Diese Methode, die natürlich nur für die deutsche, nicht aber für die englische Sprache funktioniert,

erkennt vor allem Nomen und Namen. Die zweite Möglichkeit ist, alle Wörter zu extrahieren, die nicht zu den am häufigsten gebrauchten Wörtern der Sprache gehören. Hier ist z. B. ein Vergleich mit den 1000 häufigsten Wörtern des Deutschen, die durch den “Wortschatz Leipzig” zur Verfügung gestellt werden³, möglich. Die dritte Möglichkeit ist schließlich die Aufstellung von Pattern-Regeln, mit denen z. B. Ketten von Nomen oder auch Adjektive mit dahinterstehenden Nomen extrahiert werden.

6.5 Sentiment-Klassifikation des Aspekts

Da wir nun in der Lage sind, Sentiment-Ausdrücke im Text zu klassifizieren und Aspekte zu erkennen, gilt es, beides zusammen zu bringen. Die einfachste Methode dafür ist, beides im Satz zu finden und dann eine Verbindung anzunehmen. Wenn der Satz nur einen Aspekt-Ausdruck und einen Sentiment-Ausdruck enthält, ist das eine sinnvolle Methode. So erkennt die Aspekt-Klassifikation den Aspekt “Zugfahrt#Haupt” und die Sentiment-Klassifikation ein positives Sentiment in diesem Satz aus den GermEval-Daten:

@flyingdutchy04 und selbst wenn, würde ich eine reise mit der db vorziehen,

In den GermEval-Entwicklungsdaten gibt es nur wenige Fälle, in denen in einem Text Meinungen zu mehreren Aspekten geäußert werden. Die Polarität wechselt meist dabei über den Text nicht. Hier ist ein Beispiel dafür:

@DB_Bahn Wagen 21 fehlt. Reservierungen sind aufgehoben. Zug ist sehr voll.

Für dieses Beispiel ist es nicht unbedingt notwendig, die Satzgrenzen in die Analyse einzubeziehen, weil die Polarität in allen drei Sätzen negativ ist. Man nimmt also dieselbe Polarität für alle Aspekte im Dokument gleichermaßen an, ein Ansatz, den (Hu und Liu 2004) verfolgen.

Ein Beispiel für gegensätzliche Meinungen zu unterschiedlichen Aspekten ist das folgende:

Heute mal mit der @DB_Bahn zur Arbeit. Deutlich entspannter, aber doppelt so lange unterwegs

Der Aspekt “Zugfahrt#Haupt” wird positiv, der Aspekt “Zugfahrt#Fahrtzeit_und_Schnelligkeit” negativ bewertet. Dazu kommt, dass beide Bewertungen im zweiten Satz stehen, der Aspekt “Zugfahrt#Haupt” aber im ersten Satz. Der Schlüssel dazu ist das Wort “aber”. Ein ähnliches Beispiel:

³<https://wortschatz.uni-leipzig.de/de>.

@lokfuehrer_tim Schönen Feierabend! Ich bin heute zwar langsam, aber pünktlich ca. 225 km Bahn in 4 Stunden gefahren ...

Hier werden der Aspekt “Zugfahrt#Fahrzeit_und_Schnelligkeit” negativ und der Aspekt “Zugfahrt#Pünktlichkeit” positiv bewertet. Eine sinnvolle Vorgehensweise ist, im Fall von Wörtern wie “aber”, “trotzdem” oder “jedoch” den Text in zwei Teile zu teilen und diese getrennt zu kategorisieren.

Nicht immer sind Sentiment und Aspekt durch unterschiedliche Wörter oder Phrasen im Satz markiert. So gibt es Aspekte, die immer negativ sind, wie “Sonstige_Unregelmässigkeiten#Haupt” oder “Zugfahrt#Sonstige_Unregelmässigkeiten”. Diese Aspekte können als negativ markiert werden, ohne dass eine weitere Sentiment-Analyse notwendig ist.

Andererseits gibt es Sentiment-Wörter, die auch gleichzeitig implizite Aspekte bezeichnen, wie z. B. das Verb “stinkt” (negativ, Atmosphäre#Geruch) oder auch das Verb “teuer” (negativ, Ticketkauf#Haupt). Diese Wörter müssen in das Sentiment-Lexikon und das Aspekt-Lexikon aufgenommen werden.

Eine andere Möglichkeit, Sentiment-Ausdruck und Aspekt-Ausdruck in eine Beziehung zu setzen, ist die Nutzung eines Dependenzparsers. Dieser analysiert die Satzsyntax und stellt die Wörter im Satz in eine Beziehung. Dieses Verfahren kann für Social-Media-Daten problematisch sein, denn diese sind oft nicht standardsprachlich formuliert, sodass die Dependenzanalyse keine guten Ergebnisse liefert. Wenn man aber z. B. Zeitungstexte oder Buchrezensionen analysiert, kann das Verfahren genauere Ergebnisse liefern als das oben beschriebene Verfahren.

Die Dependenzanalyse von spaCy liefert für den einfachen Satz “Das ist die doofe Bahn” folgendes Ergebnis:

```
[('Das', 'PRON', 'sb', 'ist', 'Das'), ('sein', 'AUX', 'ROOT', 'ist', 'ist'), ('der', 'DET', 'nk', 'Bahn', 'die'), ('doofe', 'ADJ', 'nk', 'Bahn', 'doofe'), ('Bahn', 'PROP', 'pd', 'ist', 'Bahn')]
```

Für jedes Wort haben wir hier das Lemma, die syntaktische Kategorie, die Art der Dependenz, das Kopfwort in der Dependenzbeziehung und das Wort selbst. Das bewertende Adjektiv “doofe” hat als Kopfwort “Bahn”, woraus man schließen kann, dass hier die Bahn bewertet wird. Wir müssen also eine Dependenzanalyse durchführen, dann die bewertenden Wörter identifizieren und schauen, ob sie als Kopfwort ein Wort haben, das in der Liste der Targets steht. Wenn das der Fall ist, dann wird der dazugehörige Aspekt mit seiner Bewertung ausgegeben.

6.6 Zusammenfassung

Mehrere Aspekte einer Domäne können in einer Rezension – oft sogar in einem Satz – unterschiedlich bewertet werden. Die aspektbasierte Sentiment-Analyse versucht, Aspekte zu identifizieren, Meinungsäußerungen zu klassifizieren und dann beides miteinander zu verknüpfen. Um Aspekte zu identifizieren, wird im ersten Schritt eine Taxonomie der Entitäten und Aspekte der Domäne aufgestellt. Für diese Aspekte werden dann sprachliche Ausdrücke gesucht, die man einerseits in den Entwicklungsdaten und andererseits in externen Quellen wie Synonymlexika finden kann. Die so aufgestellten und klassifizierten Wort- und Phrasenlisten werden anschließend im zu analysierenden Text gesucht. Bei Phrasen ist das insbesondere im Deutschen alles andere als trivial. Schließlich müssen Aspekte und Sentiment zusammengeführt werden. Dabei ist es notwendig, auch Wörter wie “aber” oder “jedoch” in die Analyse einzubeziehen. Viele Systeme, die auf wissenschaftlichen Konferenzen vorgestellt werden, identifizieren mit Verfahren des maschinellen Lernens auf annotierten Trainingsdaten Aspekte und Polaritäten in getrennten Schritten und fügen diese dann satzbasiert zusammen, indem sie davon ausgehen, dass in einem Satz nur eine Meinung zu einem Aspekt geäußert wird, was in vielen Fällen auch funktioniert.

6.7 Übungen

1. Prüfen Sie Ihr Wissen:

- Warum ist es für viele Anwendungen wichtig, auch die bewerteten Aspekte zu erkennen?
- Welche Möglichkeiten gibt es, ein Phrasen-Lexikon für Aspekte einer Domäne aufzustellen?
- Welche Möglichkeiten gibt es, Aspekte im Text zu identifizieren?
- Wie kann man die extrahierten Informationen über Sentiment und Aspekt zusammenführen?

2. Setzen Sie Ihr neues Wissen ein:

- a) Nehmen Sie das GermEval-2017-Korpus und extrahieren Sie daraus die Aspekte aus dem Eintrag “<Opinion category=...”
- b) Extrahieren Sie die Targets für die Aspekte aus den GermEval-Daten.
- c) Erweitern Sie Ihre Phrasenliste mit Wörtern aus den GermEval-Texten.
- d) Erweitern Sie Ihre Phrasenliste mit weiteren Wörtern, z. B. mit der Methode PMI, mit Synonymen aus OdeNet oder mit Word Embeddings.
- e) Überarbeiten Sie die Target-Listen. Sehen Sie sich dabei besonders die Mehrwortlexeme an.
- f) Schreiben Sie eine Funktion, die – wenn sie im Text ein Target findet – den Text mit dem dazugehörigen Aspekt markiert.

- g) Identifizieren Sie Aspekt und Sentiment im Satz und führen Sie beides in einer Ausgabe zusammen, etwa so:

```
>>> aspect_and_sentiment("heut morgen schon im Zug die fahrt endet nie , wie  
weit würde es den bis zur Hölle dauern ? #Germany #Bahn #Silvester =D")
```

```
ASPECT: Zugfahrt#Fahrzeit_und_Schnelligkeit POLARITY: negative
```

Im Fall von Texten mit “aber” trennen Sie die Texte und geben Aspekt und Sentiment für die beiden Teile getrennt an, etwa so:

```
>>> aspect_and_sentiment("@lokfuehrer_tim Schönen Feierabend! Ich bin  
heute zwar langsam, aber pünktlich ca. 225 km Bahn in 4 Stunden gefahren ...")
```

```
ASPECT: Zugfahrt#Fahrzeit_und_Schnelligkeit POLARITY: negative
```

```
ASPECT: Pünktlichkeit POLARITY: positive
```

- h) Versuchen Sie, die Dependenzanalyse von spaCy zu nutzen, um Aspekt und Sentiment zusammen zu führen, etwa so:

```
>>> aspect_and_sentiment_spacy("Das ist die doofe Bahn.")
```

```
ASPECT: Allgemein#Haupt POLARITY: negative
```

3. Reflexion in Gruppenarbeit:

Überlegen Sie sich eine Domäne für Sätze in Ihrem Gold-Standard. Nehmen Sie in Ihren gemeinsamen Gold-Standard Sätze auf, die Aspekte dieser Domäne bewerten. Erkennt Ihre Software diese Sätze? Vergleichen Sie die Lösungen miteinander.

6.8 Weiterführende Literatur

In der GermEval Shared Task von 2017 (Wojatzki et al. [2017b](#)) haben von acht teilnehmenden Gruppen nur zwei an Task C (Identifizierung des Aspekts) teilgenommen: Eine Gruppe der TU Darmstadt (Lee et al. [2017](#)) und eine Gruppe aus Indien (Mishra et al. [2017](#)). Wojatzki et al. ([2017a](#)) beschreiben jedoch, dass beide nicht sehr erfolgreich waren:

Only (Lee et al. 2017) could outperform both provided baselines on the synchronic data. However, the improvements of 0,001 for aspect classification and 0,03 for aspect and sentiment classification are only slight.⁴

Eine multilinguale Shared Task für aspektbasierte Sentiment-Analyse war Teil der SemEval-Aktivitäten im Jahr 2016 (Pontiki et al. 2016). 29 internationale Gruppen haben am Wettbewerb teilgenommen. Deutsche Sprache war leider nicht Teil der Aufgaben.

(Schouten und Frasincar 2016) geben einen sehr umfassenden Überblick über die verschiedenen Lösungsansätze, die Forschungsgruppen für das Problem gefunden haben. Zu Methoden der Anaphern-Auflösung geben (Sukthanker et al. 2018) einen umfassenden Überblick.

⁴dt.: Nur (Lee et al. 2017) konnte beide vorgegebenen Baseline-Systeme auf den synchronen Daten übertreffen. Allerdings sind die Verbesserungen von 0,001 für die Aspekt-Klassifikation und 0,03 für die Aspekt- und Sentiment-Klassifikation nur gering. (eigene Übersetzung).

Wenn man sich die Ergebnisse der Analyse der GermEval-2017-Daten genauer ansieht, stellt man fest, dass ein relevanter Teil der negativen Tweets, die mit den bisher erstellten Verfahren nicht als negativ erkannt werden, ironisch sind. Ein paar Beispiele:

- Re: Deutsche Bahn Personenverkehr Danke deutsche Bahn wieder mal den Tag versüßt!!!! Sitzen in Göppingen fest! KEIN Schienenersatz KEINE Züge....was ein Tag!!
- Die Bahn hat ein neues nützliches Gadget hinzugefügt :Mücken
- Kaffee der Deutschen Bahn. Das können Sie genau so gut wie pünktlich sein. #CNL # DB #Verspätung
- RT @pinokju: In 231 einfachen Schritten ein Ticket der Deutschen Bahn online buchen.
- @DB_Bahn Heute mal wieder ein abenteuerlicher Tag mit euch! Sitzen jetzt in Köln HBF. Irgendwann gehn wir mal nen Kaffee trinken und reden!

Über die genaue Definition von Ironie gibt es einen schon lange andauernden Diskurs in der Forschungsliteratur, der bei (Karoui et al. 2019, S. 13–24) beschrieben wird. Auf S. 25 stellen die Autoren eine Gemeinsamkeit der Definitionen fest:

Although authors differ in their definition of irony, all agree that it implies a mismatch between what is said and the reality.¹

Diese Diskrepanz zwischen dem Gesagten und der Realität ist für die Sentiment-Analyse grundlegend relevant. Wir konzentrieren uns nicht darauf, genau zwischen Sarkasmus und Ironie zu unterscheiden. Für uns ist wichtig, das Gemeinte in der Meinungsäußerung zu erkennen. Ironische Meinungsäußerungen sind der Ausdruck einer negativen Meinung mit einer positiven Äußerung. Aus der Sicht der Sentiment-Analyse definieren wir ironische Texte als Texte, in denen eine negative Meinung mit positiven Ausdrücken geäußert wird,

¹ Obwohl sich die Autoren in ihrer Definition von Ironie unterscheiden, sind sich alle einig, dass sie eine Diskrepanz zwischen dem Gesagten und der Realität impliziert. (eigene Übersetzung).

oft auch übertrieben positiv. Ironische Äußerungen sind auch für Menschen nicht immer leicht zu erkennen (siehe auch (Farias und Rosso 2017, S. 114)). Da sie häufig in Social-Media-Beiträgen vorkommen können, gibt es eigene Shared Tasks dafür, mit denen versucht wird, Methoden für die automatische Erkennung zu entwickeln. 2018 gab es z.B. eine Shared Task zur automatischen Erkennung von englischsprachigen ironischen Äußerungen in Twitter, die SemEval-2018 Task 3 (Van Hee et al. 2018b). Die Twitter-Daten für diese Shared Task wurden mithilfe von Hashtags wie #not oder #irony gesammelt, dann aber noch mal manuell durchgeschaut. Die Hashtags wurden für den Wettbewerb gelöscht. Die eingereichten Systeme erreichten für die binäre Klassifikationsaufgabe (Ironie oder nicht) einen maximalen F-Score von 0,705 mit maschinellen Lernverfahren und Deep Learning und folgenden Features:

- Word Embeddings
- Sentence Embeddings
- Character Embeddings
- Sentiment
- Sentiment-Kontrast innerhalb eines Satzes
- PoS-Tags
- Emojis
- Großschreibung von Wörtern
- Satzzeichen
- bestimmte Wörter, wie z. B. Gradpartikeln

Dabei stellten die Organisatoren fest, dass für die binäre Klassifikationsaufgabe linguistisch-basierte Features am geeignetsten erscheinen (Van Hee et al. 2018b, S. 46):

A closer look at the best and worst-performing systems for each subtask reveals that Task A benefits from systems that exploit a variety of handcrafted features, especially sentiment-based (e.g. sentiment lexicon values, polarity contrast), but also bags of words, semantic cluster features and PoS-based features. Other promising features for the task are word embeddings trained on large Twitter corpora (e.g. 5M tweets). ...Neural network-based systems exploiting word embeddings derived from the training dataset or generated from Wikipedia corpora perform less well for the task.²

²dt.:Ein genauerer Blick auf die besten und schlechtesten Systeme für jede Teilaufgabe zeigt, dass Task A von Systemen profitiert, die eine Vielzahl von handgefertigten Merkmalen nutzen, insbesondere sentiment-basierte (z. B. Sentimentlexikon-Werte, Polaritätskontraste), aber auch Bag of Words, semantische Clustermerkmale und PoS-basierte Merkmale. Weitere vielversprechende Features für die Aufgabe sind Word Embeddings, die auf großen Twitter-Korpora (z. B. 5M-Tweets) trainiert worden sind. ...Systeme mit neuronalen Netzen, die Word Embeddings nutzen, die aus dem Trainingsdatensatz abgeleitet oder aus Wikipedia-Korpora generiert wurden, schneiden für diese Aufgabe weniger gut ab.(eigene Übersetzung)

In einer Shared Task für die Entdeckung von Ironie in drei Spanisch-Varianten 2019 (Ortega-Bueno et al. 2019) gewann ein System auf der Basis von Deep-Learning-Verfahren.

Wir gehen so vor, dass wir eine Ironie-Erkennung implementieren und in die Sentiment-Analyse integrieren. Wenn eine Äußerung als positiv erkannt wird, soll geprüft werden, ob sie vielleicht ironisch gemeint ist, sodass die Analyse dann angepasst werden kann. Ein großes Problem ist, ein geeignetes Textkorpus deutschsprachiger ironischer Meinungsäußerungen für die Entwicklung (und das Training) zu finden. Im Rahmen einer Lehrveranstaltung an der Hochschule Darmstadt haben wir daher ein kleines Textkorpus mit ironischen und nicht-ironischen Tweets zum Thema “Fußball” aufgestellt, das auf der zum Buch gehörenden Website verfügbar ist. Der größte Teil der ironischen Tweets in diesem Korpus wird mit automatischen Methoden allerdings wohl nicht erkennbar sein, wie diese Beispiele:

- Da hat der #Bvb aber mal die größten Stimmungskanonen zum #sportschauclub geschickt. #SGEBVB #DFBPokal #dfbpokalfinale
- “...hat den entscheidenden Elfer rausgeholt...” Ist ja auch so eine große Leistung, gefoult zu werden. ManManMan. #SGEBVB #dfbpokalfinale
- Kompliment an die Organisatoren. Eine Siegerehrung bei der null Emotionen rüberkommen. Das bekommt nicht jeder hin. #SGEBVB #DFBPokal

In wenigen Fällen ist Ironie mit Hashtags wie “#ironie” oder “#nicht” oder auch den Wörtern “Ironie” oder “IRONIE” gekennzeichnet. Diese Fälle sind durch ein Pattern-Matching einfach erkennbar:

@DFB_Frauen wie gut, dass die Übertragung der ARD so gut klappt...*Ironie aus*

7.1 Übungen

1. Prüfen Sie Ihr Wissen:

- Was ist Ironie im Kontext der Sentiment-Analyse?
- Welche Methoden gibt es in der Forschung zur Erkennung von Ironie?

2. Setzen Sie Ihr neues Wissen ein:

- a) Sehen Sie sich den Textkorpus Ironie auf der zum Buch gehörenden Website an und identifizieren Sie die Fälle, in denen Ironie durch ein Schlüsselwort (einen Hashtag, ein Wort wie “NICHT” oder ähnliches) im Text markiert ist.
- b) Optimieren Sie Ihre Sentiment-Analyse so, dass bei der Erkennung eines Schlüsselwortes für Ironie der Text als negativ erkannt wird, auch wenn er ansonsten als stark positiv klassifiziert würde. Wie viele der ironischen Äußerungen im Korpus können Sie auf diese Weise klassifizieren?

3. Reflexion in Gruppenarbeit:

Nehmen Sie ironische Äußerungen in Ihren gemeinsamen Gold-Standard auf und diskutieren Sie, mit welchen Methoden diese identifiziert werden könnten. Vielleicht könnte dies der Start eines Praxisprojekts oder einer Abschlussarbeit sein?

7.2 Weiterführende Literatur

Die Erkennung von Ironie ist eine sehr schwierige Aufgabe, für Menschen und für Computerprogramme erst recht. Einige Methoden aus der Shared Task von 2018 werden in (Van Hee et al. 2018b) dargestellt. Der Prozess des Aufbaus eines annotierten Textkorpus wird bei (Van Hee et al. 2018a) dargestellt. (Karoui et al. 2019) behandeln die automatische Erkennung von ironischen Tweets, stellen dabei eine Supervised-Learning-Methode für Französisch vor und prüfen zusätzlich ihre Übertragung in einen mehrsprachigen Kontext (Italienisch, Englisch und Arabisch). (Joshi et al. 2017) geben einen Überblick über verschiedene Methoden der Forschungsliteratur, regelbasierte und statistische Methoden.

Meinungsforschungsinstitute betreiben einen beträchtlichen Aufwand, um die Meinungstrends der Bevölkerung bezogen auf Politiker mit Telefon- und Straßenumfragen zu erfassen. Die Meinungsforschung scheint jedoch in einer Krise zu stecken. Bereits beim Brexit waren einen Tag vor dem Referendum die meisten Meinungsforschungsinstitute zum Schluss gekommen, dass die Briten in der EU bleiben wollen.¹ Aber auch bei der US-Wahl 2016 wurde in den meisten Fällen Clinton vorn gesehen.² Die automatische Sentiment-Analyse könnte hier eine Möglichkeit sein, mit viel geringerem Aufwand aktuelle Meinungstrends zu erkennen und dafür Twitter-Daten zu analysieren. Die Idee dahinter ist, dass die Plattform Twitter in Deutschland vielfach für politische Diskussionen genutzt wird.³ Eine Kernfrage der Untersuchung ist, ob Twitter als Quelle für Meinungsforschung dienen kann und klassische Meinungsforschung dadurch ersetzbar wird. Da sich Tweets auf einen Umfang von 140 Zeichen beschränken und das jeweilige Thema durch Hashtags meist eindeutig zugeordnet werden kann, scheinen sich Twitter-Daten gut für eine automatische Sentiment-Analyse zu eignen. Wir werden versuchen, diese Tweets automatisch in positive und negative Meinungsäußerungen zu klassifizieren. Anschließend versuchen wir, Tweets zu Politikern mit Veränderungen im ZDF-Politbarometer in Beziehung zu setzen. Damit nähern wir uns einer Antwort auf die Frage, ob man den Meinungstrend der Bevölkerung automatisch erfassen und damit letztlich klassische Meinungsforschung durch automatisierte Verfahren ersetzen kann.

