technische universität
dortmund

Master-Thesis:

Aspektbasierte Stimmungsanalyse anhand von Produktbewertungen

Kristof Wilke
Februar 2019

Erstgutachter:
Prof. Dr. Katharina Morik
Zweitgutachter:
Lukas Pfahler (M.Sc.)

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl für Künstliche Intelligenz, LS VIII
https://www-ai.cs.uni-dortmund.de

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Aspekt-basierte Stimmungsanalyse (engl. aspect based sentiment analysis) ist eines der Kernthemen in der Stimmungsanalyse (engl. sentiment analysis) von Produktbewertungen. Für Konsumenten sind meist nur bestimmte Aspekte eines Produktes relevant. Diese können zum Beispiel, -im Falle einer Kamera, -die Bildqualität, Akkulaufzeit oder Bedienbarkeit sein. Häufig werden diese Erfahrungsberichte von Kunden eines Produktes im Internet veröffentlicht, worin diese ihre Zufriedenheit oder Unzufriedenheit mit dem Produkt sowie seinen Aspekten kundtun und ihre Erfahrungen teilen. Diese Menge an wertvollen Informationen steht normalerweise nur ungeordnet zur Verfügung. Es werden lediglich Sternebewertungen bezüglich der Gesamtzufriedenheit abgegeben. Für den Konsumenten lässt sich kein Überblick über bestimmte Aspekte gewinnen, ohne eine große Anzahl von Bewertungen lesen zu müssen oder zu riskieren, sich auf zu wenige Meinungen zu verlassen und allzu subjektiven Bewertungen Vertrauen zu schenken. Deswegen ist es für den Konsumenten von großem Vorteil, eine automatisierte aspekt-basierte Stimmungsanalyse von Produktbewertungen zu erhalten, um so eine bessere Entscheidung beim Kauf treffen zu können. Aus diesem Grund befassen sich viele Arbeiten aus dem Bereich des maschinellen Lernens mit dieser Aufgabe und bringen das Thema voran. Grundsätzlich ist Sprache sehr komplex und es ist schwierig, alle Aspekte und Informationen von einer Maschine richtig analysieren zu lassen. Produktbewertungen liegen in einer unstrukturierten Form vor und sind sehr unterschiedlich in ihrer Ausdrucksweise. Dadurch entsteht eine Reihe von Herausforderungen, wie beispielsweise der Umgang mit Synonymen, Sarkasmus, Komparativen oder verb- und nomenbasierten Stimmungen, um nur einige zu nennen (Liu, 2015).

1.2 Ziel der Arbeit / Problemstellung

Das Ziel dieser Master-Thesis ist die Implementierung eines aspekt-basierten *Sentiment Summarizers*, dieses Framwork bewertet anhand von Rezensionstexten einzelne Aspekte. Der *Summarizer* berechnet Aspekt-Ratings, welche die Stimmung der einzelnen Aspekte in den Rezensionstexten repräsentieren. Dieses Problem wurde von Hu und Liu (2004) als *aspect-based sentiment anaylsis* definiert. Im Gegensatz dazu ist das Ziel dieser Arbeit, dieses nicht anhand eines binären Klassifikationsproblems zu lösen, sondern weiterführende Vorhersagemodelle, wie zum Beispiel Regressionsmodelle, zu definieren und zu evaluieren, um aspekt-basierte Ratings zu berechnen. Die Schwierigkeit liegt nicht nur in der Auswertung der Satzteile mittels Stimmungsanalysen, sondern auch in der Identifizierung und Zuordnung der Aspekte und der zugehörigen Stimmungen. Stimmungsanalysen von einzelnen Dokumenten oder Satzteile sind in der Vergangenheit gut erforscht und praktiziert worden, dagegen stellt die aspekt-basierte Stimmungsanalyse die aktuelle Forschung gerade im unüberwachten Lernen vor Probleme (Liu, 2015).
Ausgangspunkt sind Webseiten, auf denen Käufer von Produkten oder Dienstleistungen Bewertungen hinterlassen können. Autoren solcher Rezensionen schreiben in der Regel einen Bewertungstext und geben zusätzlich eine Sternebewertung bezüglich ihrer Gesamtzufriedenheit ab. Diese Sternebewertungen liegen im Internet in großer Zahl vor und die Menge wächst stetig weiter. Die Daten sind mittels eines „Crawlers„ einfach zu beschaffen und können in großen Mengen genutzt werden. Wenn es nötig ist, kann man die Daten auch von mehreren Seiten nutzen und so die Datenlage verbessern. Ein Ziel dieser Arbeit ist, einen generischen Ansatz zu finden, da der Domainexperte die Kategorien kennt und kann dieser mit einem halb-überwachten Ansatz die Aspekte gut steuern. Da sich die Datenlage gut verbessern kann und sich die Algorithmen auf nahezu alle Kategorien gut übertragen lassen und keine gelabelten Daten für alle Kategorien zur Verfügung stehen, werden Lösungen gesucht, die sich bei der Aspekt-Clusterung auf unüberwachte beziehungsweise halb-überwachte Lernmethoden beschränkt.
In der Regel erzielt man mit dem überwachten Lernen bessere Ergebnisse. Das Erstellen von gelabelten Daten, die aspekt-spezifisch sind, ist allerdings sehr zeitintensiv und daher kostspielig. Außerdem sind diese nur für bestimmte Probleme und Kategorien und meist auch nur in sehr geringen Mengen für die Forschung zugänglich. Aus diesen Gründen liegt der Fokus auf Algorithmen, die mit wenigen Startwörtern oder völlig ohne gelabelte Daten auskommen. Kategorien sind in diesem Fall Produktgruppen, wie zum Beispiel Fernseher, Kameras oder auch Hotels.

Zusammengefasst stellt sich diese Arbeit folgende Frage: »*Ist es möglich, aspekt-basierte Ratings anhand von automatisierten Vorhersagemodelle zu erlernen, welche Rezensionstexte und Gesamtratings verwenden, um Produkte mittels aspekt-spezifischer Ratings detaillierter vergleichen zu können?*«

1.3 Aufbau der Arbeit

Die Arbeit beginnt mit den Schritten Datenbeschaffung und Datenvorbereitung, welche in Kapitel 2 erläutert werden. In diesem Kapitel wird beschrieben, wie die Daten für das Experiment beschafft wurden und welche Grundlage die Daten für das Experiment bieten. Im weiteren Verlauf des Kapitels werden die gängigsten Vorbereitungsschritte für die automatisierte Textverarbeitung erläutert, welche eine geeignete Auswahl für das Experiment offerieren.
Eine Vorgehensweise der aspekt-basierten Stimmmungsanalyse (engl. Aspect-Based Sentiment Analysis), wie sie in dieser Arbeit implementiert wird, ist folgende:

1. Crawlen der Produktbewertungen,
2. Datenvorbereitung,
3. Aspekt-Clusterung,
4. Aspekt-Segmentierung,
5. Vorhersagemodelle,
6. Evaluation der Ergebnisse.

Dies sind die wichtigsten Schritte für den praktischen Teil und ziehen sich als roter Faden durch diese Arbeit.
In den Kapiteln 3 und 4 werden die theoretischen Grundlagen für die Schritte Aspekt-Clusterung und Vorhersagemodelle gelegt, damit in dem darauffolgenden Kapitel 5 die Theorie in die Praxis übergehen kann. In dem Kapitel 6 werden die Thesen aus den vorherigen Kapiteln überprüft und ausgewertet, indem ein Experiment vorgestellt wird, welches Ergebnisse mittels Metriken evaluiert und vergleichbar macht. Am Ende wird in einem Fazit Stellung zu den Auswertungen genommen. Der Ausblick beschreibt weitere mögliche Lösungsansätze oder auch bestehende Schwierigkeiten, die im Laufe dieser Arbeit erkannt wurden und die somit Raum für anknüpfende Arbeiten bieten.