Die Trendanalyse ist dabei natürlich nicht nur auf die Domäne der Politik und der Bewertung von Politikern beschränkt. Auch Firmen versuchen z. B. Trends zu erkennen und vor-

¹<http://www.zeit.de/politik/2016-06/umfragen-brexit-us-praesidentschaftswahl-demoskopie-einfluss>

²<http://www.morgenpost.de/politik/article208683451/Wie-Trumps-Wahlsieg-die-Demoskopen-in-die-Krise-stuerzt.html>

³In anderen Ländern ist z. B. auch Reddit sehr populär. Beispiele für Analysen von Reddit-Daten in Bezug auf Politiker sind (Nithyanand et al. 2017; Roozenbeek und Salvador Palau 2017).

auszusagen, um ihr Produktportfolio und ihr Marketing darauf auszurichten. Man könnte z. B. auch versuchen, den Ausgang des Zuschauervotings für einen Gesangswettbewerb im Fernsehen anhand von Tweets vorauszusagen. Die hier vorgestellten Methoden sind auf viele weitere Anwendungsgebiete übertragbar.

8.1 Aufstellung der Datenbasis

Für ein Experiment der Trendanalyse ist es zunächst notwendig, eine Datenbasis aufzustellen. In unserem Experiment sind das einerseits Tweets zu Politikerinnen und Politikern und andererseits Umfrageergebnisse der Meinungsforschung des ZDF-Politbarometers.

8.1.1 Tweets mit Meinungen zu Politikerinnen und Politikern

Twitter stellt für Forschungszwecke eine API zur Verfügung, die mit einer Python-Schnittstelle angesprochen werden kann. Damit lassen sich Tweets zu einem bestimmten Keyword oder Hashtag rückblickend für eine Woche aus der Twitter-Timeline extrahieren. Wir benötigen daher einen auf Python basierten Twitter-Crawler, der automatisch Tweets extrahiert, die sich mit einem der zehn Politiker beschäftigen, die auch im Politbarometer untersucht werden. Als erstes braucht man dafür einen Twitter-Account, der es erlaubt, über die API auf Tweets zuzugreifen. Mit diesem Account bekommt man Zugangstokens, die im Crawler-Programm eingefügt werden. Bei Twitter gibt es dafür eine Dokumentation unter <https://developer.twitter.com/en/docs>. Außerdem brauchen wir die Python-Module Twitter-Search und urllib, die über pip installiert und in Python importiert werden können. Hier ist der Code, mit dem mit einer Stichwortliste auf Twitter-Daten zugegriffen werden kann:

```
# Import aller Module, die wir für den Twitter-Crawler benötigen

from urllib.request import urlopen

import TwitterSearch

from TwitterSearch import *
from pprint import pprint
import json

# Hier setzen wir die Keywords, nach denen gesucht wird!

keywords = ["Merkel", "Steinmeier"]

try:
    tso = TwitterSearchOrder() # Erstellen eines TwitterSearchOrder-Objekts
```

```

tso.set_keywords(keywords) # Hier wird die Liste der Keywords aufgenommen
tso.set_language('de') # Hier wird die Sprache gesetzt
tso.set_include_entities(False) # wir wollen nicht die gesamte Information über alle Entities

# Das sind unsere geheimen Tokens, die wir von Twitter bekommen haben:
ts = TwitterSearch(
    consumer_key = 'XXX',
    consumer_secret = 'YYY',
    access_token = 'xxx',
    access_token_secret = 'yyy'
)

# Hier wird die Ausgabe erzeugt:
for tweet in ts.search_tweets_iterable(tso):
    print( '@%s tweeted: %s' % ( tweet['user']['screen_name'], tweet['text'] ) )

except TwitterSearchException as e:
    print(e)

```

Der nächste Schritt ist die Überlegung, welche Keywords eigentlich für die gesuchten PolitikerInnen relevant sind. So ist z. B. für eine Suche nach der Bundeskanzlerin Frau Merkel sicher auch das Keyword “Kanzlerin” relevant. Für die Aufstellung der relevanten Keywords ist es notwendig, in das aktuelle Twitter hinein zu sehen, mit einigen Keywords zu suchen und weitere aufzunehmen.

Für eine Trendanalyse ist der zeitliche Aspekt relevant, denn man möchte ja die Entwicklung der Meinungen zu einem Thema analysieren. Deshalb haben wir die Zeitangabe der Tweets mit extrahiert und können die gewonnenen Daten in Dateien nach Datum aufteilen, die wir einzeln analysieren.

Im studentischen Projekt “Opinion Mining” (Siegel et al. 2017) haben wir mit dem Twitter-Crawler automatisch Tweets extrahiert, die sich mit den zehn Politikern beschäftigen, die zu diesem Zeitpunkt auch im Politbarometer untersucht wurden. In einer Voruntersuchung wurden die relevanten Keywords und Hashtags erfasst, mit denen der Twitter-Crawler dann gestartet wurde. Für jeden der zehn Politiker entstand so eine Textdatei, die alle Tweets der letzten sieben Tage enthält. Jede Zeile entspricht einem Tweet. Insgesamt wurden von jedem beteiligten Politiker über vier Wochen insgesamt 800 Tweets gesammelt, eine Gesamtmenge von 8000 Tweets.

8.1.2 ZDF-Politbarometer

Der Politbarometer wird von der Forschungsgruppe Wahlen e. V. (FGW) seit 1977 betrieben. Zu den Hauptaufgaben zählen die Betreuung und die wissenschaftliche Beratung der Sendungen des Zweiten Deutschen Fernsehens (ZDF). Dabei werden Themenschwerpunkte wie politische Wahlen, Wählerverhalten, Meinungen zu politischen und gesellschaftlichen Fragen untersucht. Außerdem unterstützen und beraten sie bei der Verwendung von

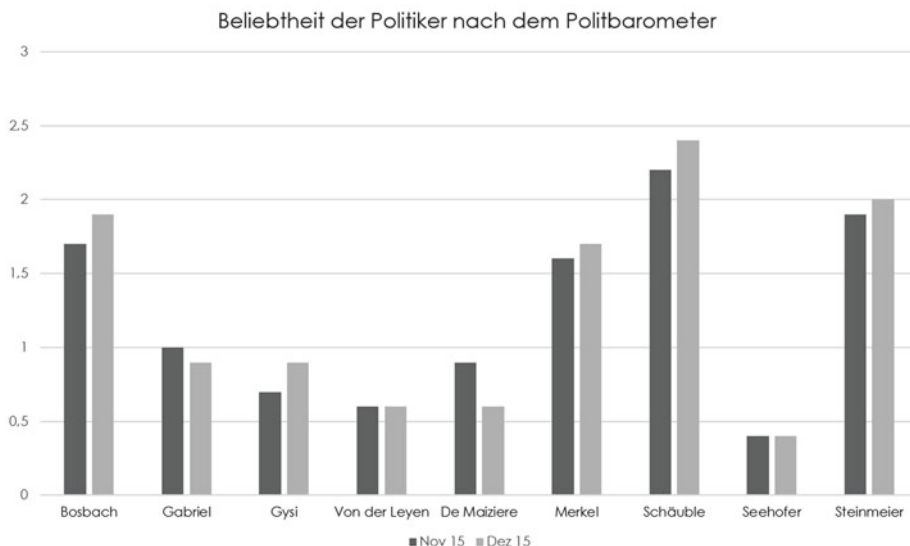


Abb. 8.1 ZDF-Politbarometer im Herbst 2015

sozialwissenschaftlichen Daten. Für diese Untersuchung ist besonders der Aspekt “Beobachtungen gesellschaftlicher Trends und Stimmungen” interessant. Die Politbarometer-Untersuchungen ermitteln die zehn aktuell beliebtesten Spitzenpolitiker. Die Erhebung erfolgt telefonisch (dienstags bis donnerstags). Zielgruppe sind Wahlberechtigte in der Bundesrepublik Deutschland. Es werden in den westlichen Bundesländern ca. 1000 zufällig ausgewählte Wahlberechtigte befragt, in den neuen Bundesländern 700. Um ein repräsentatives Ergebnis zu erhalten, werden die Befragungen im Osten überquotiert. Letztlich werden ca. 1250 Interviews evaluiert. Weiterhin wird eine Zufallsauswahl der Befragten vorgenommen und nach mehreren Faktoren gewichtet. Die Bewertungsskala der Politiker beläuft sich zwischen –5 bis 5.⁴ Für den Vergleich mit Twitter nehmen wir die Beliebtheitswerte (und die Veränderung dieser Werte) von zwei Monaten für ausgewählte PolitikerInnen von der Webseite. Im Oktober und November des Jahres 2015 haben wir mit Studierenden der Hochschule Darmstadt ein solches Experiment durchgeführt. Abb. 8.1 zeigt die Werte der Beliebtheitsskala für die untersuchten Politiker im November und Dezember 2015.

8.2 Sentiment-Analyse für Tweets

Im nächsten Schritt müssen die gesammelten Tweets daraufhin analysiert werden, ob sie Meinungsäußerungen enthalten und ob die Tweets positiv oder negativ sind. Für die Sentiment-Analyse haben wir ja bereits verschiedene Verfahren entwickelt, die hier zum

⁴<http://www.forschungsgruppe.de/Umfragen/Politbarometer/Methodik/>

Einsatz kommen können. Wenn die extrahierten Daten getrennt nach Politikern gesammelt wurden, dann ist in diesem Fall eine Sentiment-Analyse für Dokumente (Sentiment-Analyse auf Dokument-Ebene und nicht auf Satz-Ebene oder aspektbasiert) ausreichend. Um Trends zu erkennen, müssen die Tweets aber mit ihrem Datum gespeichert werden, denn die Untersuchung soll auch eine Veränderung über die Zeit beinhalten.

8.3 Zusammenfassung

In diesem Kapitel haben wir eine Anwendungsmöglichkeit für Sentiment-Analyse gezeigt, bei der politische Meinungstrends anhand von Twitter-Analysen vorausgesagt werden. Bei der Trendanalyse ist die Zeit des Social-Media-Beitrags eine wichtige Information. Wir haben gezeigt, wie Daten aus Twitter extrahiert werden können. Trendanalysen mithilfe von Sentiment-Analysen sind aber nicht auf politische Fragestellungen beschränkt. Sie werden z. B. für Börsenkurse (Minh et al. 2018) und Anforderungen an mobile Apps (Nagappan und Shihab 2016) gemacht.

8.4 Übungen

1. Prüfen Sie Ihr Wissen:

- Wie kann die Sentiment-Analyse zur Trendanalyse von politischen Diskussionen beitragen?

2. Setzen Sie Ihr neues Wissen ein:

- Erstellen Sie sich einen Account für die Twitter-API und nutzen Sie den Crawler, um nach Politikernamen zu suchen.
- Erweitern Sie den Crawler so, dass die Ergebnisse in eine Datei geschrieben werden.
- Suchen Sie aus dem ZDF-Politbarometer die aktuell untersuchten Politikernamen heraus und crawlen Sie Tweets zu diesen Politikern. Recherchieren Sie dazu die Keywords, die für die jeweiligen PolitikerInnen relevant sind. Diese sind neben den Namen auch Rollen, Schreibvarianten, Hashtags usw.
- Wenden Sie nun Ihre Sentiment-Analyse auf die extrahierten Tweets an und vergleichen Sie das Ergebnis mit dem Politbarometer. Können Sie eine Übereinstimmung feststellen?
- Wiederholen Sie das Experiment mindestens noch einmal eine Woche später und analysieren Sie den Trend der Beliebtheit.

3. Reflexion in Gruppenarbeit:

Diskutieren Sie in der Gruppe, ob ähnliche Sentiment-Analysen zur Trendanalyse verschiedener Social-Media-Daten auf Dauer die klassische Umfrage ersetzen könnte. Welche konkreten Chancen und welche Herausforderungen sehen Sie dabei? Diskutieren Sie die Rolle der Sentiment-Analyse bei Trendanalysen in anderen Bereichen wie z. B. für die Erkennung von Produkttrends.

8.5 Weiterführende Literatur

(Tumasjan et al. 2010) haben bereits 2009 versucht, mit der automatischen Analyse von Twitter die Ergebnisse der Bundestagswahl in Deutschland vorauszusagen. Dabei haben sie 100.000 Twitter-Nachrichten mit einer automatischen Text-Analyse-Software klassifiziert und die Ergebnisse mit den Ergebnissen der Bundestagswahl verglichen. Zu diesem Zeitpunkt war Twitter noch relativ neu, noch nicht mal vier Jahre alt. Die politische Diskussion in Twitter wurde von wenigen Nutzern beherrscht: “In sum, it becomes clear that, while Twitter is used as a forum for political deliberation, this forum is dominated by a small number of heavy users.”⁵ (Tumasjan et al. 2010, S. 181) Dennoch konnte eine klare Relation zwischen der Anzahl der Tweets zu einer politischen Partei und dem Wahlergebnis dieser Partei festgestellt werden: Je mehr Tweets zu einer Partei, desto besser das Ergebnis der Wahl. Die Abweichung war lediglich 1,65 %, also nicht viel schlechter als bei den Vorhersagen der “Forschungsgruppe Wahlen” mit einer Abweichung von 1,48 %. (Jungherr et al. 2012) versuchten zwei Jahre später, diese Analyse nachzuvollziehen und kamen zum gegenteiligen Ergebnis: “The number of party mentions in the Twittersphere is thus not a valid indicator of offline political sentiment or even of future election outcomes.”⁶ (Jungherr et al. 2012, S. 233).

(Maynard und Funk 2011) haben politische Tweets im Zusammenhang mit den britischen Parlamentswahlen im Jahr 2010 untersucht und automatisch Meinungsäußerungen klassifiziert. Im letzten Schritt haben sie aber die Klassifikation der Tweets manuell korrigiert, denn die automatische Klassifikation hatte eine Präzision von 62,2 %, was nicht aussagekräftig genug ist, um z. B. Wahlprognosen abzugeben.

⁵dt.: Insgesamt wird deutlich, dass Twitter von einer kleinen Anzahl sehr aktiver Nutzer dominiert wird, während es als Forum für politische Deliberation genutzt wird. (eigene Übersetzung).

⁶dt.: Die Anzahl der Erwähnungen einer Partei in Twitter ist daher kein valider Indikator für Offline-Meinungen oder Wahlergebnissen. (eigene Übersetzung).

Die Meinung der Kunden ist ein wichtiger Geschäftsfaktor geworden. Firmen wollen wissen, was Verbraucher von ihrem Produkt oder ihrer Dienstleistung halten, und versuchen, sich und ihre Produkte schnell an Kundenbedürfnisse anzupassen und die geäußerten Meinungen bestenfalls als Marketinginstrument einzusetzen. Manchmal wird man mittlerweile sogar nach Benutzung einer Toilette gebeten, auf einen lachenden oder weinenden Smiley zu drücken. Für Firmen kann es existenziell sein, vor einem aufkommenden “Shitstorm” rechtzeitig gewarnt zu werden. Gleichzeitig kann nur derjenige schnell auf Trends reagieren, der diese auch schnell erkennt.

Mit der Zunahme der Relevanz der Kundenmeinungen steigt jedoch auch die Anzahl der Manipulationsversuche. Systematische Untersuchungen zum Anteil der Fake-Bewertungen an den gesamten Bewertungen gibt es bisher nicht, man muss wohl auch davon ausgehen, dass die Anteile je nach Branche sehr unterschiedlich sind. Die Betreiber von Portalen werden diese Zahlen wohl auch nicht veröffentlichen, sofern sie sie haben. Schätzungen sprechen von 20–30 %, z. B. (Mukherjee 2015). Diese Schätzungen betreffen aber englischsprachige Meinungsäußerungen bezogen auf eine begrenzte Anzahl von untersuchten Portalen. In einer Befragung von Hoteliers, durchgeführt an der FH Worms, haben fast die Hälfte der Hoteliers angegeben, Erfahrungen mit gefälschten Bewertungen zu haben. Selbst Erpressungen durch Gäste wurden von Hotelbesitzern berichtet, wobei die Gäste einen Preisnachlass forderten und ansonsten mit negativen Bewertungen drohten (Conrady 2015).

Für die Kunden bedeutet das, dass sie sich nicht immer auf Online-Bewertungen verlassen können und erheblich getäuscht werden. Bewertungen sind Bestandteil des Geschäftsmodells vieler Portale. Da ihre Glaubwürdigkeit erheblich unter den Manipulationen leidet, gehen die Betreiber der Online-Portale mittlerweile gegen Opinion Spam vor, wie z. B. Amazon¹. Fake-Bewertungen haben sich zu einem eigenen Geschäftsmodell entwickelt. Es gibt

¹<http://www.heise.de/newsticker/meldung/Amazon-verklagt-Haendler-wegen-eingekaufter-Fake-Bewertungen-3227189.html>.

Plattformen und Anbieter für gekaufte Reviews, die ständig öffnen und wieder schließen, wie noch bis Frühjahr 2016 buyamazonreviews.com, fiverr.com oder reselleratings.com. Inzwischen wird man bei diesen Seiten umgeleitet auf eine Seite von Amazon², die dem Nutzer droht: “If we determine that you have attempted to manipulate reviews or violated our guidelines in any other manner, we may immediately suspend or terminate your Amazon privileges, remove reviews, and delist related products.”³

Die Zeitschrift Ökotest berichtete in ihrer Online-Ausgabe vom 25. Januar 2013 von kriminellen Betrügern, die in einem Online-Portal ein Produkt angeboten, positiv bewertet, dann verkauft und nie ausgeliefert haben.⁴ Opinion Spam war hier essenzieller Aspekt der kriminellen Machenschaften.

Wir haben es hier also mit einem gesellschaftlich und ökonomisch erheblichen Problem zu tun. Bewertungsportale wie HolidayCheck betreiben einen erheblichen Aufwand, um gefälschte Bewertungen zu löschen (<https://www.holidaycheck.de/glaubwuerdigkeit-hotelbewertungen>). Es stellt sich daher die Frage, wie Opinion Spam erkannt werden kann und ob es möglich ist, den Erkennungsprozess durch automatische Methoden zu unterstützen.

9.1 Gefälschte Bewertungen

Im ersten Schritt sehen wir uns an, worum es bei diesem Phänomen eigentlich geht.

Liu (2012) unterscheidet zunächst Hype-Spam, bei dem ein Produkt oder eine Firma hochgelobt wird, von Defaming Spam, bei dem verunglimpft wird. Außerdem unterscheidet er “Fake Reviews”, bei denen es um einzelne Produkte oder Dienstleistungen geht, von unspezifischen Bewertungen von ganzen Firmen oder Dienstleistern (“Ich liebe XYZ!”).

Wer sind nun die Personen, die gefälschte Bewertungen verfassen? Erwähnt wurden schon fragwürdige Dienstleister oder Unternehmen mit kriminellem Hintergrund. Diese Fälle sind eindeutig. Unklarer ist jedoch die Einschätzung, wenn Freunde und Familie eines Buchautors dessen neues Buch bei Amazon bewerten oder die Bewertung eines neuen Autos durch die Mitarbeiter des Herstellers. Was ist darüber hinaus davon zu halten, wenn eine Firma einen Preis dafür ausschreibt, dass Kunden Bewertungen schreiben? Es wird deutlich, dass es hier keine klaren Grenzen gibt und sich die Erkennung von Opinion Spam daher zunächst auf die klaren Fälle beschränken muss.

Liu (2012) stellt – wie im Kap. 2 beschrieben – Quintupel auf, um Bewertungen zu beschreiben. Diese Quintupel enthalten die relevanten Elemente einer Bewertung: Die Enti-

²https://www.amazon.com/gp/help/customer/display.html?nodeId=201749630&ref=cm_udrp_bar.

³dt.: Wenn wir feststellen, dass Sie versucht haben, Bewertungen zu manipulieren oder auf andere Weise gegen unsere Richtlinien verstoßen haben, können wir Ihre Amazon-Privilegien sofort aussetzen oder beenden, Bewertungen entfernen und verwandte Produkte aus der Liste nehmen. (eigene Übersetzung).

⁴<http://www.oekotest.de/cgi/index.cgi?artnr=11617;gartnr=91;bernr=23>.

tät, die bewertet wird, der bewertete Aspekt dieser Entität, die Meinung dazu, der Reviewer und der Zeitpunkt des Reviews. Auf diese Informationen wird auch zugegriffen, wenn man versucht, Fälschungen zu entdecken. (Liu 2012) beschreibt die Daten, die für die Entdeckung von Fälschungen zur Verfügung stehen, auf drei Ebenen: Textebene, Meta-Daten und Produktinformationen. Auf der Textebene gibt es Informationen über die Inhalte (wenn Bewertungen unterschiedlicher Produkte sehr ähnlich sind oder etwa die Bewertung eines Buchs einfach aus dem Klappentext kopiert wurde), über die Professionalität (Satzlänge, Fehleranteile), über die benutzten Wörter und über syntaktische und semantische Hinweise auf Lügen. Auf der Ebene der Meta-Daten gibt es Information über die vergebenen Sterne (z. B. die 5-Sterne-Bewertung von Amazon), über die User-ID in einem Bewertungsportal, über den Zeitpunkt des Postings, IP- und MAC-Adressen, Ort des Computers, von dem die Bewertung kommt, und die Reihenfolge von Klicks. Auf der Produktebene haben wir Informationen über den Verkaufsrang und die Produkteigenschaften.

9.2 Annotierte Korpora für Opinion Spam

Ein wesentliches Problem für die automatische Erkennung von Opinion Spam: Es steht kein solide validiertes Korpus von gefälschten Bewertungen für die deutsche Sprache zur Verfügung, mit dem man z. B. Data Mining-Systeme trainieren oder testen könnte. Das erste Korpus für das Englische wurde 2011 entwickelt und in (Ott et al. 2011) beschrieben. Es enthält 400 echte und 400 gefälschte Bewertungen. (Ott et al. 2011) sind so vorgegangen, dass sie positive Hotel-Bewertungen mit fünf Sternen für die 20 populärsten Hotels in der Gegend von Chicago als echte Bewertungen aus TripAdvisor genommen und mit Amazon Mechanical Turk Versuchspersonen beauftragt haben, gefälschte Bewertungen für diese Hotels zu erstellen.

Wang et al. (2012) erstellten ein Korpus für das Englische, indem sie Testpersonen aufforderten, gefälschte Bewertungen für Produkte, die sie nicht kennen, zu produzieren. Das Problem bei dieser Vorgehensweise mit Versuchspersonen ist, dass man nicht sicher sein kann, ob die Ergebnisse authentisch sind.

Li et al. (2015) berichten über ein Korpus für das Chinesische mit 6 Mio. Bewertungen für Restaurants in Shanghai, welches auch Metadaten, z. B. Zeit des Postings, IP-Adressen und Orte der Computer enthält. Die chinesische Firma Dianping benutzt einen eigenen Filter für gefälschte Bewertungen, dessen Algorithmus jedoch Firmengeheimnis ist. Die Information darüber, welche Bewertungen gefiltert worden sind, stand den Wissenschaftlern jedoch zur Verfügung. Sie klassifizierten Reviewer (und ihre IP-Adressen), von deren Bewertungen mehr als 50% von Dianping als gefälscht eingestuft werden, als nicht vertrauenswürdige Reviewer.

Sandulescu und Ester (2015) hatten ebenfalls Zugriff auf Bewertungen, die von Portalen automatisch gefiltert wurden und damit auf ein Korpus von Bewertungen mit Spam-Verdacht. Unklar sind jedoch Precision und Recall der Filtermethoden, die zu diesem Korpus geführt

haben. Das Portal Yelp stellt die gefilterten Beiträge auf den Bewertungsseiten zur Verfügung. Es ist jedoch auf den ersten Blick nicht ersichtlich, auf welcher Grundlage diese Beiträge gefiltert worden sind. Sehr wahrscheinlich basieren diese Filter eher auf einer Kategorisierung des Verhaltens der Reviewer als auf dem Text der Bewertung. Als Gold-Standard für eine Evaluation oder für Machine Learning-Verfahren eignen sich die deutschen Beispiele jedenfalls nicht.⁵

Shojaee et al. (2015) erstellten ein Korpus, indem sie Annotatoren nach gefälschten Bewertungen in einem Online-Portal suchen lassen und dann das Inter-Annotator Agreement (also die Übereinstimmung zwischen den Annotatoren) gemessen haben. Dies ist sicher eine recht zuverlässige Methode, auch wenn Fake-Bewertungen nicht leicht zu erkennen sind. Die Autoren unterstützten den Annotationsprozess dadurch, dass sie den Annotatoren gezielte Fragen stellten⁶ und alle Bewertungen eines Reviewers gleichzeitig präsentierten.

Mit Studierenden der Hochschule Darmstadt haben wir Beispiele für Opinion Spam im deutschsprachigen Amazon-Portal gesucht und diese dann analysiert. Dafür haben wir zunächst Kriterien für die Klassifikation der Bewertungen als gefälscht aufgestellt. Sehr häufig verwendeten Reviewer identische Texte für unterschiedliche Produkte. Dabei waren die Texte nichtssagend, also allgemein verwendbar. Auch das Datum ist ein wichtiges Kriterium: Wenn ein Reviewer eine große Anzahl an z. B. Baugeräten einer Firma an einem Tag positiv und die Baugeräte einer anderen Firma am selben Tag negativ bewertet, ist das ein starker Hinweis darauf, dass es sich um eine Fälschung handeln kann. So haben vier Studierende gefälschte Bewertungen im Amazon-Portal gefunden und die Kriterien für die Klassifikation im Textkorpus mit angegeben, sodass die Entscheidung nachvollziehbar ist.

Das Annotationsschema für das Korpus deutscher Opinion Spam beinhaltet die Ebenen, die auch (Liu 2012) beschrieben hat: Textebene, Meta-Daten und Produktinformationen.

- Textebene
 - Überschrift
 - Text
 - Datum
 - Tonalität
- Meta-Daten
 - Review-URL
 - Rating
 - Anzahl der Bewertungen
 - Anzahl der Bewertungen mit 5 Sternen
 - Anzahl der Bewertungen mit 1 Stern
 - diese Bewertung

⁵Z.B. <https://www.yelp.de/biz/die-kaffee-d%C3%BCsseldorf>.

⁶Z.B.: „Is this review unrelated to the product?“.

- Reviewer
 - User-ID
 - verifizierter Kauf
 - Bewertung des Reviews als hilfreich
- Produktinformation
 - Produktname
 - Verkaufsrank
- Shop
- Begründung für die Klassifikation

Das so entstandene Textkorpus mit 218 als gefälscht klassifizierten und 60 als echt klassifizierten Bewertungen ist auf der Webseite zu diesem Buch verfügbar. Es ist für maschinelles Lernen noch zu klein, aber schon groß genug, um einige statistische Untersuchungen zu machen und heuristische Methoden für eine Klassifikation und daraus Features für Machine-Learning-Methoden zu entwickeln.

9.3 Klassifikation von Bewertungen

Die Erkennung von Opinion Spam ist eine klassische Klassifikationsaufgabe, die Dokumente (Bewertungen) als gefälscht oder als nicht gefälscht klassifizieren soll. Im Folgenden werden Methoden für diese Klassifikationsaufgabe vorgestellt. Diese Methoden unterscheiden sich vor allem dadurch, auf welche Daten sie sich beziehen.

9.3.1 Klassifikation mit Meta-Daten

Hooi et al. (2016) klassifizieren Reviewer auf der Basis ihres Verhaltens. So sind beispielsweise Reviewer, die ausschließlich positive Bewertungen in großer Menge abgeben, verdächtig, ebenso wie Reviewer, die viele Bewertungen in einer sehr kurzen Zeit abgeben. Mit diesen Meta-Daten trainieren sie ein Bayesianisches Modell des Data Mining, um verdächtige Reviewer zu finden.

Wang et al. (2012) betrachten die Relation zwischen dem Reviewer, den Bewertungen und dem Shop, der das Produkt anbietet. Sie teilen also die Klassifikationsaufgabe in drei Unteraufgaben ein. Sie stellen fest, dass nicht vertrauenswürdige Reviewer (Spammer) eine Beziehung zum Shop haben. Weiterhin stellen sie fest, dass es gute und schlechte Shops gibt und dass schlechte Shops Spammer engagieren. Fake-Bewertungen sind nicht ehrlich. Für die Klassifikation der Bewertungen betrachtet man also die Korrelationen der Vertrauenswürdigkeit der Reviewer, der Echtheit der Bewertungen und der Seriosität des Shops:

- Die Vertrauenswürdigkeit des Reviewers ist abhängig von der Anzahl seiner echten Bewertungen.
- Ein Shop ist seriös, wenn viele vertrauenswürdige Reviewer ihn positiv bewerten, und weniger seriös, wenn viele vertrauenswürdige Reviewer ihn negativ bewerten.
- Die Echtheit einer Bewertung hängt von der Seriosität des Shops und der Übereinstimmung mit anderen Bewertungen in einem gegebenen Zeitfenster ab.

Als gravierendes Problem für die Evaluation dieser Methode wird auch von (Wang et al. 2012) hervorgehoben, dass es kein ausreichend großes Korpus mit Bewertungen gibt, bei denen zuverlässig annotiert ist, ob sie echt oder falsch sind.

Mit der Datenbasis, die (Li et al. 2015) mit ihrem chinesischem Korpus haben, konnten sie Regelmäßigkeiten bzgl. Zeit und Ort feststellen. So sind Spammer häufiger an Wochentagen aktiv als vertrauenswürdige Reviewer, die eher am Wochenende Restaurants bewerten. Je weiter entfernt die Reviewer von Shanghai waren, desto mehr Spammer waren unter ihnen. (Li et al. 2015) zeigen so die Möglichkeit auf, die Klassifikation von Reviewern auf der Basis von vorklassifizierten Daten zu lernen. Verknüpft mit den Methoden von (Wang et al. 2012) könnte man mit diesen Ergebnissen Shops und Bewertungen klassifizieren.

Die Zeit des Postings von Bewertungen betrachten (Ye et al. 2016) unter verschiedenen Aspekten, die sie als Alarmsignale interpretieren, wie:

- wenn die durchschnittliche Bewertung eines Produkts sich plötzlich ändert
- wenn plötzlich extrem viele Bewertungen zu einem Produkt erscheinen
- wenn Reviewer regelmäßig jeden Tag Bewertungen posten

Banerjee et al. (2015) weisen darauf hin, dass Fake-Bewertungen oft ohne den vorherigen Kauf eines Produkts oder den Aufenthalt in einem bewerteten Hotel stattfinden. Auch Amazon markiert Bewertungen mit „verifizierter Kauf“, wenn ein Produkt über Amazon bestellt worden ist. Hier sind also Metadaten vorhanden, die helfen können, Bewertungen als echt zu klassifizieren und ein Korpus aus echten Bewertungen aufzubauen.

9.3.2 Klassifikation mit linguistischer Information

Sandulescu und Ester (2015) stellen fest, dass Spammer zwar häufig ihre Benutzernamen wechseln, aber dennoch häufig sehr ähnliche Texte für unterschiedliche Produkte schreiben. Die Autoren berechnen daher die Ähnlichkeit zwischen Bewertungen unter Verwendung von Synonymie-Beziehungen aus WordNet sowie Informationen über Lemmatisierung von Wörtern und ihren syntaktischen Kategorien. Sie bestimmen dann in Experimenten einen Grenzwert, ab dem die Bewertungen anderen so ähnlich sind, dass sie als gefälscht klassifiziert werden können. Als Datenbasis für die Evaluation nehmen sie Bewertungen aus Yelp und Trustpilot, wobei sie die von den Firmen gefilterte Bewertungen als gefälscht klassifi-

zieren. Damit knüpfen sie an Experimente von (Jindal und Liu 2008) an, die ebenfalls nach Bewertungen suchen, die ein hohes Maß an Ähnlichkeit aufweisen und damit ein Korpus von Fake Reviews aufbauen.

Banerjee et al. (2015) erstellen ein Korpus gefälschter Bewertungen mithilfe von Versuchspersonen. Dieses Korpus nutzen sie, um linguistische Hinweise in Bewertungen zu sammeln, die auf Fakes hindeuten. Sie unterscheiden dabei vier linguistische Merkmale: Verständlichkeit, Detailgenauigkeit, Schreibstil und Kognitionsindikatoren. Für die Bestimmung der Verständlichkeit wurden Standard-Verständlichkeitsmaße wie der “Flesch-Kincaid Grade Level” (Kincaid et al. 1975) berechnet. Detailgenauigkeit wird zunächst mit den verwendeten syntaktischen Kategorien (POS) berechnet. In informativen Texten gibt es mehr Nomen, Adjektive, Artikel und Präpositionen als Verben, Konjunktionen, Adverbien und Pronomen. Dazu kommt die Berechnung der lexikalischen Diversität. Die Berechnung des Schreibstils beruht auf der Benutzung von Emotionswörtern, dem Tempus der Verben, dem Gebrauch von verstärkenden Wörtern wie “always” oder “never” und Satzzeichen wie Fragezeichen oder Ausrufezeichen. Kognitionsindikatoren sind sprachliche Merkmale, die auf Lügen hindeuten, wie z.B. der Gebrauch von Wörtern wie “should” und “may” oder auch der Gebrauch von Füllwörtern. Mithilfe dieser Merkmale trainieren sie Machine-Learning-Systeme. Im Ergebnis zeigt sich, dass die linguistischen Merkmale dabei helfen, echte von gefälschten Reviews zu unterscheiden. Kritisch dabei ist jedoch, dass das Korpus der gefälschten Bewertungen in einer künstlichen Experiment-Situation entstanden ist und daher seine Authentizität in Frage gestellt werden kann.

9.4 Beobachtungen über Opinion Spam im deutschsprachigen Amazon-Portal

Sehen wir uns einmal die Beispiele für Opinion Spam an, die die Studierenden der Hochschule Darmstadt im deutschsprachigen Amazon-Portal gefunden haben. Diese Beispiele sind auf der zum Buch gehörenden Webseite verfügbar. Erste Beobachtungen zeigen, dass die Forschungsergebnisse für das Englische und das Chinesische zum Teil auf das Deutsche übertragbar sind.

Wir haben uns auf Hype-Spam und Fake Reviews konzentriert, weil wir vor allem diese im deutschen Amazon-Portal gefunden haben. Wenige Beispiele für Defaming Spam sind aber ebenfalls dabei. Anders als von (Wang et al. 2012) beobachtet, scheint im deutschen Amazon-Portal aus dem Jahr 2016 der Shop nicht ausschlaggebend zu sein. Wenn wir eine gefälschte Bewertung gefunden haben und weitere Bewertungen zu Produkten im selben Shop analysiert haben, so haben wir nur sehr selten weitere gefälschte Bewertungen gefunden. Es müsste somit untersucht werden, ob eher die Herstellerfirma (z.B. im Fall von technischen Geräten) oder der Autor, Komponist oder ähnliche Protagonisten Opinion Spam in Auftrag geben.

Wie auch (Hooi et al. 2016) feststellt, haben wir häufig verdächtige Reviewer gefunden, die denselben Text am selben Datum für verschiedene Produkte verwenden. Dies ist auch ein Ansatzpunkt für eine Erweiterung des Korpus, denn weitere Bewertungen von notorischen Spammern können aufgenommen werden.

Das Datum scheint eine Rolle zu spielen, etwa wenn es direkt nach Erscheinen einer CD sehr viele positive Bewertungen innerhalb weniger Tage gibt und später dann in erster Linie negative. Wir können – ebenso wie (Li et al. 2015) – feststellen, dass die Spammer meist an Wochentagen und seltener an Wochenenden agieren. Es sind nur ca. ein Viertel der gefälschten Bewertungen am Wochenende entstanden und der Rest an einem Wochentag. Allerdings sind auch ca. 70 % der echten Bewertungen an einem Wochentag entstanden.

Anders als bei (Banerjee et al. 2015) festgestellt, handelt es sich bei den gefälschten Bewertungen im deutschen Amazon-Portal oft um verifizierten Kauf, im Korpus in ca. 82 % der Fälle. Dies deutet auf eine gewisse Professionalität der Spammer hin, die entweder direkt von den Shops beauftragt werden oder die Produkte bestellen und danach zurücksenden. Jedenfalls scheint für das deutsche Amazon-Portal des Jahres 2016 die Methode des Korpusaufbaus mit nicht verifizierten Käufen nicht zu funktionieren.

Die verdächtigen Texte – gerade wenn sie von Spammern mehrfach verwendet werden – sind wenig konkret, z. B.:

Alles bestens und schnell wie immer gelaufen - würde ich immer wieder wiederholen. Die Ware ist OK

also die lieferung ist schnell und unkompliziert. die ware ist top und es gibt keine beanstandungen. da würde ich wieder bestellen. :-)

Häufig beziehen sich die Spammer auf die Lieferung, wie im oben genannten Beispiel, und nicht auf das Produkt selbst, da sie dann für jedes Produkt eine eigene Bewertung schreiben müssen. Manche versuchen jedoch, auch diesen Prozess zu automatisieren, was im folgenden Fall schiefgegangen ist, weil die Variablen im Text geblieben sind:

Ich kann das oben angegebene Produkt \$article_name vorbehaltlos empfehlen. Als ich \$article_medium endlich erwerben konnte, war ich mehr als positiv überrascht. Ich werde auch in Zukunft \$article_name immer wieder konsumieren und habe gleich noch einmal zugegriffen, da auch der Preis \$article_price für das Produkt \$article_name sehr gut ist. Ich freue mich schon auf weitere sehr gute Angebote von \$article_manufacturer.

Die durchschnittliche Anzahl an Nomen ist in echten Bewertungen etwa doppelt so hoch wie in gefälschten. Autoren von Bewertungen nutzen Nomen, um konkrete Informationen über Aspekte des Produkts zu geben, das sie bewerten. Diese Texte können dann nicht mehr für andere Produkte weiterverwendet werden. Viele der häufig in gefälschten Bewertungen genutzten Nomen sind eher unspezifisch, so wie “Produkt” oder “Preis”. Die lexikalische Diversität ist daher in gefälschten Bewertungen geringer als in echten.

Tab.9.1 Durchschnittliche Werte für Metadaten-Merkmale in gefälschten und echten Bewertungen

	FAKE	GENUINE
salesrank	142,660	418,591
no_ratings	83.51	82.10
thisrating	4.60	3.32
no_five_star_ratings	51.98	48.85
no_one_star_ratings	9.44	13.70
verified	0.82	0.75
review_date_weekend	0.24	0.30

Gefälschte Texte sind im Durchschnitt kürzer als echte Bewertungen (38 Tokens pro gefälschter Bewertung vs. 71 Tokens pro echter Bewertung). Viele Spammer reagieren auf die Anforderungen von Amazon nach einer Mindestlänge eines Reviews von 20 Wörtern mit Tricks, wie sinnlose Sätze, Wiederholungen und Wörtern mit Leerzeichen zwischen den Buchstaben:

alles war gut, ich habe leider keine weitere Lust noch mehr dazu zu schreiben mit recht freundlichen grüßen danke !!!

gefällt mir, sieht gut aus, ist sehr praktisch,einfach gut,gefällt mir, sieht gut aus, ist sehr praktisch,einfach gut,gefällt mir, sieht gut aus, ist sehr praktisch,einfach gut

Hab den Anhänger damals für ne Freundin bestellt - hat Ihr gefallen - empfehlenswert

Die durchschnittliche Anzahl der Tokens, die nur aus einem Zeichen bestehen, ist in gefälschten Bewertungen deutlich höher.

Tab.9.1 zeigt die Verteilung der Metadaten-Merkmale im annotierten Textkorpus. Man sieht, dass der Salesrank bei echten Bewertungen signifikant höher ist. Anders als erwartet haben gefälschte Bewertungen häufiger den Status “verified” als echte Bewertungen.

Tab.9.2 zeigt die Verteilung der linguistischen Merkmale im annotierten Textkorpus. Gefälschte Bewertungen sind tendenziell kürzer: Die Anzahl der Tokens sowohl in den

Tab.9.2 Durchschnittliche Werte für linguistische Merkmale in gefälschten und echten Bewertungen

	FAKE	GENUINE
no_tokens_headline	2.63	3.90
no_tokens_text	37.56	70.82
one_letter_tokens	0.21	0.17
no_nouns	0.17	0.2
no_adjectives	0.1	0.1
no_verbs	0.13	0.12
no_selfreferring_pronouns	1	1.05

Überschriften als auch im Text ist bei echten Bewertungen signifikant höher. Die Anzahl der Tokens, die nur aus einem Zeichen bestehen, ist bei gefälschten Bewertungen höher.

9.5 Maschinelles Lernen für die automatische Klassifikation

Eine automatische Klassifikation als gefälschte Bewertung erscheint mit maschinellen Lernverfahren also insgesamt als möglich. Wir kennen die Merkmale von Bewertungen, die für die Klassifikation relevant sein können. Mit diesen Merkmalen trainieren wir jetzt auf unseren Daten ein Modell. Jede der Bewertungen im Korpus wird als Vektor mit folgenden Informationen dargestellt:

- salesrank (Zahlenwert)
- number_of_ratings_for_this_product (Zahlenwert)
- this_rating (Zahlenwert 1–5)
- number_of_five_star_ratings_for_this_product (Zahlenwert)
- number_of_one_star_ratings_for_this_product (Zahlenwert)
- verified (1 oder 0)
- review_date_weekend (1 oder 0)
- number_of_tokens_headline (Zahlenwert)
- number_of_tokens_text (Zahlenwert)
- selfreferring_pronouns (Prozentanteil aller Tokens)
- nouns (Prozentanteil aller Tokens)
- adjectives (Prozentanteil aller Tokens)
- verbs (Prozentanteil aller Tokens)
- one_letter_tokens (Prozentanteil aller Tokens)
- fake_words (Zahlenwert)
- class (echt oder gefälscht)

Die Meta-Daten (salesrank, number_of_ratings_for_this_product, this_rating, number_of_five_star_ratings_for_this_product, number_of_one_star_ratings_for_this_product) können dabei direkt aus dem XML extrahiert werden, während die linguistischen Daten mithilfe einer linguistischen Analyse, z. B. mit TextBlob oder spaCy, erfasst werden. Um eine Liste von “Fake Words” aufzustellen, also Wörtern, die vor allem in gefälschten Bewertungen vorkommen, machen wir eine Differenzanalyse wie auch schon bei den Sentimentwörtern: Wir vergleichen die Wörter in den gefälschten Bewertungen mit denen in den echten Bewertungen und nehmen in die Liste diejenigen Wörter auf, die nur in gefälschten Bewertungen vorkommen. Diese Liste wird dann manuell nachbearbeitet.

Reviews sind dann z. B. so dargestellt:

2919,73,5,71,2,1,0,1,20,0,0.25,0,0,0.1,0,0.05,FAKE

21,514,5,331,64,1,0,2,27,0,0.37,0.07,0.04,0,0,0,GENUINE

Diese Vektoren können nun als Trainingsmaterial für maschinelles Lernen verwendet werden. Wie im Abschn. 3.6.2 beschrieben, arbeiten wir hier mit Pandas für die Datenanalyse und Sklearn für das maschinelle Lernen.

9.6 Zusammenfassung

Sentiment-Analyse verliert ihren Wert in dem Maße, in dem gefälschte Bewertungen – Opinion Spam – in den Foren auftritt. Die Nutzer wie die Portale leiden erheblich unter dem Verlust an Glaubwürdigkeit der Bewertungen. Daher müssen Algorithmen entwickelt werden, die diese Fälschungen entdecken und klassifizieren. Wir haben zunächst definiert, welche Arten von Fälschungen es gibt. Da es für eine automatische Klassifikation unablässig ist, auf annotierte Textkorpora zuzugreifen, haben wir im nächsten Schritt annotierte Korpora dargestellt und sind näher auf ein kleines Textkorpus deutschsprachiger Opinion Spam eingegangen. Anhand der Forschungsliteratur haben wir verschiedene Klassifikationsmethoden anhand von Meta-Daten und anhand von linguistischer Information vorgestellt. Mit einer Analyse des annotierten Textkorpus haben wir Merkmale identifiziert, die für eine Klassifikation relevant sind. Mit diesen Merkmalen können wir anschließend mit maschinellem Lernen ein Modell trainieren.

9.7 Übungen

1. Prüfen Sie Ihr Wissen:

- Welche Informationen stehen für die Erkennung von Opinion Spam zur Verfügung?
- Warum ist das Fehlen eines zuverlässigen annotierten Textkorpus für Opinion Spam ein Problem?
- Welche Methoden zur Erkennung gibt es?