Kapitel 2

Datengrundlage

2.1 Auswahl und Spezifikation der Daten

Für das Experiment werden Daten in Form von Bewertungen verwendet, die sich kategorisieren lassen und in großer Menge auf diversen Webseiten zu Verfügung stehen und täglich enorm wachsen. Die Kategorien sind zum Beispiel Produkte, Hotels, Events oder auch Restaurants. Die Einordnung in diese Kategorien ist nicht automatisiert, wird aber beim Crawlen der Daten oder bei der Verwendung der Daten beachtet und ist die erste Gruppierung für die verwendeten Algorithmen. Die Algorithmen können kategoriespezifisch lernen und führen so zu aspketspezifschen Ergebnissen, welche für das Experiment wichtig sind. Eine wichtige Beobachtung der Daten ist die Gesamtbewertung, die meist als Sternebewertung auf Webseiten anzeigt wird und zusätzlich als numerische Zahl vorliegt. Der Autor einer Bewertung fasst seine Zufriedenheit am Ende seiner Bewertung in einer Zahl zwischen 1 und 5 zusammen. Diese Information wird als Label für die Vorhersagemodelle verwendet und ist neben dem Text die Grundlage für das Experiment dieser Master-Thesis. Bei *amazon.com* und den meisten anderen Bewertungsseiten liegen die Rezensionen in einer unstrukturierten Textform vor und bieten keinerlei Übersicht über die Aspekte. Die Länge der Rezensionen variiert stark und nicht immer sind alle Aspekte vom Verfasser der Bewertung kommentiert. Die Anzahl der Kommentare wächst aber stetig und liegt gerade im Englischen in „ausreichender“ Menge vor. Auf der Seite von Google-Shooping sind mehr als 30 Seiten zusammengetragen und könnte eine weitere Datengrundlage für diese Pipeline sein.
In dieser Master-Thesis werden die Ergebnisse quantitativ evaluiert. Die quantitative Evaluation erfolgt unter Verwendung des englischen Tripadvisor-Datensatzes von Wang[1]. Dieser Datensatz enthält zusätzlich zu dem Bewertungstext und der Gesamtrating noch bis zu neun weitere Aspekt-Ratings. Diese Sternebewertungen bewerten die verschiedenen Aspekte eines Hotels oder Restaurants.

[1] http://times.cs.uiuc.edu/ wang296/Data/

Die Daten werden beim Crawlen in einer Json gespeichert und enthalten den Rezensionstext b, das Erstellungsdatum t, die Gesamtrating r und den Autor a, womit man das Quintupel $d = (b, t, r, a)$ für jede Rezension d erhält. Bei dem Tripadvisor-Datensatz enthält man die Kategorie Hotel zu den Aspekten *Value, Room, Location, Cleanliness, Service Check In/Front Desk, Business Service, Sleep Quality* ein Aspekt-Rating s von 1 bis 5 gehören.

2.2 Textvorverarbeitung

In dem Kapitel *Textvorbereitung (engl. Preprocessing)* wird erläutert, welche Vorbereitungsschritte sinnvoll angewendet werden können, um aus den Rohdaten Informationen zu gewinnen. Immer wenn man von Textdaten lernen möchte, sind spezielle Textvorverarbeitungsschritte sinnvoll, um die Güte des Lernens zu verbessern. Es ist unerlässlich eine Textpräsentationen zu wählen, damit mit einer Maschine Informationen gewonnen werden können. Die Textvorbereitungsschritte, die in dieser Arbeit Verwendung finden, lassen sich in zwei Gruppen aufteilen. Die Gruppe *Stoppwörter entfernen*, *Buchstaben kleinschreiben Wortstammbildung* und *Lemmatisierung* haben alle den Zweck die Wortanzahl (Wortcorpus) zu verkleinern, um effizienter lernen zu können. Das *Britisch Nation Corpus*[2] trägt beispielsweise 100 Millionen Wörter zusammen. Wenn nun die Verbkonjugationen, Kasus und Numerus erschwerend hinzu kommen, dann potenziert sich die Anzahl der Wörter um einiges mehr. Nachdem alle Wörter für die weitere Verarbeitung kleingeschrieben sind, können die *Wortstammbildung* und die *Lemmatisierung* verwendet werden. Der Vorgang wird in dem folgenden Beispiel erläutert:

2.2.1 Beispiel. Lemmatisierung: went → go
Wortstammbildung: runs → run
Lemmatisierung: heard → hear
Wortstammbildung: logged → log

Ein wichtiger Vorbereitungsschritt ist die Anwendung des *Bag-of-Words*-Modells, welches die Texte für die Maschine geeignet transformiert. Die einfachste Form des BoW-Modells ist das Aufsummieren gleicher Wörter und die Zuweisung von IDs. Hierzu ein Beispiel:

2.2.2 Beispiel. It is very big and very beautiful. → 0:1 1:1 2:2 3:1 4:1 5:1

An diesem Beispiel ist zu erkennen, dass auf der linken Seite des Doppelpunktes die ID eines Worts steht und rechts die Anzahl der vorkommenden Wörter. Es fällt auf, dass die ID 2 zweimal vorkommt, was dem Wort „very“ entspricht. Diese Transformation ist die Ausgangslage für alle Modelle, die in dieser Arbeit vorkommen.

[2] http://www.natcorp.ox.ac.uk/

2.2.1 Stoppwörter entfernen

Als Stoppwörter bezeichnet man üblicherweise die Wörter, die im Sprachgebrauch sehr häufig auftreten, aber für den Textinhalt in den meisten Fällen keine beziehungsweise wenig Relevanz haben. Stoppwörter übernehmen grammatikalische oder syntaktische Funktionen im Text und tragen deswegen nur wenig zum Textinhalt bei (Vijayarani et al., 2015). Allgemein übliche Stoppwörter im Englischen sind Konjunktionen, Artikel, Präpositionen, und die meisten Stoppwörterlisten enthalten auch Negationen. Englische Wörter wie „and“, „the“ und „a“ beinhalten weder Information zu einem Aspekt, noch enthalten sie relevante Stimmungen eines Aspekt. Hingegen sind Negationen, wie zum Beispiel „not“, „hasn't“ usw., sehr kritisch zu sehen diese zu entfernen, da sie bei Klassifizierungsverfahren zu einem Klassenwechsel führen und bei Regressionsmodellen ein Vorzeichenwechsel beachtet werden muss. Der Satz „I didn't like the food“ wird durch die Stoppwordliste zu „like food“ gefiltert. Diese Art der Filterung führt zu einer kompletten komplementären Aussage und führt zu Fehlern. Wie damit umgangen wird, hängt von den weiteren Transformationschritten ab. Mit Hilfe von *Part-of-Speech Tagging* können diese Wörter schnell identifizieren werden und man kann den Vorzeichenwechsel beachten.
Wenn man bei einem reinen syntaktischen Auswertungsverfahren, wie zum Beispiel Liu et al. (2016) es umsetzten, dann kann dieses am Ende der Pipline gut implementiert werden. Eine Beispielimplementierung befindet sich auf der CD. Wenn aber zum Beispiel eine Transformation mit *Bag-Of-Words* folgt, kann man das mithilfe von N-Gramm-Modellen beheben, indem man die Wörter nicht entfernt und das N-Gramm-Modell die Wörter zusammenführt. In dem Beispielsatz von eben werden die Wörter „didn't_ like“ zusammengeführt und werden als ein Wort von der Maschine angesehen. Mit der Anwendung eines N-Gramm-Modells einsteht der Satzteil: „didn't_ like food“, womit mit die *Bag-Of-Word*-Repräsentation ein eigenes Wort erkennen kann und dieses gesondert beim Lernen ausgewertet werden kann. Die genaue verwendetete Stoppwortliste befindet sich auf der CD. Als Grundlage wird die *Onix Text Retrieval Toolkit*[3] verwendet und mit mit ein paar Wörtern für die Kategorie Hotel angereichert.