2. Setzen Sie Ihr neues Wissen ein:

- a) Suchen Sie im Amazon-Portal manuell nach gefälschten Bewertungen und verifizieren Sie dabei die im Text genannten Kriterien.
- b) Nehmen Sie den annotierten Textkorpus von der Buch-Website zur Hand und konvertieren Sie die Bewertungen in Vektoren, wie im Text beschrieben. Trainieren Sie anschließend ein Modell mit Methoden des maschinellen Lernens darauf.
- c) Evaluieren Sie dieses Modell an den von Ihnen selbst gesammelten Bewertungen.

3. Reflexion in Gruppenarbeit:

- a) Diskutieren Sie in der Gruppe, für wen Opinion Spam ein Problem sein kann (Öffentlichkeit/Online-Nutzer, Portal-Betreiber, Unternehmen).
- b) Diskutieren Sie in der Gruppe, ob die Manipulation von Bewertungen ein “legitimes” Instrument der Verkaufsförderung ist. Betrachten Sie in diesem Zusammenhang die Möglichkeit von “incentivized” Bewertungen, also den Einsatz von Anreizsystemen zur Review-Abgabe, z.B. Bonus-Meilen, monetäre Vergütung, Urlaubsreise, kostenfreie Produkte.

9.8 Weiterführende Literatur

Einen aktuellen Überblick über Forschungsmethoden geben (Ren und Ji [2019](#)). Dort werden auch Arbeiten vorgestellt, die neuronale Netze nutzen. Das Problem der Korpuserstellung wird von (Ott et al. [2011](#); Wang et al. [2012](#); Li et al. [2015](#); Sandulescu und Ester [2015](#); Shojaei et al. [2015](#)) adressiert. Beispiele für Klassifikationen mit Meta-Daten sind in (Hooi et al. [2016](#); Wang et al. [2012](#); Li et al. [2015](#)) zu finden. Eine Klassifikation mit linguistischen Merkmalen beschreiben (Sandulescu und Ester [2015](#); Banerjee et al. [2015](#)).

Erkennung und Klassifikation von Aggression in Meinungsäußerungen

10

Mit dem Aufkommen der sozialen Netzwerke entstand eine Problematik, die sich in letzter Zeit deutlich verstärkt hat: Aggressive, hasserfüllte, beleidigende Postings, bis hin zu Bedrohungen von Politikern, Journalistinnen und anderen Menschen, die ihre Meinung äußern. Es bleibt nicht immer bei verbaler Aggression, in manchen Fällen folgten darauf Verbrechen, wie 2019 im Fall des Kasseler Regierungspräsidenten Walter Lübcke¹. Aggression in Meinungsäußerungen umfasst das weite Spektrum vom Gebrauch von Schimpfwörtern über Beleidigungen und Diskriminierungen bis hin zu Gewaltandrohungen (Ruppenhofer et al. 2018b). Die sozialen Medien wie Twitter, Facebook und auch die Kommentarspalten der Online-Präsenzen von Zeitungen und Radiosendern werden zunehmend von Menschen dominiert, die diffamieren, beleidigen und bedrohen. In einer Studie der Landesanstalt für Medien NRW (Forsa 2018) wurde festgestellt:

So gibt die überwiegende Mehrheit der Befragten (78 %) an, schon einmal Hassrede bzw. Hasskommentare im Internet gesehen zu haben, z. B. auf Webseiten, in Blogs, in sozialen Netzwerken oder in Internetforen. Davon geben 10 Prozent an, schon sehr häufig Hassrede bzw. Hasskommentare im Internet gesehen, 26 Prozent häufig und 42 Prozent weniger häufig.

Automatisch generierte Nachrichten werden verwendet, um den Eindruck zu erwecken, dass diese extremen Meinungen in der Bevölkerung weit verbreitet sind, aber auch, um politische Gegner mundtot zu machen. Infolgedessen gelingt es vielen Betreibern von Social-Media-Webseiten nicht mehr, Nutzerbeiträge manuell zu moderieren, und die Moderation hasserfüllter Kommentarspalten bedeutet für die Moderatoren eine enorme psychische Belastung. Betreiber von sozialen Netzwerken werden verpflichtet, strafbare Inhalte nicht nur zu filtern sondern auch zu melden. Daher besteht ein dringender Bedarf an Methoden zur automatischen Identifizierung verdächtiger Beiträge.

¹https://de.wikipedia.org/wiki/Mordfall_Walter_L%C3%BCbcke.

Im ersten Schritt ist es aber notwendig zu definieren, was eigentlich als verdächtige Beiträge klassifiziert werden soll. Im Rahmen des GermEval Projekts 2018 und 2019² wurde ein Dokument mit “Annotation Guidelines” erstellt, das sehr spezifische Richtlinien für diese Klassifikation enthält (Ruppenhofer et al. 2018a, eigene Übersetzung). Die Hauptkategorien sind³:

- **INSULT**: Zuschreibung von negativ bewerteten Qualitäten oder Mängeln oder die Kennzeichnung von Personen als unwürdig oder nicht wertvoll. Beleidigungen vermitteln Respektlosigkeit und Verachtung.
 - z. B.: ein #Tatort mit der Presswurst #Saalfeld geht gar nicht #ARD
- **ABUSE**: Das Ziel der Bewertung wird als Repräsentant einer Gruppe angesehen. Dieser Gruppe werden negative Eigenschaften zugewiesen, die universal, omnipräsent und unveränderbar sind. Weitere Formen sind die Entmenschlichung eines Individuums und Bedrohungen.
 - z. B.: Ich persönlich scheisse auf die grüne Kinderfickerpartei
- **PROFANITY**: Nicht akzeptierbare Sprache kann auch ohne Beleidigung oder Diskriminierung auftreten. Es handelt sich hierbei um Schimpfwörter und Flüche.
 - z. B.: ob ich sterbe darauf geb ich fick
- **OTHER**: Alle Äußerungen, die positiv oder neutral sind oder negative Äußerungen, die nicht unter die Kategorien INSULT, ABUSE oder PROFANITY fallen. Ironische beleidigende Äußerungen, die oberflächlich betrachtet positiv sind, fallen nicht unter diese Kategorie.
 - z. B.: Hamas-Vertreter bekräftigt Ziel der Zerstörung Israels

Ein Hauptmerkmal von ABUSE ist, dass die dem Ziel zugeschriebenen negativen Qualitäten direkt aus seiner Zugehörigkeit zu einer Gruppe oder einem Kollektiv folgen, von den anderen Mitgliedern der Gruppe geteilt werden und unveränderlich sind. Die prototypischen Klassen sind also diejenigen, die durch die Geburt definiert sind, wie Geschlecht, Nationalität, Ethnizität, Glaube. Es gibt jedoch auch andere Klassen, die, obwohl sie nicht durch die Geburt definiert sind, Ziele von Diskriminierung sein können, zum Beispiel Migranten/Flüchtlinge oder politische Parteien.

Gruppen, die häufig Ziele von ABUSE sind:

- Feministinnen
- Menschen mit schwarzer Hautfarbe
- Muslime
- Juden
- Homosexuelle (LGBT)

²<https://projects.fzai.h-da.de/iggsa/>.

³Es lässt sich nicht ganz vermeiden, dass wir einige sehr hässliche Beispiele hier geben. Diese spiegeln natürlich nicht unsere Meinung wider.

- Flüchtlinge
- Mitglieder politischer Parteien
- Journalisten

Den Gruppen werden häufig vermeintliche Eigenschaften – Stereotype wie Faulheit, Unsauberkeit, sexuelle Perversion – zugeschrieben. Personen oder Gruppen von Personen werden verflucht (*Die soll der Teufel holen!*) oder bedroht (*Den mach ich fertig!*). In einigen Fällen wird zur Gewalt aufgerufen (*Haut den ... in die Fresse!*).

Es bleibt – trotz umfassender Definitionen – letztlich unklar, welche Merkmale ABUSE-Sprache im Detail notwendigerweise haben muss, um als solche klassifiziert zu werden. Muss eine Person oder Personengruppe direkt angesprochen werden oder reicht es, wenn indirekte Formen der Bezugnahme auf Personen und Gruppen vorliegen (z. B. eine Erwähnung von *Negermusik* im Bezug auf eine Fernsehsendung)? Diese Fragen lassen sich nicht objektiv und aus rein sprachlicher Analyse entscheiden. Welche Art der sprachlichen Äußerung als Herabsetzung gilt, welche Zuschreibung zu einer marginalisierten sozialen Identität führt ist, muss letztlich gesellschaftlich ausgehandelt werden. Eine Software lernt aus den Klassifikationen, die von Menschen vorgegeben wurden. Daher ist es erstens wichtig, dass die zugrundeliegenden annotierten Daten sorgfältig erstellt wurden und zweitens, dass das Ergebnis der Software immer auch durch Menschen evaluiert wird, dass also nicht ein automatischer Filter, sondern nur ein Warnsystem implementiert wird.

10.1 Daten, Daten, Daten

Wie wir schon gesehen haben, ist eine wichtige Voraussetzung für die automatische Klassifikation von Texten die Verfügbarkeit von annotierten Daten. Um die Forschung und Entwicklung in einem Themengebiet voranzubringen, wird im Bereich der Sprachverarbeitung daher häufig zunächst ein Wettbewerb aufgesetzt, eine sogenannte “Shared Task”. Wir haben ja bereits mit den Daten der GermEval Shared Task 2017 zu Kundenkommentaren der deutschen Bahn gearbeitet. Für die automatische Klassifikation aggressiver Meinungsäußerungen gab es in den letzten Jahren einige Shared Tasks:

- Kaggle’s 2018 Toxic Comment Classification Challenge⁴ war eine Shared Task, die sich mit Kommentaren im englischsprachigen Wikipedia beschäftigte. Es gab sechs Kategorien für die Klassifikation: “toxic, severe toxic, obscene, insult, identity hate, threat”.
- Bei der TRAC Shared Task on Aggression Identification (Kumar et al. 2018) ging es um Facebook-Kommentare in Englisch und Hindi. Die teilnehmenden Forschungsgruppen mussten diskriminierende Kommentare entdecken und offene und versteckte Aggression klassifizieren.

⁴<https://www.kaggle.com/c/jigsawtoxic-comment-classification-challenge>.

- Die Shared Task on Automatic Misogyny Identification (AMI) (Fersini et al., 2018) betraf englische, spanische und italienische Tweets. Der Fokus lag hier auf Tweets mit sexistischem Inhalt. Neben der Entdeckung sexistischer Tweets wurden diese auch klassifiziert. Die Klassen waren “Discredit, Derailing, Dominance, Sexual Harassment & Threats of Violence, Stereotype & Objectification, Active, Passive”.
- Bei SemEval 2019 – Task 5 (HatEval) (Basile et al. 2019) ging es um die automatische Erkennung von englisch- und spanischsprachiger Hassrede gegen Immigranten auf Twitter. In zwei Subtasks wurden die Tweets einmal als Hate Speech oder nicht klassifiziert und einmal wurde das Ziel des Angriffs als individuell oder generisch klassifiziert, was grob der Unterscheidung zwischen “Insult” und “Abuse” entspricht. Bei SemEval 2019 – Task 6 (OffenseEval 2019) Zampieri et al. 2019) ging es um englischsprachige Tweets. Das Datenset bestand aus 14.000 Tweets. Aus diesen sollten die offensiven Tweets extrahiert und klassifiziert werden. Außerdem sollte erkannt werden, wer oder was dort angegriffen wurde. Auch 2020 wurde die OffenseEval als SemEval 2020 – Task 12 fortgeführt. Hier ging es um ein mehrsprachiges Datenset mit arabischen, dänischen, englischen, griechischen und türkischen Daten (<https://sites.google.com/site/offensevalsharedtask/>).
- GermEval 2018 (Wiegand et al. 2018b) beschäftigte sich mit der Klassifikation deutschsprachiger Twitterdaten als “Insult”, “Abuse”, “Profanity” oder eben nicht offensiv, “Other”. In einer zweiten Auflage, bei GermEval 2019 (Struß et al. 2019) wurden zusätzlich offensive Tweets als explizit und implizit klassifiziert. Insgesamt besteht das Datenset aus über 15.000 annotierten deutschsprachigen Tweets. Diese Daten sind auf der Projektwebseite⁵ verfügbar. Wir können sie nutzen, um ein Klassifikationssystem für Tweets zu implementieren.

Neben einer möglichst präzisen Definition dessen, was klassifiziert werden soll, ist eine umfassende Analyse zur Art der Daten sehr wichtig. Man könnte z.B. hingehen und alle Tweets der letzten zwei Wochen aus Twitter extrahieren und dann annotieren. Das Ergebnis wäre ein Daten-Set mit sehr wenigen offensiven Tweets, wahrscheinlich nur 2–3 %. Eine Klassifikation, die dann jeden Tweet als nicht offensiv klassifizieren würde, hätte sehr gute Gesamtwerte in der Evaluation, denn sie würde ja in den meisten Fällen richtig liegen (siehe auch unsere Diskussion des Accuracy-Werts bei unausgewogenen Daten im Abschn. 3.4). Diese Klassifikation würde uns aber nicht weiterhelfen bei der Lösung des Problems. Man könnte auch mit offensiven Stichwörtern – also z.B. Schimpfwörtern – nach offensiven Tweets suchen. Eine Klassifikation könnte dann aber einfach einen Abgleich mit diesen Stichwörtern machen und wäre dann schnell fertig. Tweets mit Wörtern, die nicht in dieser Liste stehen, würden aber nicht erkannt. Bei der GermEval 2018 und 2019 haben wir daher einen anderen Weg gewählt: Wir haben zunächst mit einschlägigen Stichwörtern nach Twitter-Usern gesucht, die häufig auch offensive Tweets posten. Dann haben wir aus den Timelines dieser User 200 Tweets ausgewählt und diese dann annotiert. Die Information über die User haben wir dann wieder gelöscht und die Daten um Retweets, extrem kurze Tweets

⁵<https://projects.fzai.h-da.de/iggsa/data-2019>.

und Tweets mit Links bereinigt. Auf diese Weise bekamen wir offensive und nicht offensive Tweets, die sich um ähnliche Themenfelder drehen und die sich sprachlich nicht allzu sehr unterscheiden, wie das der Fall wäre, wenn wir z. B. Tweets von Zeitungsredaktionen mit offensiven Tweets privater User gesammelt hätten. Da die Daten zu einem bestimmten Zeitpunkt gesammelt wurden, überwiegen Themen, die zu diesem Zeitpunkt aktuell in der Öffentlichkeit diskutiert wurden. Ein Problem ist noch, dass bestimmte Themen in diesen Daten überwiegend negativ bewertet wurden. Dies betraf z. B. Tweets zu Angela Merkel und Heiko Maas. Dies würde es einem Klassifikator wieder zu einfach machen. Zum sogenannten “Debiasing” der Daten haben wir daher noch neutrale oder positive Tweets zu diesen Themen aufgenommen. Im Fall der beiden Politiker waren das Tweets der Parteien CDU und SPD.

Der nächste Schritt ist die Annotation. Hier ist besondere Sorgfalt angesagt, denn wir benötigen nicht nur viele annotierte Daten, sondern die Annotation muss möglichst konsistent sein. Daher haben wir zunächst einige Daten mit drei Annotatoren parallel annotiert und die Annotationen dann verglichen und diskutiert. Im Rahmen dieser Diskussionen sind die oben erwähnten Richtlinien zur Annotation entstanden. Manche Äußerungen sind sehr schwer zu klassifizieren. Was würden Sie z. B. mit diesen machen, wenn Sie die Möglichkeiten OTHER, ABUSE, INSULT und PROFANITY haben?

- *Meine Kollegin hat unendlich gute Laune, weil sie nächste Woche drei Wochen Urlaub hat. Wünsche ihr den Tod.*
- *24 x “Digga”, 41 x “Ich schwör” und 3 x “Drecksjude”. Manche Imbissnachbarn möchte man gerne mal unbetäubt durchs Sieb streichen.*
- *Eine Ergänzung zur Täterbeschreibung. Täterherkunft fehlt. #Rosenheimerplatz #Messerattacke #München*
- *Alle 11 min fällt ein liberales Backpfeifengesicht auf einen Spruch von Merkel rein.*

Unklare Fälle haben wir schließlich nicht annotiert und aus den Daten herausgenommen.

Bei der Aufteilung der annotierten Daten in Trainings- und Testset haben wir dann noch darauf geachtet, dass die Klassenverteilung in den beiden Datensätzen sehr ähnlich ist.

10.2 Methoden zur automatischen Klassifikation

Einen Überblick über aktuelle Methoden zur automatischen Erkennung von aggressiver Sprache gibt (Mishra et al. 2019). Es wird dort deutlich, dass die weitaus meisten Methoden für englischsprachige Datensätze entwickelt wurden. Die Forschungsgruppen arbeiten mit ähnlichen Methoden wie die, die wir auch in der Sentiment-Analyse kennengelernt haben, mit Wörterbüchern, maschinellem Lernen und N-Grammen.

Mit den Daten der GermEval 2019 werden wir nun unser eigenes Klassifikationssystem implementieren. Dabei beginnen wir damit, ein heuristisches Verfahren mit Grenzwerten zu

implementieren, das unsere Sentiment-Analyse mit Wortlisten kombiniert. Wir beschränken uns auf die binäre Klassifikation in OFFENSE – OTHER. Laden Sie zunächst die Trainingsdaten und die Testdaten der GermEval 2019 von der Website des Projekts herunter (<https://projects.fzai.h-da.de/iggsa/data-2019/>).

Die annotierten Trainingsdaten haben ein sehr einfaches Format, z. B.:

@anna_Ilina Kann man diesen ganzen Scheiß noch glauben..? OFFENSE PROFANITY

Für die binäre Klassifikation ignorieren wir die rechts stehende Annotation (hier: PROFANITY). Unser Klassifikationssystem soll den Testdaten, die keine Annotation haben, eine Annotation hinzufügen. Die grundlegende Idee: Wir nutzen unsere Sentiment-Analyse, um positive und neutrale Aussagen mit OTHER zu klassifizieren. Bei negativen Aussagen gleichen wir diese mit zwei Wortlisten ab: einer Wortliste für ABUSE oder INSULT und einer Wortliste für PROFANITY.

Im ersten Schritt müssen wir die Tweets vorverarbeiten, das sogenannte “Preprocessing”. Wir löschen alle Zeichen heraus, die uns in der Programmierung Schwierigkeiten machen und schicken den Tweet durch spaCy für die Tokenisierung und eine linguistische Analyse:

```
def preprocessing(sent):
    sent = return "".join((i if ord(i) < 10000 else '\ufffd' for i in sent))
    sent = sent.replace('lBRI', '')
    analysis = nlp(sent)
    return(sent, analysis)
```

Für die binäre Klassifikation arbeiten wir mit separaten Wortlisten für OFFENSE (ABUSE, INSULT) und PROFANITY. Der Grund dafür ist: PROFANITY kann auch bei positivem Sentiment auftreten.

```
def hate_or_not(sent):
    sent, analysis = preprocessing(sent)           # die Vorverarbeitung,
                                                    # die wir oben definiert haben
    offense_value = 0
    profane_value = 0
    sentiment_value = my_sentiment(sent)           # hier die eigene Sentiment-Lösung aufrufen
    if sentiment_value > -0.1:
        # wenn der Tweet positives oder neutrales Sentiment hat, dann kann er höchsten PROFANITY sein
        offense_value = 0
        for token in analysis:
            if token.text.lower() in profane_words: # dafür wird nach Wörtern in
                profane_value = profane_value + 1   # der Liste der profanen
                                                    # Wörter nachgesehen
        else:
            # ein Tweet mit negativem Sentiment
            for token in analysis:
                if token.text.lower() in offensive_words: # es wird in der Liste mit
                    offense_value = offense_value + 1      # offensiven Wörtern nachgesehen
                elif token.text.lower() in profane_words:  # außerdem in der Liste mit
                    profane_value = profane_value + 1      # profanen Wörtern
    return (offense_value, profane_value)
```

Der Algorithmus gibt zunächst nur Werte aus, keine Entscheidung. In einem zweiten Schritt müssen die Grenzwerte festgelegt werden: Ab welchen Werten für “offense_value” und “profane_value” soll entschieden werden, den Tweet als OFFENSE zu klassifizieren? Diese Entscheidung sollte anhand von Tests auf den Trainingsdaten getroffen werden. Eine Alternative dazu, einen Grenzwert festzulegen, ist, die Werte in ein System zum maschinellen Lernen zu füttern. Dazu schreiben wir sie in die Trainingsdaten herein, um dann ein Modell darauf zu trainieren:

```
out = open('training_annotated.txt', 'w', encoding='utf-8', errors='ignore')
# wir öffnen die Ausgabedatei
with open('germeval2019.training-subtask1.2.txt', 'r', encoding='utf-8',
          errors='ignore', newline='') as csvfile:
    # wir öffnen die Trainingsdatei, die im TSV-Format vorliegt
    training_vanilla = csv.reader(csvfile, delimiter='\t')
    for line in training_vanilla:
        (sentiment_value, offense_value, profane_value) = hate_or_not(line[0])
    # wir berechnen die Werte für Sentiment, Offense und Profane
    out.write(line[0] + '\t' + str(sentiment_value) + '\t' + str(offense_value) +
              '\t' + str(profane_value) + '\t' + line[1] + '\n')
# wir schreiben alles in die Ausgabedatei
```

Jetzt können wir das Modell trainieren. Wir lesen unsere annotierten Daten mit dem Python-Modul Pandas ein:

```
dataset = pandas.read_csv('training_annotated.txt', sep='\t')
```

X sind die Werte, die wir auswerten lassen wollen, Y ist die Kategorie, die wir klassifizieren wollen, also OFFENSE oder OTHER. Die Werte sind Zahlen, “float”, die Kategorie ist ein String. Wir nutzen den “Decision Tree Classifier”⁶ und speichern das gelernte Modell unter dem Namen “offense_other_model.sav”:

```
array = dataset.values
X = array[:,1:4]
X = X.astype('float')
Y = array[:,4]
Y = Y.astype('str')

classifier = DecisionTreeClassifier()
classifier.fit(X, Y)
model_file = 'offense_other_model.sav'
pickle.dump(classifier, open(model_file, 'wb'))
```

⁶Das ist eine willkürliche Festlegung. Sie können im Rahmen der Übung den Einsatz weiterer Klassifikatoren testen.