2.3 Kreuzvalidierung

Beim Modelltest trennt man in der Regel den Datensatz in einen Trainings- und Testdatensatz. Die Kreuzvalidierung ist ein Verfahren Störungen in den Daten auszugleichen. Die allgemein bekanntesten Verfahren sind die Leave-One-Out-Kreuzvalidierung, k-fache stratifizierte Kreuzvalidierung und die einfache Kreuzvalidierung. Die Vorgehensweise wird in Abbildung 2.1 anhand einer 5-fachen Kreuzvalidierung mit fünf Iterationen dargestellt. Sie kann also ergänzend zu den meisten Metriken sein und so zu aussagekräftigen Messungen

[3] http://www.lextek.com/manuals/onix/stopwords1.html

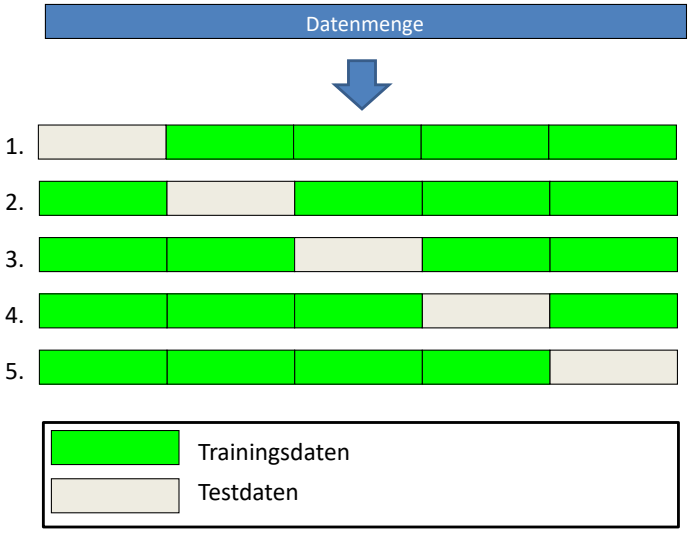


Abbildung 2.1: 5-fache Kreuzvalidierung

führen. Ein weiterer Grund für die Verwendung der Kreuzvalidierung in dieser Evaluation ist, dass sie auch eine Überanpassung vermeidet. Mit Überanpassung ist gemeint, dass ein Modell gelernt wird, welches sich im Extremfallan jeden Datenpunkt anpasst. Dieses geschieht durch zu viel Training. Jede Teilmenge, die sich vom Trainingssatz unterscheidet, wird in der Regel zu einem größeren Fehler führen (Kohavi und others, 1995). Die Kreuzvalidierung meist längere Laufzeiten benötigt, kann es sinnvoll sein, die Kreuzvalidierung zu verwenden. Besonders wenn der Datensatz nicht so ergiebig ist, kann die Kreuzvalidierung zu aussagekräftigen Ergebnissen führen. Bei der Modellverbesserung und der damit verbundenen Hyperparameteranpassung ist die Kreuzvalidierung eines der bekanntesten Verfahren (Sarkar et al., 2017) und findet auch in Kapitel 5 Verwendung.

Kapitel 3

Aspekt-Segementierung

Längere Kundenrezensionen sind in der Regel so geschrieben, dass die verschiedenen Aspekte erläutert und bewertet werden. Diese Informationen gilt es zu nutzen und zu extrahieren. Wenn man sich die Rezensionstexte genauer anschaut, erkennt man, dass die Autoren von Rezensionen die meisten Sätze oder auch Satzteile mit einer bewertenden Absicht geschrieben sind. Es handelt sich dabei meistens, um ein Gesamtresümee oder von bestimmte Themenbereiche, die in dieser Arbeit als Aspekte bezeichnet sind. Maschinelle Auswertungen über die Gesamtzufriedenheit lassen sich auf Grund des Gesamtratings recht einfach durchführen und wird auf den meisten Bewertungsseiten schon praktiziert. Aufgrund dessen liegt der Fokus dieser Arbeit auf der aspekt-basieren Auswertung. Liu et al. (2005) zeigen in ihrer Forschung, dass sich diese Auswertung auch kombinieren lässt, indem ein Cluster *Allgemein* gebildet und erlernt wird.Die grundlegende Idee der Aspekt-Segmentierung wird anhand der Abbildung 3.1 erläutert. Autoren von Rezensionen gewichten diese Aspekte, unter anderem mithilfe von Adjektiven, unterschiedlich stark. Dieses wird auch oft mit Stimmung oder Sentiment bezeichnet. Die zu erlernende Gewichte für jedes einzelne Wort wird im nächsten Schritt als eigenständiges Thema in 4 behandelt. Das Ziel der Aspekt-Segmentierung ist, dass man am Ende eine Zuordnung von den einzelnen Satzteilen zu den verschiedenen Aspekten erhält, welches in Abbildung 3.1 visualisiert ist. An diesem Beispiel erkennt man, dass der Autor in dieser Rezension hauptsächlich die Aspekte *Location*, *Service* und *Room* bewertet. Das Ziel ist es diese Aspekte zu finden und auswerten. Eine mögliche Vorgehensweise ist, dass man zuvor die Aspekte durch clustering identifiziert. Das Ergebnis der Clusterung sind dann aspekt-spezifsche Wortlisten, mit denen man im nächsten Schritt eine Aspekt-Segmentierung durchführen kann. In dem Schritt der Aspekt-Segmentierung unterteilt man den Text in Satzteile und ordnet diese Satzteile mithilfe dieser Wortlisten den Aspekten zu. In der Literatur werden gerade im unüberwachten Bereich diese Aspekt-Cluster als *Topics* bezeichnet.