Dann laden wir das Modell wieder hinein und definieren für einen Satz, dass zunächst die Werte berechnet werden und anschließend das Modell angewendet wird:

```
model = pickle.load(open('offense_other_model.sav', 'rb'))

def pred_offense_other(sent):
    (sentiment_value, offense_value, profane_value) = hate_or_not(sent)
    sent_array = [[sentiment_value, offense_value, profane_value]]
    result = model.predict(sent_array)[0]
    return(result)
```

Im Ergebnis können wir für einen Satz mit diesem Modell die Kategorie vorhersagen:

```
>>> pred_offense_other("Er ist halt ein kriminelles Dreckstück")
OFFENSE
```

Die Qualität dieses Klassifikationsalgorithmus ist abhängig von der Qualität des Preprocessings, der Sentiment-Analyse und der Wortlisten. Verschiedene Verfahren der Sentiment-Analyse haben wir in den vorhergehenden Kapiteln ausführlich behandelt. Für den Aufbau von Wortlisten gibt es in diesem Kontext die Möglichkeiten, die im Kap. 4 beschrieben wurden.

10.3 Zusammenfassung

In diesem Kapitel haben wir die automatische Klassifikation von Aggression in Meinungsäußerungen vorgestellt. Diese Klassifikation ist, auch wenn sie von Menschen und nicht von Algorithmen durchgeführt wird, nicht immer einfach und eindeutig. Im Rahmen der GermEval-Wettbewerbe 2018 und 2019 wurden die Klassen OFFENSE und OTHER mit den Unterklassen INSULT, ABUSE und PROFANITY eingeführt. Die Basis der automatischen Klassifikation sind manuell klassifizierte Daten. Wir haben verschiedene Methoden der Forschungsliteratur vorgestellt und anschließend einen einfachen Algorithmus für die binäre Klassifikation entwickelt. Hier können Sie mit weiteren Arbeiten ansetzen und den vorgestellten Algorithmus als Baseline betrachten.

10.4 Übungen

1. Prüfen Sie Ihr Wissen:

- Welche Arten von aggressiven Äußerungen gibt es?
- Warum sind annotierte Textkorpora wichtig für die Klassifikation?
- Welche Methoden der Klassifikation gibt es?

2. Setzen Sie Ihr neues Wissen ein:

- a) Implementieren Sie eine automatische Klassifikation von Tweets in OFFENSE und OTHER, auf Basis der Trainingsdaten der GermEval 2019.
- b) Wenden Sie Ihren Klassifikationsalgorithmus (bzw. Ihr Modell) auf die Testdaten an.
- c) Auf der Webseite der GermEval 2019 finden Sie das Evaluationsscript. Werten Sie damit Ihr System aus. Wie gut hätten Sie im Wettbewerb abgeschnitten?
- d) Welche Techniken haben diejenigen Gruppen verwendet, die besser abgeschnitten haben? Lesen Sie die Beiträge dieser Gruppen im Konferenzband (Struß et al. 2019).

3. Reflexion in Gruppenarbeit:

- a) Diskutieren Sie in der Gruppe, für welche Zwecke eine genaue Erkennung der Arten von aggressiven Meinungsäußerungen notwendig und sinnvoll wäre.
- b) Diskutieren Sie in der Gruppe konkrete Beispiele von Erfahrungen mit Aggression im Internet, die Sie gemacht haben. Welche Bereiche betreffen diese? Auf welchen Social-Media-Kanälen? Welche Herausforderungen für die automatische Erkennung stellen Sie für die konkreten Beispiele fest? Begründen Sie Ihre Einschätzung.

10.5 Weiterführende Literatur

Mishra et al. (2019) geben einen Überblick über aktuelle Methoden der Erkennung von Hassrede. Frühe⁷ Methoden zur automatischen Klassifikation basieren auf heuristischen Regeln über die Sprache (TF-IDF-Gewichte für Wörter, Sentiment-Analysen u. a.) in Kombination mit Decision Trees und manuell entwickelten Merkmalen für Support Vector Machines (SVM). Lexikonbasierte Ansätze folgen wie z. B. (Gitari et al. 2015). Dabei werden auch Methoden entwickelt, um solche Lexika automatisiert zu generieren, wie (Wiegand et al. 2018a). Eine alternative Methode zum Aufbau von Wörterlisten ist eine Kombination aus Character-N-Grams, Token-N-Grams und TF-IDF, wie sie vom Gewinner der Shared Task 2018 (Padilla Montani und Schüller 2018) implementiert wurde. Auf der Basis von Wörtern operieren Systeme mit Methoden der Bag-of-words (BOW), wobei in diesem Fall SVMs mit N-Grammen und Lexika kombiniert werden (z. B. (Salminen et al. 2018)). Dabei stellen (Nobata et al. 2016) fest, dass Methoden mit zeichenbasierten N-Grammen (character n-grams) sehr gute Ergebnisse liefern.

⁷Das Themengebiet ist noch neu, da die Problematik erst mit der Verbreitung der sozialen Netzwerke auftrat.

Seit Kurzem werden Methoden des Deep Learning auch für die Aufgabe der Klassifikation von offensiver Sprache verwendet, wie z. B. von (Mishra et al. [2019](#)). In einem systematischen Vergleich der Methoden (van Aken et al. [2018](#)), aber auch als Ergebnis der verschiedenen Shared Tasks der letzten Jahre, wurde deutlich, dass eine Kombination von neuronalen mit nicht-neuronalen Methoden aktuell zu den besten Ergebnissen führt.

Einen umfassenden Überblick über Ansätze zur automatischen Klassifikation deutschsprachiger Twitter-Daten bekommen Sie beim Lesen der Tagungsbände der GermEval 2018 und 2019, (Ruppenhofer et al. [2018b](#); Struß et al. [2019](#)). Ausführlichere Informationen, auch über andere Sprachen, bekommen Sie in (Ruppenhofer et al. [2020](#)).

Sentiment-Analyse im Unternehmenskontext und Softwarelösungen im Markt

11

Die Sentiment-Analyse hat sich in den letzten Jahren als manifester Anwendungszeitweig von Natural Language Processing entwickelt. Dies wird nicht nur durch die große Anzahl an wissenschaftlichen Veröffentlichungen oder durch die Vielfalt der Softwarelösungen deutlich, die im Markt angeboten werden, sondern auch dadurch, dass es zurzeit eine Reihe von verschiedenen Anwendungsfeldern in der Wirtschaft gibt, die Sentiment-Analyse als Methode einsetzen. In diesem Kapitel geben wir einen Überblick über die Softwarelösungen im Markt für deutschsprachige Sentiment-Analyse sowie die Praxis in der Wirtschaft. Zunächst betrachten wir den Markt für Sentiment-Analyse-Tools und – Services. Anschließend stellen wir Anwendungsfelder verschiedener Industrien vor und zeigen Fragestellungen, die mithilfe von Sentiment-Analyse im Unternehmenskontext beantwortet werden. Dabei geben wir Beispiele, wie die Erkenntnisse und Ergebnisse der Sentiment-Analyse für die Erfolgsmessung benutzt werden.

11.1 Markt für kostenpflichtige Sentiment-Analyse-Tools und -Services

Die Anzahl der professionellen bzw. kostenpflichtigen Software zur Sentiment-Analyse für den Einsatz in der Wirtschaft hat sich in den vergangenen Jahren weltweit vergrößert. Betrachtet man die Software ohne gleich die Sprachabdeckung zu berücksichtigen, gibt es neben den OpenSource-Lösungen eine Vielfalt von kommerziellen Lösungen. Das Spektrum reicht von Webservices über Software Development Kits (SDK), die für verschiedene Programmiersprachen verfügbar sind, bis hin zu Business-Lösungen, bei denen eine Sentiment-Analyse Bestandteil anderer Analytics-Prozessabläufe ist.

Während der letzten Jahre ist im Anbietermarkt viel Bewegung zu beobachten, beispielsweise in Form von Zusammenschließungen und Übernahmen (Brandwatch und Crimson Hexagon, IBM Watson und Alchemy API) oder bei Änderungen der Schnittstellen von rele-

vanten Kanälen und Portalen aufgrund von neuen Datenschutzregulierungen und -gesetzen oder auch mit einer Zunahme der Integration von Machine-Learning-Verfahren zur Analyse große Textmengen. Diese Bewegung führt zu rasanten Veränderungen und neuen Entwicklungen der Tools. Dies erfordert von Berufspraktikern, wie zum Beispiel Informationswissenschaftlern, Entwicklern im Bereich Natural Language Processing und Business Intelligence, Marktforschern und Analysten, Marketing- und Kommunikationsexperten sowie Unternehmensberatern, dass sie sich kontinuierlich informieren bzw. weiterbilden. Insbesondere hinsichtlich der Forschungsmöglichkeiten und Praxis-Anwendungen sowie der verfügbaren Services und Lösungen müssen sie ihr Wissen aktuell und relevant halten.

Nach Einschätzung von (Market Research Future 2019) soll der Markt bis 2023 eine beachtliche Größe von ca. 6 Mrd. USD erreichen.

Für den Zeitraum 2017 bis 2023 wird ein durchschnittliches jährliches Wachstum von 14 % vorausgesagt. Für den Bericht von (Market Research Future 2019) wurden 10 Anbieter analysiert, darunter IBM, Brandwatch, Clarabridge oder Aylien. Die analysierten Anbieter sind ausschließlich in den englischsprachigen Ländern USA, Großbritannien und Kanada ansässig. Die Anzahl der Angebote erhöht sich jedoch wesentlich, sobald wir auch diejenigen Anbieter betrachten, die in Asien oder weiteren europäischen Ländern ihren Firmensitz haben. Dazu kommen die Anbieter von Online- und Social-Media-Monitoring sowie Social-Listening-Tools, denn die Mehrheit dieser Lösungen bietet eine Sentiment-Analyse-Funktion an.

Durchgeführte Markterhebungen und -berichte legen in der Regel die Auswahlkriterien fest, auf dessen Grundlage sie den Markt auswerten. Nach der Zusammenstellung von Kriterien ergibt sich daher eine unterschiedliche Anzahl von Lösungen, über die die jeweiligen Marktanalysten berichten: Sie können zum Beispiel einen sehr umfangreichen Katalog von Lösungen umfassen, wenn sie deskriptiv vorgehen und wenige Kriterien für die Auswahl festlegen. Alternativ können sie eine kompakte Anzahl von relativ vergleichbaren Lösungen umfassen, wenn sie die Auswahlkriterien restriktiv halten. So zum Beispiel der Forrester-Wave-Bericht (Liu und Chien 2018), der insgesamt zehn Social-Listening-Technologien für B2C-Marketing weltweit auswertet. Die Entscheidung darüber, welche Anbieter evaluiert wurden, fußt auf Kriterien, die auch die wirtschaftliche Größe und Stabilität der Anbieter berücksichtigt. Ein Kriterium ist zum Beispiel, dass die Anbieter 2017 mindestens einen Umsatz in Höhe von 15 Mio. USD durch die Lizenzierung der Social-Listening-Technologie erzielt hatten. Ein weiteres Kriterium betrifft die Kundenbasis der Anbieter, die über 100 Großunternehmen zählen muss. Dabei wird ein Großunternehmen im Bericht so definiert, dass es entweder mindestens 1000 Arbeitnehmer hat oder einen Umsatz von mindestens einer Milliarde USD erzielt hat.

Einen anderen Ansatz verfolgt hingegen der Marktbericht von (Ideya 2018). Die Analyse umfasst 150 Social-Media-Tools und -Services bei denen die Sentiment-Analyse eine von vielen Funktionen sein kann.

Diese beiden Beispiele sollen zeigen, wie unterschiedlich die Marktteilnehmer für Markterhebungszwecke betrachtet werden. Das ist relevant, wenn man auf Basis einer

Markterhebung eine Entscheidung für eine konkrete Lösung treffen möchte, aber die am besten geeignete z. B. aus Kostengründen ausgeschlossen hat. Es gibt über die geographische Zugehörigkeit hinaus eine Reihe von Parametern, die einer Segmentierung der Sentiment-Analyse-Angebote dienen können, u. a.:

- das Bereitstellungsmodell (s. 11.1.1)
- das Preismodell (z. B. jährliche/monatliche Lizenz, Anzahl der Transaktionen/Calls p. M. usw.)
- die Tiefe der Sentiment-Analyse (s. 11.1.2)
- das Vorhandensein weiterer Dienstleistungen über die Analyse hinaus, z. B. Datenintegration, regelmäßige Reports
- die Branchen, die sie überwiegend unterstützen
- ihre Eignung für kleine oder große Unternehmen

Diese Parameter hängen mit den drei wesentlichen Herausforderungen für die Anbieter von Softwarelösungen zusammen:

- **Volumen:** Auch wenn die Menge der relevanten Textdaten an sich heute keine allzu große Herausforderungen darstellen dürfte, sind die Identifikation, die Lizenzbedingungen der Portalbetreiber für Datenabfragen und -Nutzung, sowie die Gewährleistung von Live-Analysen klare Herausforderungen, die sich auf die Preismodelle, die Analysetiefe und zusätzliche Dienstleistungen auswirken
- **Geschwindigkeit:** Aufkommen der neuen und relevanten unstrukturierten Textdaten über einen Zeitraum
- **Verschiedenheit/Vielfalt** der relevanten Datenquellen und Texttypen bzw. sprachlichen Besonderheiten von Textdaten, die analysiert werden sollen

Diese Herausforderungen wirken sich auf die Qualität und die Quantität des Leistungsumfangs aus, erschweren einen direkten Vergleich der Angebote und gestalten die Toolauswahl komplex. Betrachten wir die kostenpflichtigen Lösungen im Sentiment-Analytics-Markt genauer, wird schnell deutlich, dass die Anzahl der Software-Lösungen umfangreich ist. Im Folgenden erläutern wir vier Kategorisierungsmerkmale, die für die Einordnung sowie für eine Tool-Auswahl relevant sind.

11.1.1 Technische Bereitstellung der Lösung

Sentimentanalyse-Lösungen werden nach folgenden Bereitstellungsmodellen angeboten:

- Cloud-/webbasierte Lösung in Form von “Software-as-a-Service (SaaS)”
- Anwendungsprogramm-Schnittstelle, “Application Program Interface (API)”

- Teil-Funktionalität oder Service einer vorhandenen Plattform, zum Beispiel für Business Intelligence oder Social Listening
- Vor-Ort-Bereitstellung (“on-premise”)
- Einbindung von Software anderer Hersteller in die Dienstleistung (“White Label”)
- Mobile App, oft zusätzlich zu einer SaaS-Bereitstellung

Die Arten der Bereitstellung lassen die unterschiedlichen Bedürfnisse der Kunden von Sentimentanalyse-Lösungen erkennen. Einerseits möchten Systemhersteller, zum Beispiel von Systemen für Social-Media-Management, Business-Intelligence oder Kundenservice, die Funktionalitäten ihrer Software durch Sentiment-Analyse erweitern. Kleine und mittelständische Unternehmen suchen andererseits nach kosteneffizienten und für ihren Bedarf anpassbaren Möglichkeiten zur Sentiment-Analyse, zum Beispiel für die Analyse von Kundenfeedback auf ihrer Shop- oder Website oder von Interaktionen auf Social Media. Agenturen benötigen robuste Plattformen mit Zugang zu möglichst vielen und vielfältigen Datenquellen, sodass sie Monitoring-, Analytics- und Social-Listening-Services für ihre Kunden anbieten können. Dabei geht es darum, Kampagnen, Themen, Produkte oder Marken zu beobachten und eine umfassende Tonalitäts-, Emotions- oder Sentiment-Analyse der Online-Erwähnungen für ihre Reportings zu generieren. IT- und Business-Berater suchen nach flexiblen Möglichkeiten, spezielle Anwendungen für ihre Kunden zu implementieren, ohne dass sie selbst eine (eigene) Sentiment-Analyse von Grund auf implementieren müssen. Alle diese Bedarfe sind der Grund für die Vielfalt der Bereitstellungsmodelle. Für die Unternehmen, die eine für sie passende Lösung aus der Fülle der Möglichkeiten auswählen, ist erforderlich, dass sie ihren Bedarf in Verhältnis zu ihren Geschäftszielen setzen (hierzu siehe auch Abschn. 11.3). Relevante Fragen für die Auswahl sind zum Beispiel:

- Welche konkreten Erkenntnisse, Indikatoren und Beratung werden benötigt?
- In welchem Arbeitskontext und von wie vielen Nutzern soll eine Sentiment-Analyse genutzt werden?
- Wie hoch ist das interne Know-how in diesem Bereich? Wie oft soll die Analyse durchgeführt werden?
- Ist eine 100%ige Sentiment-Analyse für über 50 Sprachen für mehr als 1–2 Geschäftsszenarien notwendig?
- Sollen sowohl Erwähnungen in Online- als auch Erwähnungen in Offline-Medien abgedeckt werden?
- Reichen Twitter und Facebook aus?
- Sollen ausschließlich Bewertungen auf Nutzerportalen (Amazon, Tripadvisor u.ä.) analysiert werden?

Einige der Antworten setzen bestimmte Bereitstellungsmodelle voraus, einige jedoch deuten auf weitere Kriterien hin, die die Kategorisierung der Lösungen betreffen.

11.1.2 Art der Sentiment-Analyse

Lösungen unterscheiden sich in der Art der Sentiment-Analyse, die sie durchführen:

- Dokumentbasierte Sentiment-Analyse: Es wird die Polarität für einen Text bestehend aus mehreren Sätzen oder Phrasen berechnet. Dieser Text kann ein ganzes Dokument, eine Nutzer-Bewertung oder Rezension, ein Facebook-Kommentar, ein Tweet usw. sein (siehe Kap. 3).
- Satzbasierte Sentiment-Analyse: Es wird die Polarität für jeden einzelnen Satz eines Beitrags ermittelt (siehe Kap. 5).
- Aspektbasierte Sentiment-Analyse: Für jeden Aspekt, der innerhalb eines Beitrags erkannt wird, wird die Polarität ermittelt (siehe Kap. 6).

Die meisten Lösungen im Markt unterstützen lediglich eine dokumentbasierte Sentiment-Analyse. Satzbasierte oder aspektbasierte Sentimentanalyse-Lösungen setzen oft voraus, dass die Nutzer bereits relevante Meinungsdaten z. B. als CSV-Dateien gesammelt und aufbereitet haben. Die Art der Klassifikation kann variieren (siehe Abschn. 3.5):

- Binäre Klassifikation: positiv und negativ
- Klassifikation mit drei möglichen Polaritäten: positiv, neutral, negativ. Wobei eine neutrale Polarität auch bedeuten kann, dass die Sentiment-Analyse für diesen Text nicht durchgeführt werden konnte, z. B. weil die Sprache des Textes nicht unterstützt wird.
- Klassifikation mit vier möglichen Polaritäten: negativ, neutral, positiv und gemischt.
- Regression: sehr negativ, negativ, neutral, positiv und sehr positiv.

Über die Zuordnung der Äußerungen von User Generated Content (UGC) zu Sentiment-Kategorien hinaus kann die Analyse die Erkennung von konkreten Emotionen in Text beinhalten. Nach (Yadollahi et al. 2017) ist Emotion Mining die Untersuchung von geäußerten Emotionen, zum Beispiel Freude, Trauer usw., in einem natürlichsprachigen Text. Teilaufgaben der Emotionsanalyse sind die Erkennung und die Polaritätsklassifikation von Emotionen, sowie die Kategorisierung von erkannten Emotionen und die Erkennung von Faktoren oder Indikatoren, die eine erkannte Emotion erklären bzw. begründen. Die Mehrheit der kommerziellen Lösungen unterstützt lediglich eine Sentiment-Analyse und bietet keine Funktion zur Emotionsanalyse an.

Weitere Kategorisierungsmöglichkeiten betreffen die Korrekturmöglichkeiten nach der automatischen Analyse, d.h. ob Nutzer nach der automatischen Analyse eine manuelle Korrektur der automatischen Analyse durchführen können. Dazu kommt die Möglichkeit, dass der Algorithmus für die Sentiment-Analyse lernfähig ist, sodass durch eine manuelle Korrektur die künftigen Klassifikationen automatisch verbessert werden.

11.1.3 Sprachenabdeckung

Fast alle professionellen Sentimentanalyse-Lösungen werden für die englische Sprache angeboten. Darüber hinaus werden oft weitere Sprachen unterstützt. Manche Lösungen unterstützen über 50 Sprachen, wobei die Abdeckung einer Lösung für eine aspektbasierte Sentiment-Analyse selten über 10 Sprachen hinausgeht. Eine deutschsprachige Sentiment-Analyse wird nicht von jedem Tool angeboten. Zum Beispiel decken zurzeit NetOwl von SRI International, Rosette Text Analytics, Intellexer Semantic Analyser oder MeaningCloud die Analyse deutschsprachiger Texten nicht ab.

11.1.4 Leistungsumfang und Funktionen

Software-Lösungen unterstützen folgende Prozesse bzw. bieten folgende Funktionen an, wobei nicht alle Funktionen von allen Lösungen angeboten werden:

- Daten-Crawling, -Extraktion und -Sammlung aus unterschiedlichen online verfügbaren Datenquellen
- Datenaufbereitung der gesammelten Daten
- Automatische Erkennung, Klassifikation und Visualisierung von Sentiments und Emotionen
- Professionelle Beratung für das Setup und die Anpassung von Standard-Lösungen hinsichtlich des konkreten Anwendungsbedarfs von Kunden, z. B. Domänen-Anpassung, Datenerfassung (Data-Scraping) und Einbindung weiterer Datenquellen
- Integration der Lösung in bestehende Systeme
- Erstellung von Reportings und Alerts
- Support und Maintenance

Welche Funktionen für ein Unternehmen relevant sind, hängt von den Zielen und dem Anwendungskontext für die Sentiment-Analyse ab.

Zusammenfassend machen die Kategorisierungsmerkmale zum einen deutlich, welche Vielfalt die Einsatzmöglichkeiten der Sentiment-Analyse im Unternehmen haben kann. Zum anderen wird aber dadurch auch deutlich, dass die Auswahl für die passende Lösung eine Herausforderung für Unternehmen sein kann. Die Auswahl der geeigneten Software kann nicht ohne die Berücksichtigung der konkreten Anwendungsfelder, der spezifischen Unternehmensfaktoren und der Geschäftsziele getroffen werden. Diese müssen im Vorfeld geklärt und festgelegt werden, damit passende Bewertungskriterien für die Tool-Auswahl definiert werden können.