Quelle: Wang et al. (2010)

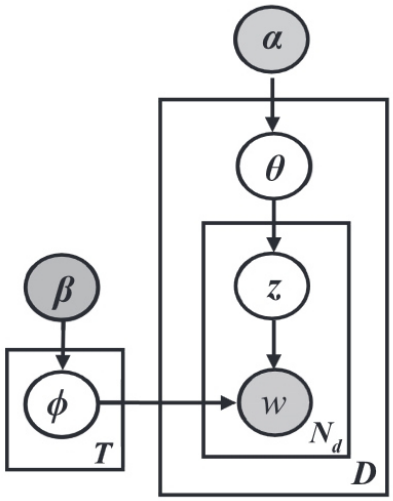
Abbildung 3.1: Aspektsegmentierung

3.1 Unüberwachte Algorithmen

Im Detail unterscheiden sich die *Topic Models* dadurch, dass sie auf verschiedenen statistischen Annahmen beruhen. Es ist ein großer Vorteil von unüberwachten *Topic Models*, dass sie keine gelabelten Daten benötigen und am Anfang auch noch kein Expertenwissen, um den Text in verschiedene Topics einzuteilen. Der Nachteil besteht darin, dass man nur wenig Kontrolle über das Ergebnis hat. Der Output enthält dann die verschiedenen Topics, die von einem Experten interpretiert werden müssen. Im Idealfall kann man diesen Vorgang automatisieren. Es ist möglich, dass der Algorithmus zwei Topics clustert, die im Ergebnis optimalerweise individuell geclustert werden sollten, da die Auswertung so für den Konsumenten aussagekräftiger ist.

3.1.1 Latent Dirichlet Allocation (LDA)

Eine der im unüberwachten Lernen bekanntesten Lösungen zum Herausfinden von verschiedenen Topics in Texten ist die Latent Dirichlet Allocation (LDA) von Blei et al. (2003). Die LDA ist ein *Topic Model*, welches ein generatives probabilistisches Modell für diskrete Daten, wie zum Beispiel Texte, ist. Außerdem ist es ein dreistufiges hierarchisches bayesisches Modell, welches versteckte Semantiken im Text entdecken kann, in diesem Fall die Topics eines Produktes. Im Idealfall entsprechen die Topics den verschiedenen Aspekten und den Stimmungswörtern eines Produktes. Wie die meisten Topic-Modelle kann die LDA graphisch dargestellt werden (3.2). Es basiert auf dem bayes'schen Modell, wobei



Quelle: Wang et al. (2010)