11.2 Tools für deutschsprachige Texte

Im Folgenden stellen wir in kompakter Form zehn Lösungen vor, die eine Sentiment-Analyse deutschsprachiger Textdaten unterstützen. Diese Lösungen sind beispielhaft und wurden ausgewählt, um die Vielfalt der Lösungen sowie die Ansätze und Bereitstellungsmodelle aufzuzeigen. Weder ist Ziel dieses Kapitels eine vollständige Liste der verfügbaren Lösungen in Markt zu geben, noch sprechen wir durch die Auswahl eine Empfehlung aus. Die Tools werden in alphabetischer Reihenfolge vorgestellt, die Reihenfolge drückt daher keine Wertung aus.

11.2.1 Amazon Comprehend

Amazon Comprehend¹ ist Teil der Amazon Web Services (AWS) Infrastruktur. Mit Comprehend stellt Amazon einen NLP-Service zur Verfügung, der mit Methoden des maschinellen Lernens (darunter Deep-Learning-Algorithmen) das Sentiment eines Textes automatisch erkennt. Ähnlich wie MonkeyLearn (siehe Abschn. 11.2.4) setzt Amazon Comprehend voraus, dass die Unternehmen bzw. die Nutzer des Services, relevante Textdaten (z. B. Kunden-E-mails, Social-Media-Beiträge, Online-Reviews) bereits gesammelt haben. Diese werden als Trainingsdaten für das maschinelle Lernen verwendet, um geschäftsrelevante Informationen wie positive oder negative Kundenerfahrungen und Kundenstimmung aus dem Text automatisch zu gewinnen (siehe Abschn. 3.6). Der Service kann u. a. eine Spracherkennung, eine Sentiment-Analyse sowie eine Entitätserkennung durchführen. Bei der Sentiment-Analyse wird ein Text einer negativen, neutralen, positiven oder gemischten Polarität zugeordnet. Zur Polaritätsklassifikation gehört ein Konfidenz-Wert, der die Wahrscheinlichkeit für die Korrektheit der automatisch erkannten Polarität festlegt.

Der Screenshot in Abb. 11.1 zeigt das Ergebnis der Sentiment-Analyse innerhalb des Amazon-Interfaces von (Posey 2019).

Die Sentiment-Analyse von Amazon Comprehend ist dokumentbasiert. Amazon Comprehend ist eine plattformbasierte Schnittstelle, die als AWS-Dienst vollständig serverlos benutzt werden kann, d. h. weder muss ein dafür gesonderter Server bereitgestellt werden noch müssen die Nutzer eigene Machine-Learning-Modelle entwickeln oder trainieren. Die Kosten werden auf Basis der tatsächlichen Aufrufe des Services und der Transaktionen für die Nutzung ermittelt. Der Service dürfte insbesondere für Unternehmen interessant sein, die nicht über personelle Ressourcen mit NLP-Wissen verfügen, um ein professionelles Setup für die Sentiment-Analyse zu implementieren, das für große Mengen von geschäftsrelevanten Textdaten aus verschiedenen Quellen und Systemen, z. B. CRM, Online-Shop oder Social-Media-Präsenzen geeignet ist.

¹ <https://aws.amazon.com/de/comprehend/>

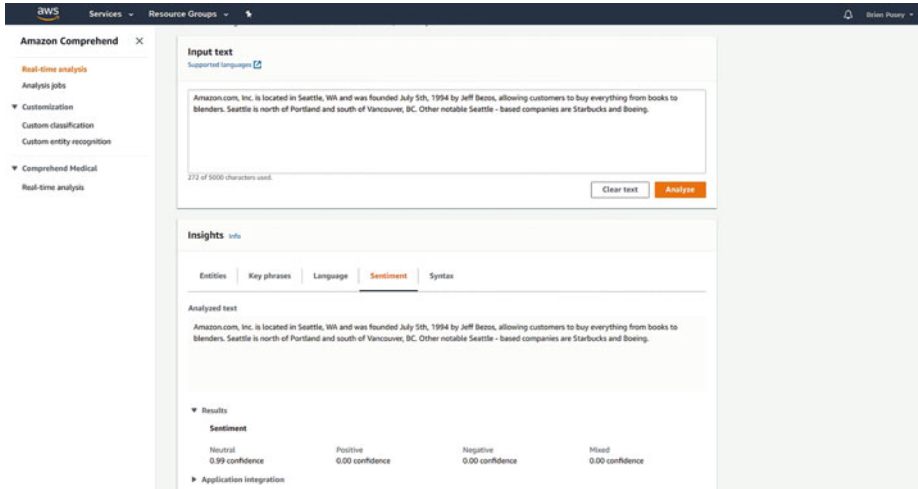


Abb.11.1 Beispiel einer neutralen Polarität der Sentiment-Analyse mit Amazon Comprehend (Posey 2019)

11.2.2 Cogito Intelligence Plattform von Expert System

Die Software-Lösungen von Expert System² für Information Intelligence und Risiko-Monitoring beinhalten eine "Linguistic Engine" zur Analyse natürlichsprachiger Texte. Kern der Cogito-Lösung ist ein umfangreiches semantisches Netz für die Sprachen, die unterstützt werden, darunter auch Deutsch. Dieses semantische Netz bildet die Konzepte einer Sprache zusammen mit ihren verschiedenen Bedeutungen ab, z. B. ist das Konzept Jaguar im semantischen Netz von Cogito sowohl als Automodell als auch als Tier vorhanden. Darüber hinaus sind die Konzepte über semantische Beziehungen wie Synonyme oder Ober-/Unterbegriffe zwischen den Konzepten verbunden. Dadurch ist eine semantische Analyse von unstrukturierten Textdaten möglich. Die Lösung wird als API angeboten und bietet bereits vorimplementierte domänenspezifische Taxonomien an, zum Beispiel zu den Themen Terrorismus, Cyber-Crime oder geographische Domänen. Durch die Technologie ist es möglich, mehrdeutige Aussagen zu disambiguieren. Hinsichtlich der Sentiment-Analyse verfolgt Cogito einen aspektbasierten Ansatz (siehe Kap. 6). Außerdem wird eine Emotionserkennung angeboten.

²<https://expertsystem.com/de/products/cogito-intelligence-platform/>

11.2.3 InMap, Insius

Der Lösungsanbieter Insius³ wurde als Spin-off der Universität Köln gegründet und bietet die Software InMap an, welche deutschsprachige Online-Texte nach Themen und Stimmungen analysiert. InMap wird als On-Premise-Lösung angeboten und unterstützt die Funktionen des Datencrawlings online sowie der Sentiment-Analyse und Auswertung der gesammelten Daten. Bereits beim Crawling verfolgt InMap den Ansatz durch ein Maschinenlernverfahren themenrelevante Inhalte (Meinungsäußerungen) zu erkennen und gleichzeitig irrelevante Beiträge zu verhindern – ein Manko bei vielen schlüsselwortbasierten Crawlern. Für die so gesammelten Texte führt InMap eine Sentiment-Analyse durch, indem nicht der Beitrag als Ganzes analysiert wird, sondern die erkannten Aussagen innerhalb des Beitrags getrennt voneinander. Somit wird eine aspektbasierte Sentiment-Analyse durchgeführt, die Sentiments auf der Konzept- und nicht der Dokument-Ebene zuordnet. Dabei werden konkrete Aspekte, d. h. die Merkmale oder Eigenschaften, die die Nutzer schätzen oder kritisieren, ermittelt und für die Auswertung genutzt (siehe Kap. 6).

11.2.4 Monkey Learn

MonkeyLearn⁴ bietet eine Plattform an, mit der man Texte durch den Einsatz von Verfahren des maschinellen Lernens nach drei Sentiment-Kategorien klassifizieren kann. Es gibt Integrationen von MonkeyLearn für Microsoft-Excel, Google-Sheets, Zendesk und andere, es ist aber auch möglich, die Sentiment-Analyse in Dritt-Software zu integrieren. Für kleinere Projekte und Anwendungen ist MonkeyLearn kostenfrei. Für große und umfangreiche Szenarien und Anwendungen ist die Lösung jedoch kostenpflichtig. Es wird vorausgesetzt, dass das Unternehmen relevante Daten bereits gesammelt hat, zum Beispiel über Befragungen, den Kundeservice-Desk, Social-Media, E-Mail-, Chat- oder CRM-Systeme usw., die für Analyse Zwecke verwendet werden sollen. Nutzer ohne Programmierkenntnisse haben die Möglichkeit, ein eigenes Modell für die Sentiment-Analyse zu trainieren und anschließend zu testen und somit ein Sentimentanalyse-System für ihre Business- und Anwendungszwecke relativ schnell zu entwickeln. Nutzer mit Programmierkenntnissen stehen eine API-Schnittstelle und SDKs zur Verfügung. MonkeyLearn unterstützt eine aspektbasierte Sentiment-Analyse. Nutzer können zunächst Trainingsdaten mithilfe des “Aspect Classifiers” für relevante Aspekte trainieren, die analysiert werden sollen (z. B. Design, Motor, Service für ein Auto-Modell) und sie anschließend Polaritätsklassen für den jeweiligen Aspekt zuordnen. Voraussetzung für diese Art der Analyse ist, dass die Daten vorab so aufbereitet werden, dass sie in Meinungseinheiten aufgeteilt wurden. Zum Beispiel für die Äußerung “Das Frühstück war super, die Dame an der Rezeption war leider unmöglich und

³<http://insius.com/de/home/>

⁴<https://monkeylearn.com>

frech” werden Meinungen für zwei Aspekte geäußert. Für die aspektbasierte Sentiment-Analyse werden zwei Meinungseinheiten extrahiert: a. “Das Frühstück war super”, und b. “die Dame an der Rezeption war leider unmöglich und frech”.

11.2.5 OpenText Magellan Text Mining

OpenText Magellan⁵ ist eine Plattform des in Kanada ansässigen Unternehmens OpenText. Das Unternehmen ist im Bereich Enterprise Information Management tätig und hat mehrere Niederlassungen, darunter auch in Deutschland. Die Plattform OpenText Magellan wird für Unternehmen aus verschiedenen Branchen als Online- oder On-Premise-Lösung zur Verfügung gestellt. OpenText Magellan bietet Funktionen für die Analyse von strukturierten sowie unstrukturierten Daten an. Für letztere bietet die Lösung den Text-Mining-Service mit einer Sentiment-Analyse für mehrere Sprachen inkl. Deutsch an. Der Service extrahiert Phrasen und Entitäten (z. B. Personen-, Orts-, Organisations- oder Ereignisnamen sowie Datumsangaben) und identifiziert Themen, Subjektivität und Polarität für die Meinungsäußerungen eines Textes. Die Ergebnisse der Analyse werden zusammengefasst in einem separaten Tab des OpenText-Dashboards visualisiert, wie Abb. 11.2 zeigt.

11.2.6 ParallelDots

ParallelDots⁶ stellt Schnittstellen zur Sentiment- und Emotionsanalyse für mehrere Sprachen zur Verfügung, dabei wird auch Deutsch abgedeckt. Des Weiteren können zusätzliche APIs zur Emotionsanalyse von Gesichtsbildern oder zur Erkennung von Sarkasmus lizenziert werden. Anbieter von Software-Lösungen, die ihren Kunden eine Sentiment-Analyse anbieten möchten, können die APIs in ihre eigenen Systeme integrieren. Einzelne Unternehmen können die Software auch in ihre Webseiten einbinden, eine Bereitstellung auf private Clouds oder eine On-Premise-Bereitstellung verwenden. Einzelne Nutzer können das ParallelDots-Plugin für Spreadsheets zum Beispiel für Microsoft-Excel oder die SaaS-Tools nutzen. Als Input der Software dienen Daten, die die Unternehmen oder Nutzer vorab gesammelt und für Sentimentanalyse-Zwecke aufbereitet haben. ParallelDots analysiert einen Beitrag als Ganzes nach Sentiments und Emotionen und vergibt einen Wert, der den Beitrag als positiv, neutral oder negativ klassifiziert. Abb. 11.3 zeigt eine Visualisierung des Ergebnisses der Sentiment-Analyse innerhalb der Webdemo.

Das Gesamtsentiment des Beispielbeitrags in dieser Demo wird mit knapp 80 % als negativ kategorisiert. Dabei wird für die Analyse nicht berücksichtigt, ob der Beitrag aus mehreren Sätzen besteht, ob überhaupt Meinungen im Text geäußert werden oder ob unterschiedliche Meinungen im Text geäußert werden.

⁵<https://www.opentext.com/products-and-solutions/products/ai-and-analytics/opentext-magellan>

⁶<https://www.paralldots.com/>

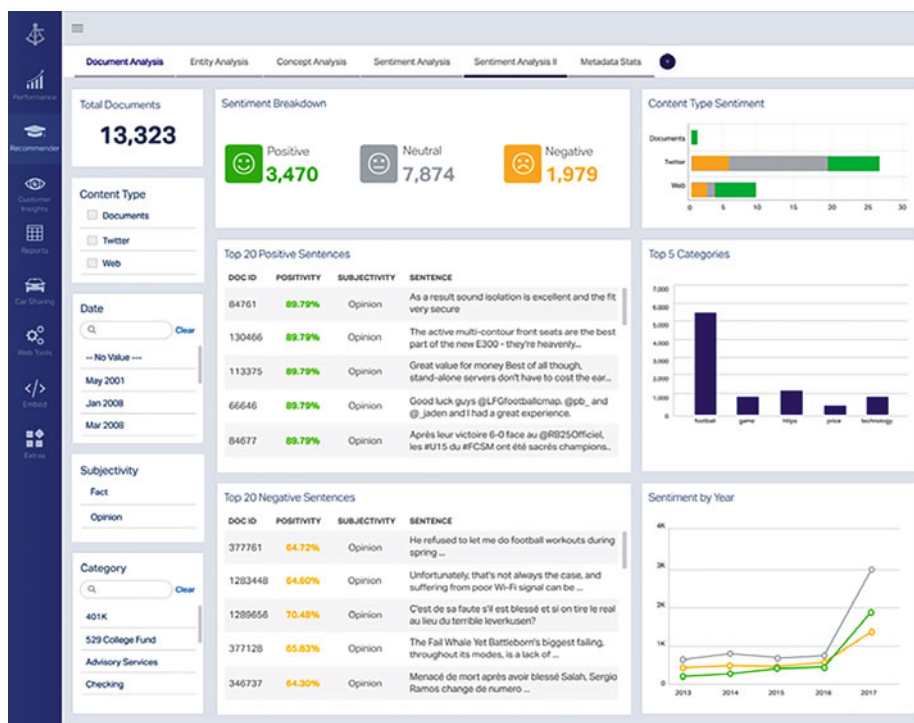
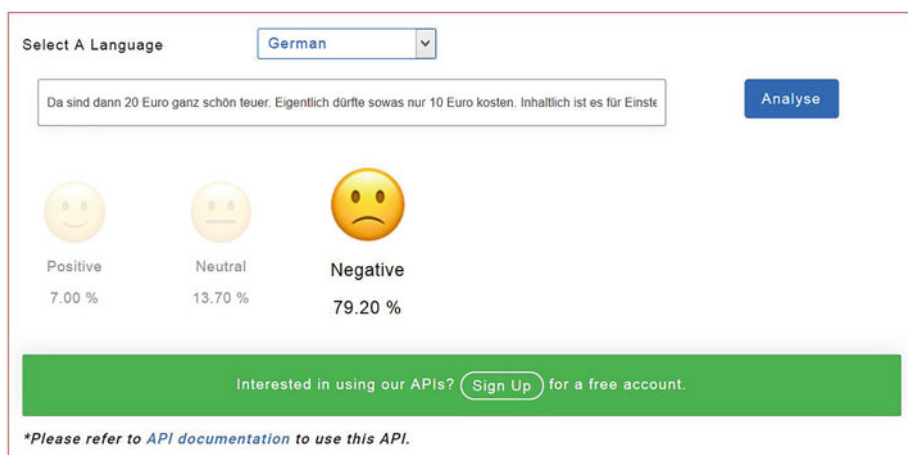


Abb. 11.2 Sentiment-Analyse von OpenText Manguerra (OpenText 2019)

Abb. 11.3 Beispiel für das Ergebnis der Sentiment-Analyse von ParallelDots für Deutsch. (Demo unter <https://www.paralldots.com/sentiment-analysis>)

11.2.7 SAS® Visual Text Analytics

SAS Analytics Software and Solutions bietet Firmen das Tool-Kit Visual Text Analytics zur multilingualen Sentiment-Analyse an. Laut SAS wird die “Out-of-the-Box-Textanalyse” für 33 Sprachen angeboten, zu denen auch Deutsch gehört. Die Funktionen für Sentiment-Analyse beinhalten zum einen das Data-Scraping von relevanten Quellen wie Online-Portalen, Webseiten oder Social-Media-Kanäle, und zum anderen die automatische Klassifikation der gesammelten Textdaten nach Sentiment. Die Analyseergebnisse können in verschiedenen Reports visualisiert werden. Als Beispiel dient Abb. 11.4, die eine Visualisierung der Sentiment-Analyse der Kommentare zum Stromausfall beim Super-Bowl-Spiel 2013 in New Orleans (Zaratsian et al. 2013) zeigt.

Der SAS-Ansatz für die Sentiment-Analyse besteht aus einem Mix von linguistischen und Machine-Learning-Verfahren. Es steht eine domänen-unabhängige Taxonomie für 14 Sprachen zur Verfügung. Interessant dabei ist, dass Visual Text Analytics Daten sowohl auf Dokument-Ebene nach Gesamtsentiment analysiert als auch eine aspektbasierte Sentiment-Analyse anbietet. SAS Visual Text Analytics wird auch für mobile Geräte angeboten.

11.2.8 Sentiment Intelligence in SAP Hana

Bei SAP Hana handelt sich um eine offene Plattform des Unternehmens SAP. Die Plattform unterstützt Unternehmen dabei, große Datenmengen, auch Textdaten, automatisch auszuwerten. SAP Hana beinhaltet Data-Mining und Textanalyse-Funktionen. Die Sentiment-Intelligence-Funktion gehört zu den letzteren: Sie führt eine Sentiment-Analyse für unstruk-

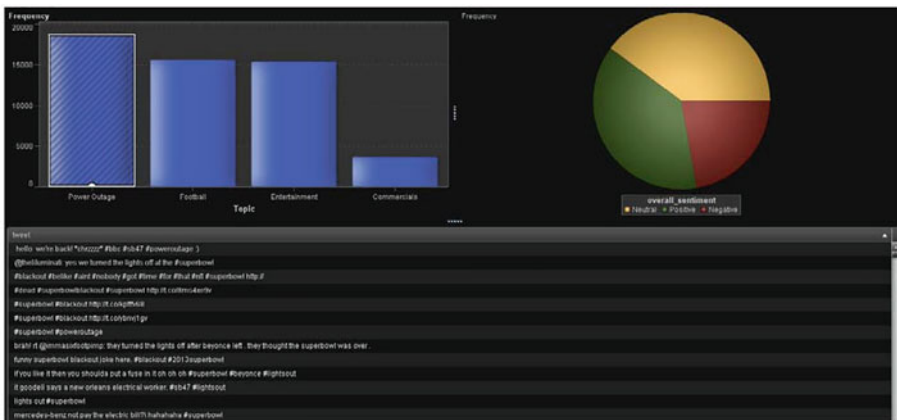


Figure 8. High-Level Categories Found within the Tweets Pertaining to Super Bowl Blackout

Abb. 11.4 Beispiel SAS-Sentiment-Analyse (Zaratsian et al. 2013, S. 8)

turierte Textdaten in neun Sprachen durch, darunter auch Deutsch. Sentiment-Analyse wird bereits von SAP-Produkten genutzt, wie zum Beispiel das CEI (Customer-Engagement-Intelligence). Mit der Sentiment-Analyse werden Erwähnungen bestimmter Konzepte zusammengefasst und jeder Beitrag wird einer von insgesamt fünf Kategorien zugeordnet: neutral, stark positiv, stark negativ, schwach positiv sowie schwach negativ. In einem Dashboard werden die Erwähnungen aufgelistet und die Analyseergebnisse visualisiert: Mittels verschiedener Typen von Diagrammen wird die Häufigkeit der automatisch erkannten Sentiments pro Beitrag gezeigt oder das Häufigkeitsverhältnis für die beobachteten Kanäle wird visualisiert.

Damit man die Sentiment-Analyse-Funktion nutzen kann, ist es nötig, dass man sie direkt durch SAP-Produkte implementiert oder alternativ als native SAP-HANA-Entwicklung durch Dritt-Anbieter realisieren lässt.

11.2.9 Sentiment Lab von m-result

Das Unternehmen m-result mit Sitz in Mainz stellt mit Sentiment Lab⁷ einen webbasierten Service zur Inhalts- und Sentiment-Analyse deutschsprachiger Online-Texte bzw. -Bewertungen zur Verfügung. Für den Zugang zu einem Dashboard über einen Webserver bietet m-result eine Basis- und eine Premium-Lizenz-Variante an. Die Sentiment-Lab-Dashboards werden für vordefinierte Domänen erstellt, z. B. für die Domänen Versicherung, Energie, Automotive. Der Nutzer kann die tagesaktuellen Analyseergebnisse auswerten lassen und verschiedene Visualisierungsmöglichkeiten nutzen, um sich zum Beispiel über konkrete Produkte (Automarken oder – Modelle) und ihre Eigenschaften (Design, Motor, Kundenservice) in dem untersuchten Bereich zu informieren. Somit wird es mit der Sentiment-Analyse möglich, sich über die Kundenmeinungen zu informieren, aber auch über die Positionierung der Marke oder der Wettbewerber auf Basis der ausgewerteten Textdaten, zum Beispiel durch die Ermittlung der Metrik “Social Buzz” (Gesamtheit des Erwähnungen auf den beobachteten Kanälen) pro Marke im Verhältnis zu den ermittelten Sentiments. Ein Visualisierungsbeispiel zeigt Abb. 11.5 für das Thema E-Mobilität: Die verschiedenen Automarken werden in Bezug zu der ermittelten Polarität von -1 (negatives Sentiment) bis +1 (positives Sentiment) und zu den aggregierten Social Buzz positioniert. Die Grafik macht deutlich, dass bezogen auf die gesammelten und ausgewerteten Meinungen zum Thema E-Mobilität Toyota im Vergleich zu den anderen Automobil-Marken am häufigsten und am positivsten erwähnt wurde.