Abbildung 3.2: LDA Plate-Notation

die Eingabe in der LDA ein Corpus ist, das aus einem Set von Dokumenten D besteht. Der Output der LDA ist eine Verteilung über Topics für jedes Dokument. Diese Verteilung wird auch Welt-Topic-Verteilung ϕ genannt. Es wird angenommen, dass Θ and ϕ multinomiale Verteilungen sind. Um die Verteilung zu glätten wird angenommen, dass der Dirichlet-Prior die Hyperparamter α und β hat. Das Wort w ist die beobachtete Variable. Die Anzahl der Topics sind indexiert als $1, ..., T$ und die Einträge des Vokabulars eines Corpus werden durch $1, ..., V$ definiert, wobei V die Anzahl der einzelnen Wörter im gesamten Dokumenten-Corpus ist (Liu, 2015).
Das Corpus hat D Dokumente und jedes Dokument d ist eine Folge von N_d Wörtern. w ist das *Bag-of-Words* von allen beobachten Wörtern mit Kardinalität,

$$|w| = \sum_{d} N_d z. \tag{3.1}$$

z bezeichnet die Zuweisung der Topics von allen Wörtern, die in einem Dokument sind. Die graphische Darstellung in der Plate-Notation ist in Abbildung 3.2 dargestellt, wobei *Theta*, ϕ und z latente Variablen sind. Die Dirichlet-Hyperparameter α und β gelten als Konstanten und werden daher ebenfalls beobachtet Liu (2015).

3.2 Bootstrapping-Algorithmen

Bootstrapping-Algorithmen sind auf sogenannte Startwörter angewiesen. Am Anfang erstellt der Experte eine Reihe von kleinen Wortlisten, welche die verschiedenen Aspekte der Kategorie repräsentieren. Da sich die Aspekte für jede Kategoriegruppe unterscheiden, sind diese Startwörter für jede Domain einmalig anzufertigen. Der große Vorteil, der sich dar-

aus ergibt, ist, dass man die Kontrolle über die Anzahl der verschiedenen Aspekte behält und eine direkte Zuordnung von den zuvor formulierten Aspekten mit den Startwörtern zu den Wortlisten erhält. Diese resultiert direkt aus dem Mechanismus des Bootstrapping-Algorithmus', da dieser die Startwörter mit den am nächsten korrespondierenden Wörtern anreichert. Bootstrapping-Algorithmen erstellen also keine komplett neuen Wortlisten, stattdessen verbessern sich die Wortlisten bei jeder Iteration bis zu einem im Input definierten Schwellenwert. Für diese Berechnung verwendet man das komplette Corpus ohne die Unterteilung auf Satz- oder Bewertungsebene. Bootstrapping-Algorithmen werden im maschinellen Lernen den halb-überwachten Verfahren zugeordnet.

3.2.1 Bootstrapping mit Chi-Quadrat-Test

Als Input für den Algorithmus benötigt man, wie in 3.2 beschrieben, Startwörter für jeden Aspekt und die Textinhalte der einzelnen Bewertungen. Für die Abbruchbedingung kann man entweder einen Schwellenwert oder ein Limit für den maximale Iteration festlegen. Der Algorithmus 1 der den Vorgang der Aspekt-Segmentierung und Aspekt-Clusterung mit Chi-Quadrat-Test beschreibt, ist von Wang et al. (2010) als Pseudocode folgendermaßen formuliert:

Im ersten Schritt unterteilt man die Rezensionen in Sätze beziehungsweise Satzteile. Für

Algorithm 1 Aspect Segmentation Algorithm

Input: A collection of reviews $\{d_1, d_2, ..., d_{|D|}\}$, set of aspect keywords $\{T_1, T_2, ...T_k\}$, vocabulary V, selection threshold p and iteration step limit I

Output: Reviews split into sentences with aspect assignments

1: Split all reviews into sentences, $X = \{x_1, x_2, ...x_M\}$
2: Match the aspect keywords in each sentence of X and record the matching hits for each aspect i in $Count(i)$
3: Assign the sentence an aspect label by $\arg\max_i Count(i)$. If there is a tie, assign the sentence with multiple aspects
4: Calculate χ^2 measure of each word (in V)
5: Rank the words under each aspect with respect to their χ^2 value and join the top p words for each aspect into their corresponding aspect keyword list T_i
6: If the aspect keyword list is unchanged or iteration exceeds I, go to line 7, else go to line 1
7: Output the annotated sentences with aspect assignments

diese Zerlegung wird das Apache *OpenNLP*[1] Framework Apache verwendet, welches eines der bekanntesten Framworks für Textverarbeiten im Java Umfeld ist. Im nächsten Schritt zählt man die Anzahl der vorkommenden Aspekte in den zuvor bestimmten Satzteilen und