⁷ <https://www.sentimentlab.com/>

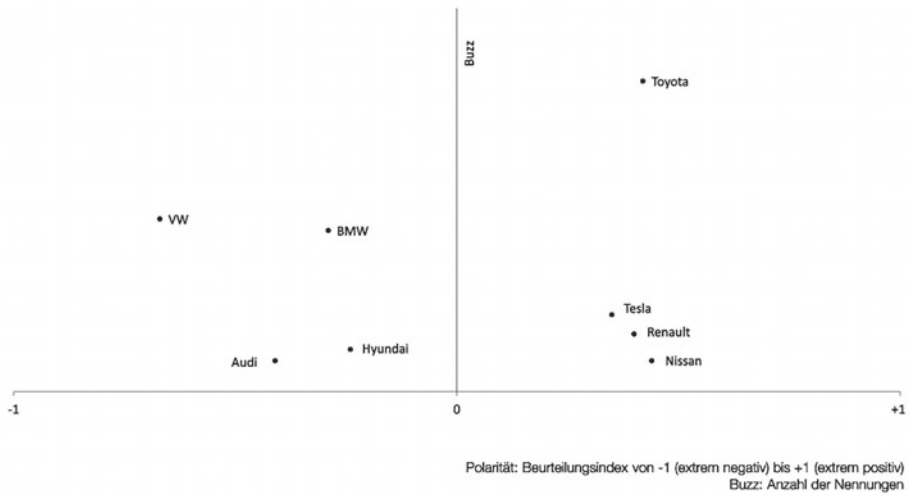


Abb. 11.5 Visualisierung von Sentiment Lab für Polarität und Social Buzz zur E-Mobilität (Latendorf et al. 2017)

11.2.10 Ubermetrics

Ubermetrics⁸ stellt eine Plattform als SaaS bereit, die Funktionen für das Media-Monitoring anbietet. Die Software beinhaltet Funktionen für das Crawling von verschiedenen Online- und Print-Quellen, die Aufbereitung der Daten und ihre Visualisierung in einem Dashboard. Die Daten können nach verschiedenen Kriterien visualisiert und gefiltert werden, zum Beispiel hinsichtlich der Social-Media-Kanäle, Sentiments, Zeit der Veröffentlichung, Autoren usw. Ubermetrics gehört zu den Online-Marketing und -PR Lösungen, die Social Listening, Social Media Monitoring und allgemein Monitoring unterstützen. Über SaaS hinaus steht anderen Softwareanbietern eine API zur Integration in Drittanwendungen zur Verfügung sowie die Möglichkeit einer White-Label-Lizenz. Nutzer können innerhalb der Anwendung Suchagenten definieren. Hierzu werden relevante Keywords benutzt, die mithilfe von Booleschen Operatoren den Suchraum für das Monitoring und die Analyse festlegen. So kann man zum Beispiel nach Erwähnungen der Themen E-Mobilität und Energieeffizienz suchen, indem eine Suche nach den Schreibvarianten “emobility” oder “e-mobility” oder “e mobility” als alternative Begriffe definiert wird, die mit den Begriffen “Energieeffizienz” oder “Stromverbrauch” gekoppelt wird. Die Ergebnisse der Suche sind Erwähnungen (Treffer), die nach ihrer Zugehörigkeit zu einem Media-Segment, zum Beispiel Nachrichten, Tweet, Kommentare usw. gefiltert werden können, wie Abb. 11.6 zeigt.

Jeder Treffer in einer Sprache, für die Ubermetrics eine Sentiment-Analyse unterstützt, wird nach Polarität (negativ, neutral oder positiv) klassifiziert, siehe Abb. 11.7.

⁸<https://www.ubermetrics-technologies.com/de/>

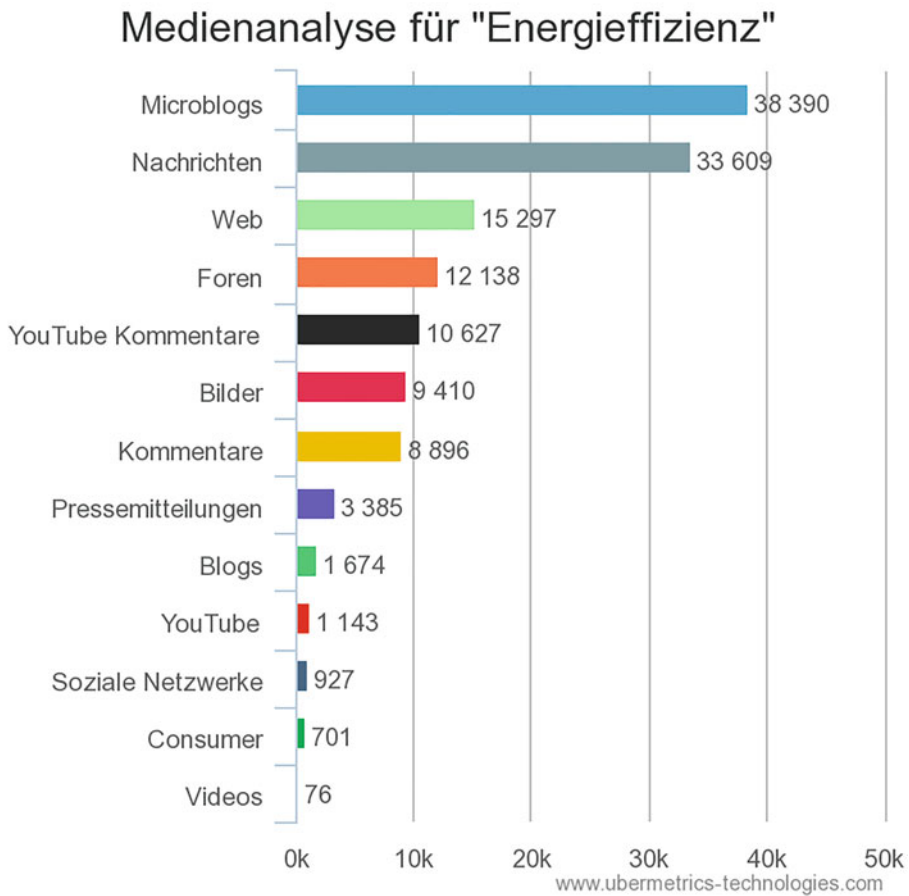


Abb. 11.6 Beispiel für die Ergebnisse zum Thema Energieeffizienz im Ubermetrics-Dashboard

Somit findet eine Sentiment-Analyse auf Dokument-Ebene statt, ähnlich wie bei dem Ansatz von ParallelDots. Der Algorithmus für die Sentiment-Klassifikation prüft weder, ob eine oder mehrere Meinungen im Text geäußert werden noch, ob Teile eines Beitrags unterschiedliche Meinungen ausdrücken. Die Nutzer haben die Möglichkeit, falsche Klassifikationen nachträglich manuell zu korrigieren und können Quellen, die nicht für die Analyse berücksichtigt oder gefunden wurden, manuell der Analyse hinzufügen.

11.3 Anwendung der Sentiment-Analyse in der Praxis

Wird Sentiment-Analyse als Methode im Unternehmenskontext eingesetzt, so geschieht dies typischerweise innerhalb der Abteilungen der Marketing- und Unternehmenskommunika-

Mediasentimentanalyse für "Automotive Themen"

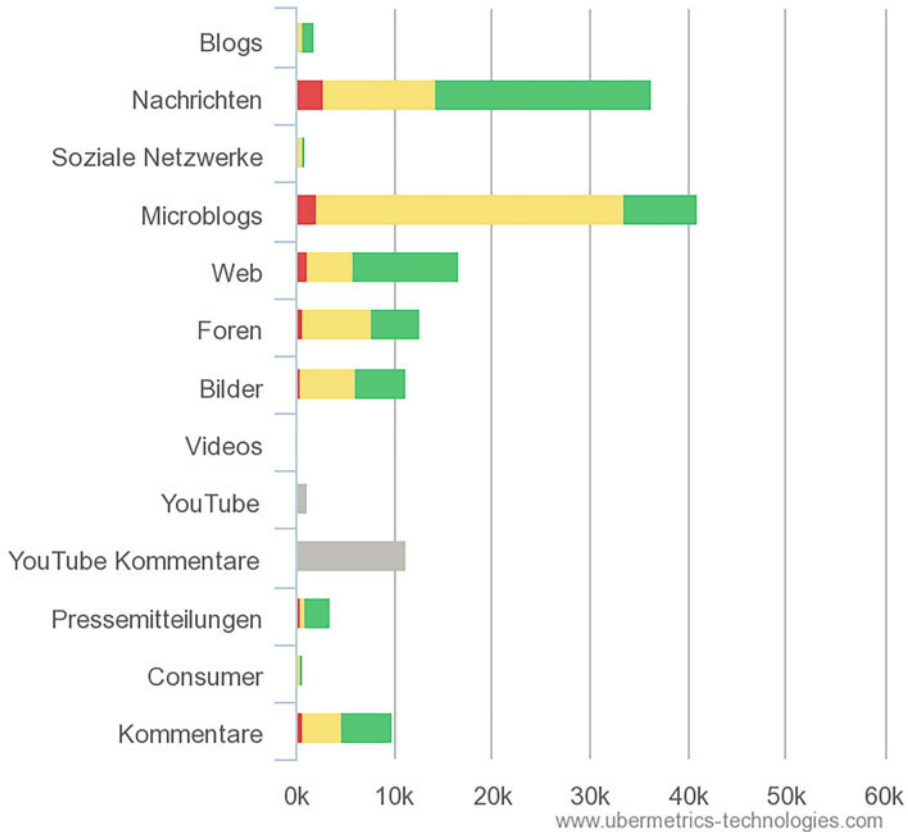


Abb. 11.7 Visualisierung der Sentiment-Analyse in Ubermetrics

tion, des Vertriebs, der Human Resources (HR) oder im Bereich Produktentwicklung. Die Ziele des Einsatzes sind u. a. die qualitative Analyse sowie die Messung des Erfolgs von Marketingkampagnen, die Wahrnehmung des Unternehmensimages, die Beobachtung und Analyse des Wettbewerbs, die Identifikation von Trends oder neuen Features für die Produktentwicklung und die Attraktivität als Arbeitgeber. Recherchiert man die Kundenreferenzen von Lösungsanbietern sowie die Publikationen von Unternehmen, die Sentiment-Analyse einsetzen, findet man folgende Branchen:

- Reise- und Hotelbranche, Gastgewerbe
- Finanzindustrie, Banksektor
- Retail, Einzelhandel

- Gesundheitswesen
- Fluglinien
- Medienindustrie, Journalismus
- Human Resource Management

Der Erkenntnisgewinn, den Sentiment-Analyse liefern soll, erstreckt sich auf unterschiedliche Interessensbereiche und betrifft Informationen und Wissen über Kunden, Zielgruppen, Produkte, Services, Stakeholder, Geschäftspartner, Angestellte, Image, Brand aber auch neue Ideen und Trends. Es kristallisieren sich folgende Anwendungsfelder heraus, für die die Erkenntnisse der Sentiment-Analyse für die Geschäftszwecke direkt nutzbar sind:

- Wissen über Zielgruppen:
 - geografische und soziodemografische Merkmale
 - konkrete Produkte und Services sowie Merkmale, die die Zielgruppe interessieren und worüber sie sich konkret äußern
 - weitere Eigenschaften, Features oder Services, die die Zielgruppe nutzt bzw. vermisst
- Wissen über Kunden:
 - Über welche Produkte und Services über welche Produkteigenschaften äußern sich die Kunden?
 - Welche Probleme werden wie oft angesprochen?
 - Was vermissen/wünschen sich die Kunden?
 - Welche Themen können zu Krisen führen (Krisenmanagement)?
- Wissen über den Wettbewerb (Competitive Intelligence)
 - Wie wird über die Produkte und Services der Wettbewerber gesprochen?
 - Wie positiv/negativ verläuft eine Kampagne der Wettbewerber?
 - Welche Themen werden von den Wettbewerbern und mit welcher Tonalität besprochen?
- Wissen für Produktmanagement und Innovation
 - Gibt es Vorschläge und Ideen zur Erweiterung des Produkt-/Service-Portfolios?
 - Welche Trends sind zu erkennen, die für die Weiterentwicklung relevant sind?
- Wissen für die Optimierung des Kundenservices bzw. des Kundensupports
 - Wie äußern sich die Kunden über den Kundensupport?
 - Welche Bereiche funktionieren gut, an welcher Stelle gibt es Handlungs- bzw. Optimierungsbedarf?
 - Wie ist das Verhältnis zwischen positiven und negativen Äußerungen zum Kundensupport?
 - Gibt es Plattformen/Kanäle, die für den Dialog mit den Kunden noch unterstützt werden müssen?

In Abhängigkeit vom Anwendungsfeld und der konkreten Fragestellung der Sentiment-Analyse legen Unternehmen fest und priorisieren, welche Art von Texten, wie granular, wie regelmäßig und mit welchem Volumen analysiert werden soll und welche Datenquel-

len beobachtet und analysiert werden sollen. Diese Entscheidungen sind Bestandteil der Grundlage für die Auswahl geeigneter professioneller Monitoring- und Sentiment-Analyse-Lösungen. Weitere Bestandteile bilden organisations- und unternehmensspezifische Faktoren, die nicht direkt mit dem Setup einer Sentiment-Analyse im Unternehmen zusammenhängen. Zu solchen gehören zum Beispiel vorhandene Erfahrung und Know-how innerhalb von Unternehmen sowie verfügbare personelle Ressourcen, vorhandene IT-Infrastruktur oder verfügbares Budget. Ist beispielsweise der Anwendungskontext im Unternehmen der, dass ausschließlich deutschsprachige Nutzer-Bewertungen online auf Amazon nach ihrem Gesamtsentiment analysiert werden sollen, ist die Tool-Auswahl eine andere, als wenn der Anwendungskontext die Beobachtung und Analyse sämtlicher Online-Bewertungen auf mehreren Produktportalen in mehreren Sprachen ist. Grundsätzlich dient der Polaritätswert der Sentiment-Analyse im jeweiligen Anwendungskontext als eine Metrik, d. h. eine quantitative Aussage, für die Erfolgsmessung. Zum Beispiel, um den Erfolg von Kampagnen zum Reputationsmanagement oder zur Zufriedenheit von Arbeitnehmern mit ihrem Arbeitgeber zu messen. Nachfolgend listen wir einige relevante Key-Performance-Indicators (KPI) auf, die den Polaritätswert der Sentiment-Analyse als eine Metrik verwenden, welche im Verhältnis zu mindestens einer weiteren Metrik gesetzt wird:

- Sentiment-Ratio (Stimmungsrate) bezogen auf das Verhältnis positiver zu negativer Beiträge
- Sentiment-Ratio bezogen auf das Verhältnis positiver und neutraler zu negativer Beiträge
- Sentiment-Index bezogen auf die Gesamtanzahl der Äußerungen (z. B. über ein Produkt, eine Marke usw.) und das Verhältnis von positiven und negativen Äußerungen. Dieser KPI ist z. B. relevant für das Reputationsmanagement.
- Empfehlungsrate bezogen auf das Verhältnis zwischen der Anzahl an Empfehlungen vs. der Anzahl an Negativempfehlungen für ein Produkt. Dieser KPI ist z. B. relevant für Messung der Kundenloyalität.
- Verhältnis der positiven Äußerungen zur Gesamtanzahl der Äußerungen für eine Entität oder einen Aspekt. Dieser KPI ist relevant z. B. für die Messung von erfolgreichem Krisenmanagement.
- Anteil kritischer Äußerungen am gesamten Unternehmensbuzz, d. h. Gesamtanzahl von Äußerungen zu einem Unternehmen. Dieser KPI ist u. a. für das Reputationsmanagement von Unternehmen relevant.
- Verhältnis zwischen der Summe der aktiven Nutzer und der Gesamtanzahl und Polaritätswert der Online-Beiträge: Dabei wird das Verhältnis von positiven und negativen Beiträgen im Verhältnis zur identifizierten Anzahl von aktiven Nutzern, die sich zu einem Thema oder einer äußern und der durchschnittlichen Anzahl der Beiträge pro Nutzer gebracht, was zum Beispiel relevant für das Stakeholder- und Reputationsmanagement ist.
- Anzahl der positiven und negativen Bewertungen über Unternehmen als Arbeitgeber pro Kanal bzw. Portal, wie zum Beispiel bei kununu.com oder glassdoor.de. Dieser KPI ist relevant bei der Messung der Mitarbeiterzufriedenheit.

An dieser Stelle ist es wichtig zu betonen, dass die Festlegung von Metriken und KPIs die Definition von Geschäftszielen sowie des Anwendungskontexts für den Einsatz von allgemeinem Monitoring von Meinungsäußerungen und inkl. der Sentiment-Analyse voraussetzt. Wenn wir beispielsweise die positiven Sentiments für eine Marke messen und dabei feststellen, dass diese sich im letzten Monat verdoppelt haben, ist die Relevanz und Aussagekraft dieser Messung maßgeblich vom festgelegten Unternehmensziel abhängig. Denn ist das Ziel die Leadgenerierung, so ist dies für die Vertriebs-Abteilung nur bedingt relevant und stellt keinen bedeutenden Erfolg dar. Ist jedoch das Unternehmensziel die Erhöhung der Marken-Wahrnehmung, so ist die Messung anhand der KPI dieser positiven Entwicklung relevant und aussagekräftig.

11.4 Zusammenfassung

Der kommerzielle Sentimentanalyse-Markt wächst kontinuierlich und wandelt sich dabei sehr dynamisch. Es gibt eine große Vielfalt hinsichtlich der Modelle der Bereitstellung der Lösungen und des Leistungsumfangs. Die Markterhebungen zu den Software-Lösungen können oft nur bedingt für die Auswahl von passenden Sentiment-Analyse-Lösungen herangezogen werden, denn häufig beruhen sie auf sehr spezifischen Auswahl- und Bewertungskriterien, die im eigenen konkreten Fall nicht geeignet sein müssen. Die Anzahl der Anbieter von Tools, die die Sentiment-Analyse deutschsprachiger Texte unterstützen, ist im Verhältnis zu den Angeboten für die englische Sprache zwar kleiner, es lässt sich jedoch ebenfalls eine Vielfalt von Modellen und Funktionen feststellen, die einen professionellen Einsatz im Unternehmen ermöglichen. Heutzutage wird Sentiment-Analyse innerhalb von Unternehmen verschiedener Branchen eingesetzt, oft innerhalb der Abteilungen der Marketing- und Unternehmenskommunikation, des Vertriebs, der Human Resources (HR) oder im Bereich Produktentwicklung. Dabei gibt es vielfältige Anwendungsfelder, die mithilfe der Sentiment-Analyse zum einen den Erfolg verschiedener Geschäftszielen messen und zum anderen Erkenntnisse für ihre Marktposition und Strategie sowie Informationen über ihre Zielgruppe gewinnen. Für die Erfolgsmessung dient der Polaritätswert der Sentiment-Analyse als Metrik für die Ermittlung von relevanten KPIs.

11.5 Übungen

1. Prüfen Sie Ihr Wissen:

- Welche Bereitstellungsmodelle der kommerziellen Sentiment-Analyse gibt es? Wie unterscheiden sie sich? Für welches Nutzungsszenario eignet sich welches Modell? Beschreiben Sie hierfür zwei Nutzungsszenarien.
- Wie erklären Sie, dass zum jetzigen Zeitpunkt hauptsächlich die Sentiment-Analyse auf Dokument-Ebene für kommerzielle Lösungen im Einsatz ist?

- Weshalb bedarf es der Festlegung der Geschäftsziele und Anwendungsfelder, bevor man ein Tool zur Sentiment-Analyse auswählt?
2. Setzen Sie Ihr neues Wissen ein:
- a) Testen Sie die Webdemo von ParallelDots (<https://www.paralleldots.com/text-analysis-apis>) anhand der Meinungsäußerungen, die Sie in der Übung zu Kap. 2 erstellt haben.
 - b) Vergleichen Sie die Ergebnisse mit Ihren Analysen aus Kap. 3.
 - c) Alternativ anstelle der Webdemo testen Sie das Excel-Plug-in und das Google-Plug-in von Paralleldots. Vergleichen Sie die Ergebnisse mit Ihren Analyse aus Kap. 3.
3. Reflexion in Gruppenarbeit:
- a) Wie bewerten sie die Ergebnisse? Welche Stärken und welche Schwächen stellen Sie bei der Analyse fest?
 - b) Stellen Sie sich vor, Sie sind in Ihrem (Groß-)Unternehmen zuständig für die Themen Mitarbeiterzufriedenheit und Mitarbeiterbindung und möchten durch das Monitoring und Sentiment-Analyse von Arbeitnehmer-Bewertungen zu Ihrem Unternehmen auf Bewertungsportalen, z. B. auf www.kununu.com analysieren und Erkenntnisse über die Zufriedenheit der Mitarbeiter zu gewinnen. Sie haben ein Budget von €5000 p.a. für eine Software-Lizenz, haben ein Team von 4 Personen von Informationswissenschaftlern, Entwicklern und Marketingexperten und haben insgesamt für die Aufgabe 6 Personenmonate zur Verfügung. Bearbeiten Sie folgende Fragen, begründen Sie dabei Ihre Positionen und stellen Sie das Ergebnis Ihrer Arbeit in einer Präsentation von 15 Min. vor:
 - Welche Kriterien (u. a. Technik, Leistung, Sprachabdeckung, Preismodell) legen Sie fest, um eine Entscheidung über die passende Lösung zu treffen?
 - Welche konkreten Ziele möchten Sie durch das Monitoring und die Analyse erreichen?
 - Welche Kennzahlen definieren Sie für die Erfolgsmessung?

11.6 Weiterführende Literatur

(Mohammad 2016) gibt einen umfangreichen Überblick über die Sentiment-Analyse inkl. der Analyse von Emotionen. Zum Thema Erfolgsmessung siehe (BVDW 2016) für eine Richtlinie zur Social-Media-Erfolgsmessung, die Sentiment-Analyse im Rahmen der KPIs für das Monitoring der Social-Media-Beiträge berücksichtigt.

Literatur

- Atalla, M., Scheel, C., de Luca, E. W., & Albayrak, S. (2011). Investigating the applicability of current machine-learning based subjectivity detection algorithms on German texts. In *Proceedings of Workshop on Robust Unsupervised and Semisupervised Methods in Natural Language Processing*, (S. 17–24).
- Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of LREC*, 10, 2200–2204.
- Banerjee, S., Chua, A. Y., & Kim, J.-J. (2015). Using supervised learning to classify authentic and fake online reviews. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, (S. 88). ACM.
- Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F. M. R., Rosso, P., & Sanguinetti, M. (2019). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, (S. 54–63).
- Benamara, F., Taboada, M., & Mathieu, Y. (2017). Evaluative language beyond bags of words: Linguistic insights and computational applications. *Computational Linguistics*, 43(1), 201–264.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. Beijing: O'Reilly Media, Inc.
- Bond, F., & Paik, K. (2012). A survey of WordNets and their licenses. *Small*, 8(4), 5.
- Bond, F., Vossen, P., McCrae, J. P., & Fellbaum, C. (2016). CILI: The collaborative interlingual index. In *Proceedings of the Global WordNet Conference*, (S. 2016).
- BVDW (2016). Erfolgsmessung in Social Media. Richtlinie zur Social-Media-Erfolgsmessung in Unternehmen des Bundesverbandes Digitale Wirtschaft (BVDW) e. V. https://www.bvdw.org/fileadmin/bvdw/upload/publikationen/social_media/Social_Media_Erfolgsmessung_2016.pdf. Zugriffen: 3. Dez. 2019.
- Carstensen, K.-U., Ebert, C., Jekat, S., Langer, H., & Klabunde, R. (2009). *Computerlinguistik und Sprachtechnologie: Eine Einführung*. Berlin: Springer.
- Chaturvedi, I., Cambria, E., Welsch, R. E., & Herrera, F. (2018). Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. *Information Fusion*, 44, 65–77.
- Clematide, S., & Klenner, M. (2010). Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, (S. 7–13).

- Conrady, R. (2015). Customer Reviews: kaufentscheidend, glaubwürdig, strategierelevant? Eine empirische Studie der ITB und der Fachhochschule Worms.
- Farias, D. H., & Rosso, P. (2017). Irony, sarcasm, and sentiment analysis. In F. A. Pozzi, E. Fersini, E. Messina, & B. Liu (Hrsg.), *Sentiment analysis in social networks, science direct e-books* (S. 113–128). Cambridge: Morgan Kaufmann.
- Fellbaum, C. (Hrsg.). (1998). *WordNet: An electronic lexical database*. Cambridge: MIT Press.
- Ferber, R. (2003). *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Heidelberg: dpunkt.
- Fersini, E., Rosso, P., & Anzovino, M. (2018). Overview of the task on automatic misogyny identification at IberEval 2018. In *IberEval@ SEPLN*, (S. 214–228).
- Forsa. (2018). Ergebnisbericht Hassrede. Auftraggeber: Landesanstalt für Medien Nordrhein-Westfalen (LfM). https://www.medienanstalt-nrw.de/fileadmin/user_upload/lfm-nrw/Foerderung/Forschung/Dateien_Forschung/forsaHate_Speech_2018_Ergebnisbericht_LFM_NRW.PDF. Zugegriffen: 11. Febr. 2020.
- Gitari, N. D., Zuping, Z., Damien, H., & Long, J. (2015). A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4), 215–230.
- Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73, 220–239.
- Heyer, G., Quasthoff, U., & Wittig, T. (2006). Text Mining: Wissensrohstoff Text. *W3l, Herdecke*, 18.
- Hooi, B., Shah, N., Beutel, A., Günnemann, S., Akoglu, L., Kumar, M., Makhija, D., & Faloutsos, C. (2016). Birdnest: Bayesian inference for ratings-fraud detection. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, (495–503). SIAM.
- Hövelmann, L., & Friedrich, C. M. (2017). Fasttext and gradient boosted trees at GermEval-2017 on relevance classification and document-level polarity. In [Wojatzki et al., 2017b], (S. 30–35).
- Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (S. 168–177). ACM.
- Ideya. (2018). Social media monitoring tools and services report excerpts. Analysis and elaborate profiles of more than 150 social technologies & services worldwide. Ideya market report, 9th edition. <http://www.ideya.eu.com/publications/social-media-monitoring-tools-and-services-report.html>. Zugegriffen: 1. Dez. 2019.
- Jindal, N., & Liu, B. (2006). Mining comparative sentences and relations. In *Proceedings of AAAI-06, the 21st National Conference on Artificial Intelligence*, (S. 1331–1336).
- Jindal, N., & Liu, B. (2008). Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, (S. 219–230). ACM.
- Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5), 73.
- Jungherr, A., Jürgens, P., & Schoen, H. (2012). Why the pirate party won the German election of 2009 or the trouble with predictions: A response to Tumasjan, A., Sprenger, T. O., Sander, P. G., & Welp, I. M. Predicting elections with twitter: What 140 characters reveal about political sentiment. *Social Science Computer Review*, (S. 229–234).
- Karoui, J., Benamara, F., & Moriceau, V. (2019). *Automatic detection of irony: Opinion mining in microblogs and social media*. London: Wiley.
- Kincaid, J. P., Fishburne, R. P., Jr., Rogers, R. L., & Chissom, B. S. (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Naval Technical Training Command Millington TN Research Branch: Technical report.

- Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018). Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics, (S. 1–11).
- Latendorf, A., Kohl, O., & Minarski, A. (2017). Warum hinkt Deutschland in der E-Mobilität hinterher? <https://m-result.com/social-media-research/e-mobilitaet-in-deutschland/>. Zugegriffen: 26. Nov. 2019.
- Lee, J.-U., Eger, S., Daxenberger, J., & Gurevych, I. (2017). UKP TU-DA at GermEval 2017: Deep learning for aspect based sentiment detection. In [Wojatzki et al., 2017b], (S. 22–29).
- Li, H., Chen, Z., Mukherjee, A., Liu, B., & Shao, J. (2015). Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *ICWSM*, (S. 634–637).
- Liu, B. (2007). *Web data mining: Exploring hyperlinks, contents, and usage data*. Berlin: Springer Science & Business Media.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 1–167.
- Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge: Cambridge University Press.
- Liu, B. (2017). Many facets of sentiment analysis. In E. Cambria (Hrsg.), *A practical guide to sentiment analysis* (Bd. 5, S. 11–39). Berlin: Springer.
- Liu, J., & Chien, A. (2018). The forrester wave: Social listening platforms Q3 2018. The 10 providers that matter most and how they stack up. www.forrester.com.
- Loria, S., Keen, P., Honnibal, M., Yankovsky, R., Karesh, D., Dempsey, E., et al. (2014). *TextBlob: Simplified text processing*. Secondary TextBlob: Simplified Text Processing.
- Market Research Future. (2019). Sentiment analytics market report. Global forecast 2023. <https://www.marketresearchfuture.com/reports/sentiment-analytics-market-4304>. Zugegriffen: 15. Nov. 2019.
- Maynard, D., & Funk, A. (2011). Automatic detection of political opinions in tweets. In *Extended Semantic Web Conference*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, (S. 3111–3119).
- Minh, D. L., Sadeghi-Niaraki, A., Huy, H. D., Min, K., & Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *IEEE Access*, 6, 55392–55404.
- Mishra, P., Mujadia, V., & Lanka, S. (2017). GermEval 2017 : Sequence based models for customer feedback analysis. In [Wojatzki et al., 2017b], (S. 36 – 42).
- Mishra, P., Tredici, M. D., Yannakoudakis, H., & Shutova, E. (2019). Abusive language detection with graph convolutional networks. *CoRR*, abs/1904.04073.
- Mishra, P., Yannakoudakis, H., & Shutova, E. (2019). Tackling online abuse: A survey of automated abuse detection methods. *arXiv e-prints*.
- Mohammad, S. M. (2016). Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement*, (S. 201–237). Elsevier.
- Mukherjee, A. (2015). Detecting deceptive opinion spam using linguistics, behavioral and statistical modeling. In *Proceedings of ACL-IJCNLP 2015*.
- Naderalvojud, B., Qasemizadeh, B., & Kallmeyer, L. (2017). HU-HHU at GermEval-2017 sub-task B: Lexicon-based deep learning for contextual sentiment analysis. In [Wojatzki et al., 2017b], (s. 18–21).
- Nagappan, M., & Shihab, E. (2016). Future trends in software engineering research for mobile apps. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, (S. 5, 21–32). IEEE.

- Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). SemEval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of SemEval-2016*, (S. 1–18).
- Nithyanand, R., Schaffner, B., & Gill, P. (2017). Online political discourse in the Trump era. *arXiv preprint. arXiv:1711.05303*
- Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, (S. 145–153). International World Wide Web Conferences Steering Committee.
- OpenText. (2019). OpenText Magellan for unstructured data. A fast, powerful, innovative way to find the value hidden in unstructured data, including documents and social media feeds. https://www.opentext.de/file_source/OpenText/en_US/PDF/opentext-magellan-variant-so-v1.pdf. Zugegriffen: 19. Nov. 2019.
- Ortega-Bueno, R., Rangel, F., Hernández Farias, D., Rosso, P., Montes-y Gómez, M., & Medina Pagola, J. E. (2019). Overview of the task on irony detection in Spanish variants. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019), co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2019)*. CEUR-WS.org.
- Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, (S. 309–319). Association for Computational Linguistics.
- Padilla Montani, J., & Schüller, P. (2018). TUWienKBS at GermEval 2018: German Abusive Tweet Detection. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018*.
- Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, (S. 271–278).
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *FNT in Information Retrieval* 2, 1–2, 1–135.
- Petz, G., Karpowicz, M., Fürschuß, H., Auinger, A., Střiteský, V., & Holzinger, A. (2014). Computational approaches for mining user's opinions on the web 2.0. *Information Processing & Management*, 50(6), 899–908.
- Pollard, C., & Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.
- Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Mohammad, A.-S., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., et al. (2016). SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, (S. 19–30).
- Posey, B. (2019). How to analyze text with amazon comprehend. <https://virtualizationreview.com/articles/2019/07/15/how-to-analyze-text-with-amazon-comprehend.aspx>. Zugegriffen: 15. Juli 2019, 3. Dez. 2019.
- Räbiger, S., Kazmi, M., Saygin, Y., Schüller, P., & Spiliopoulou, M. (2016). Stem at SemEval-2016 task 4: Applying active learning to improve sentiment classification. In *Proceedings of SemEval-2016*, (S. 64–70).
- Raschka, S. (2017). *Machine learning mit python*. Bonn: mitp Verlags GmbH.
- Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with python, scikit-learn, and tensorflow 2*. Birmingham: Packt Publishing Ltd.
- Ren, Y., & Ji, D. (2019). Learning to detect deceptive opinion spam: A survey. *IEEE Access*, 7, 42934–42945.

- Rill, S., Adolph, S., Drescher, J., Reinel, D., Scheidt, J., Schütz, O., Wogenstein, F., Zicari, R. V., & Korfiatis, N. (2012). A phrase-based opinion list for the German language. In J. Jancsary, (Hrsg.), *Proceedings of KONVENS 2012*, (S. 305–313.) ÖGAI. PATHOS 2012 workshop.
- Roosenbeek, J., & Salvador Palau, A. (2017). I read it on reddit: Exploring the role of online communities in the 2016 us elections news cycle. In G. L. Ciampaglia, A. Mashhadi, & T. Yasseri (Hrsg.), *SOCIAL INFORMATICS*, Bd. 10540 of *Lecture Notes in Computer Science*, (s. 192–220). SPRINGER INTERNATIONAL PU.
- Rosenthal, S., Farra, N., & Nakov, P. (2019). Semeval-2017 task 4: Sentiment analysis in Twitter. *arXiv preprint arXiv:1912.00741*.
- Ruppenhofer, J., Siegel, M., & Struß, J. M. (2020). *JLCL Special Issue on Offensive Language*, Bd. 1. GSCL – Gesellschaft für Sprachtechnologie und Computerlinguistik.
- Ruppenhofer, J., Siegel, M., & Wiegand, M. (2018a). Guidelines for IGGSA shared task on the identification of offensive language. ms.
- Ruppenhofer, J., Siegel, M., & Wiegand, M. (Hrsg.). (2018b). *Proceedings of the GermEval 2018 Workshop, Vienna*. Austrian Academy of Sciences: Austria.
- Salminen, J., Almerexhi, H., Milenković, M., Jung, S.-G., An, J., Kwak, H., & Jansen, B. J. (2018). Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In *Twelfth International AAAI Conference on Web and Social Media*.
- Sandulescu, V., & Ester, M. (2015). Detecting singleton review spammers using semantic similarity. In *Proceedings of the 24th International Conference on World Wide Web*, (S. 971–976). ACM.
- Schouten, K., & Frasincar, F. (2016). Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3), 813–830.
- Schulz, K., Mieskes, M., & Becker, C. (2017). h_da participation at GermEval subtask B: Document-level polarity. In [Wojatzki et al., 2017b], (S. 13–17).
- Shojaee, S., Azman, A., Murad, M., Sharef, N., & Sulaiman, N. (2015). A framework for fake review annotation. In *Proceedings of the 2015 17th UKSIM-AMSS International Conference on Modelling and Simulation*, (S. 153–158). IEEE Computer Society.
- Sidarenka, U. (2019). *Sentiment Analysis of German Twitter*. PhD thesis, Potsdam University, Dissertation eingereicht bei der Humanwissenschaftlichen Fakultät der Universität Potsdam.
- Siegel, M. (2020). *OdeNet*. Special Issue on Linking, Integrating and Extending Wordnets: Linguistic Issues in Language Technology.
- Siegel, M., Deuschle, J., Lenze, B., Petrovic, M., & Starker, S. (2017). Automatische Erkennung von politischen Trends mit Twitter - brauchen wir Meinungsumfragen noch? *Information – Wissenschaft & Praxis*, 68(1), 67–74.
- Siegel, M., & Drewer, P. (2012). *Terminologieextraktion – multilingual, semantisch und mehrfach verwendbar*. TEKOM-Frühjahrstagung: Tagungsband der. TEKOM-Frühjahrstagung.
- Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., & Klenner, M. (2019). Overview of GermEval task 2, 2019 shared task on the identification of offensive language. In *Proceedings of the GermEval 2019 Workshop*. Freie Universität Nürnberg.
- Sukthanker, R., Poria, S., Cambria, E., & Thirunavukarasu, R. (2018). Anaphora and coreference resolution: A review. *arXiv preprint arXiv:1805.11824*.
- Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting elections with Twitter: What 140 characters reveal about political sentiment. *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, (S. 178–185).
- van Aken, B., Risch, J., Krestel, R., & Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, (S. 33–42), Brussels. Association for Computational Linguistics.

- Van Hee, C., Lefever, E., & Hoste, V. (2018a). Exploring the fine-grained analysis and automatic detection of irony on twitter. *Language Resources and Evaluation*, 52(3), 707–731.
- Van Hee, C., Lefever, E., & Hoste, V. (2018b). SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, (S. 39–50).
- Vilares, D., Dovala, Y., Alonso, M. A., & Gómez-Rodríguez, C. (2016). LyS at SemEval-2016 task 4: Exploiting neural activation values for twitter sentiment classification and quantification. In *Proceedings of SemEval-2016*, (S. 79–84).
- Wang, G., Xie, S., Liu, B., & Yu, P. S. (2012). Identify online store review spammers via social review graph. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4), 61.
- Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30(3), 277–308.
- Wiegand, M., Ruppenhofer, J., Schmidt, A., & Greenberg, C. (2018a). Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (S. 1046–1056), New Orleans, Louisiana. Association for Computational Linguistics.
- Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018b). Overview of the GermEval 2018 shared task on the identification of offensive language. In *Proceedings of the GermEval. (2018). Workshop, Vienna*. Austrian Academy of Sciences: Austria.
- Wojatzki, M., Ruppert, E., Holschneider, S., Zesch, T., & Biemann, C. (2017a). GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. In [Wojatzki et al., 2017b], (S. 1–12).
- Wojatzki, M., Ruppert, E., Zesch, T., & Biemann, C. (Hrsg.). (2017b). *Proceedings of the GermEval 2017 – Shared Task on Aspect-based Sentiment in Social Media Customer Feedback, Berlin*. Germany: GSCL.
- Wolfgruber, M. (2015). *Sentiment Analyse mit lokalen Grammatiken*. PhD thesis, LMU München.
- Yadollahi, A., Shahraki, A. G., & Zaiane, O. R. (2017). Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, 50(2), 25.
- Ye, J., Kumar, S., & Akoglu, L. (2016). Temporal opinion spam detection by multivariate indicative signals. *ICWSM*, 743–746.
- Yu, H., & Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, (S. 129–136).
- Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *SemEval@NAACL-HLT*.
- Zaratsian, D., Osborne, M., & Plumley, J. (2013). Uncovering patterns in textual data with SAS Visual Analytics and SAS Text Analytics. In *Proceedings of the SAS®Global Forum 2013 Conference*, San Francisco.

Stichwortverzeichnis

A

Abkürzung, 51
Accuracy, 21
Adjektiv, 37, 41, 43, 51, 52, 54, 67, 87
Adverb, 52, 87
Amazon Customer Reviews Dataset, 5
AMI 2018, 96
Anapher, 65, 70
Annotation, 42, 63, 97, 98
Antonymie, 40
Artikel, 87

B

Bayesianisches Modell des Data Mining, 85
Bigram, 27
BOW (bag-of-words), 101
Business Intelligence, 104

C

Competitive Intelligence, 119
Crawler, 76, 77

D

Datenmodell der Domäne, 60
Datensatz, ausgewogener, 32
Debiasing, 97
Decision Tree, 99, 101
Deep Learning, 33, 45, 72, 73, 102
Defaming Spam, 82, 87
Dependenz-Parser, 57

Dependenzanalyse, 54, 67
Differenzanalyse, 90
Diversität, lexikalische, 87
Domäne, 108, 110, 115

E

Emoji, 18, 38, 52, 72
Emotionsanalyse, 107, 110, 112
Empfehlungsrate, 120
Entität, 12, 109, 112, 120
Erfolgsmessung, 103, 120, 122

F

F-Maß, 22
Fake Reviews, 82, 85–87
Fragesätze, 52

G

German Polarity Lexicon, 13
GermEval
2017, 5, 17, 21, 25, 37, 42, 59, 60, 66, 69, 71
2018, 94, 96, 102
2019, 96, 97, 102
Gold-Standard, 20–22, 84
Gradpartikel, 50, 52, 53, 72
Grenzwert, 99

H

Hashtag, 52, 72, 73, 75–77

Hasskommentare, 93
 Hassrede, 101
 HatEval, 96
 HPSG (Head-Driven Phrase Structure Grammar), 53, 57
 Hype-Spam, 82, 87
 Hyperlink, 18
 Hyponymie, 40

I

IGGSA, 38, 48
 Inter-Annotator Agreement, 84

K

Kaggle 2018, 95
 Klassifikation, binäre, 23, 98
 Kognitionsindikatoren, 87
 Kompositionalität, 52
 Konditionalsätze, 52
 Konfidenz, 109
 Konjunktion, 87
 Kontext, 37, 52
 Korpuserstellung, 92
 KPI (Key-Performance Indicator), 120–122
 Krisenmanagement, 119, 120
 Kundenloyalität, 120

L

Lemmatisierung, 37, 38, 43, 86

M

Machine Learning, 72, 84, 87, 99, 109
 Media Monitoring, 116
 Mehrwortlexeme, 37
 Meinungsforschung, 75, 77
 Merkmale, 27
 Meta-Daten, 83
 Metrik, 115, 116, 120, 121
 Min-Max-Skalierung, 24
 Mitarbeiterzufriedenheit, 120
 Modalverb, 51
 Monitoring, 104, 106, 116, 120–122
 Multi-Domain Sentiment Lexicon for German, 39

N

N-Grams, 101
 Negationen, 20, 21, 25, 29, 50, 52–54, 57
 Netz
 neuronale, 92
 semantisches, 110
 NLTK (Natural Language Toolkit), 13, 14
 Nomen, 87

O

OdeNet, 41, 42, 60, 63
 openthesaurus, 63
 Opinion Lexicon, 41

P

Pandas, 30, 91, 99
 Part-of-Speech-Tagger, 43
 Pickle, 30
 PMI (Pointwise Mutual Information), 44
 Polarität, 107, 109–112, 115, 116, 120, 121
 Polarity Lexicon, 39
 Präposition, 87
 Precision, 21, 83
 Pronomen, 51, 65, 87

Q

Quintupel, 12, 14, 82

R

Recall, 21, 83
 Regression, 23
 Reputationsmanagement, 120

S

Sachinformation, 6, 7
 Sarkasmus, 112
 SemEval, 23, 70, 72, 96
 Sentiment-Index, 120
 Sentiment-Ratio, 120
 SePL (Sentiment Phrase List), 39
 Sklearn, 30, 91
 Skopus, 53, 54
 Social Listening, 104, 106, 116
 Social Media Monitoring, 116
 spaCy, 38, 43, 51, 54, 57, 67

Sprachmodell, probabilistisches , [25](#)
Stakeholder- und Reputationsmanagement, [120](#)
Subjectivity Detection, [7](#)
Subjektivität, [112](#)
Supervised Learning, [25](#), [27](#), [51](#)
SVM (Support Vector Machines), [101](#)
SWN, [41](#)
Synonym, [23](#), [40–42](#), [63](#), [86](#)
Synset, [40](#), [41](#)

T

Target, [62–64](#), [67](#)
Taxonomie, [65](#), [110](#), [114](#)
Terminologie-Extraktion, [60](#), [65](#)
TextBlob, [13](#), [14](#), [23](#), [38](#), [43](#)
Textdaten, unstrukturierte, [5](#), [105](#), [110](#), [115](#)
Textnormalisierung, [37](#)
TF-IDF, [101](#)
Tokenisierung, [18](#), [43](#), [50](#), [57](#)
Tonalität, [119](#)

TRAC 2018, [95](#)
Trigram, [26](#), [27](#)
Twitter, [76](#), [77](#)
TwitterSearch, [76](#)

U

UGC (User Generated Content), [107](#)
urllib, [76](#)

V

Verb, [87](#)
Verstärker, [29](#)
Vorverarbeitung, [18](#)

W

Word Embeddings, [33](#), [44](#), [48](#), [64](#)
word2vec, [45](#)
WordNet, [40](#), [41](#), [48](#), [86](#)
Wortlisten, [20](#), [37](#), [42](#), [44](#), [98](#), [100](#)