[1] https://opennlp.apache.org/

ordnet diese dem Aspekt mit der höchsten Übereinstimmung zu. In der ersten Iteration benötigt man für diese Zuordnungsüberprüfung die im Input definierten Startwörter. Wenn die Anzahl der vorkommenden Aspekte gleich ist, wird auch der Satzteil mehreren Aspekten zugeordnet. Dies kommt in der Regel aber selten vor, da sich in späteren Iterationensstufen klares Maximum bildet. Im nächsten Schritt berechnet man mit der Formel

$$\chi^2(w, A_i) = \frac{C * (C_1 C_4 - C_2 C_3)^2}{(C_1 + C_3) * (C_2 + C_4) * (C_1 + C_2) * (C_3 + C_4)}. \tag{3.2}$$

die Beziehung von jedem Wort w zu den Aspekten A_i. Dabei ist C die Gesamtanzahl der vorkommenden Wörter. C_1 ist die Anzahl, mit der w in den Satzteilen auftritt, die zu Aspekt A_i gehören. C_2 entspricht der Anzahl, mit der w in den Satzteilen auftritt, die nicht zu dem Aspekt A_i gehören. C_3 ist die Anzahl der Satzteile von Aspekt A_i, die w nicht enthalten sind und C_4 entspricht der Anzahl der Satzteile, die weder zu Aspekt A_i gehören, noch enthalten w. C ist die gesamte Wortanzahl die vorkommen. Mithilfe diesem χ^2-Wert lassen sich die Wörter in verschiedene Aspekte clustern und Rankings erstellen (Yang und Pedersen, 1997). Als Ausgabe des Algorithmus' erhält man die Aspekt-Listen und die Satzteile l, die den Aspekten i zugeordnet sind. Für die Modelle in Kapitel 5, die die Aspekt-Ratings schätzen, benötigt man eine Worthäufigkeitsmatrix w_{dij}, wobei d die Rezensionen und j die einzelnen Wörter sind. Diese Matrix sind für die Modelle später die Features.

Kapitel 4

Vorhersagemodelle

Die Probleme, die mithilfe von Vorhersagemodelle gelöst werden, besitzen eine wesentliche Gemeinsamkeit: Eigenschaften einer Zielvariablen y sollen in Abhängigkeit von Features $x_1, ...x_k$ ausgedrückt werden. Meist werden die Zielvariablen auch als abhängige Variablen und die Features als erklärende Variablen bezeichnet (Fahrmeir et al., 2018). Die Features werden in diesem Fall aus den extrahierten Aspekt-Segmenten abgeleitet. In diesem Kapitel werden die theoretischen Grundlagen der lineare Regression (4.1) und logistischen Regression (4.2) geben, um in nächsten Schritt die Modelle zu konstruieren. Ein wichtiger Unterschied zwischen der linearen Regression und der logistischen Regression ist, dass bei einer linearen Regression die Zielvariablen stetig sind. Dagegen sind bei der logistischen Regression die Zielvariablen kategorial oder auch binär. Logistische Regression gehört zu den Klassifikationsverfahren. Ein bestimmendes Charakteristikum von Vorhersagemodelle ist, dass der Zusammenhang zwischen der Zielgröße y und den Features nicht exakt als Funktion $f(x_1, ..., x_k) = x_1, .., x_k$ definiert ist, sondern durch zufällige Ausreißer überlagert wird. Die Zielgröße y hängt von den erklärenden Variablen ab und lässt sich als bedingter Erwartungswert $E(y|x_1, ..., x_k)$ von y in Abhängigkeit zu den Features beschreiben. Der Erwartungswert ist also eine Funktion der erklärenden Variablen:

$$E(y|x_1, ..., x_k) = f(x_1, ..., x_k) \tag{4.1}$$

Die Zielgröße lässt sich dann immer zerlegen in

$$y = E(y|x_1, ..., x_k) + \epsilon = f(x_1, ..., x_k) + \epsilon, \tag{4.2}$$

wobei ϵ der zufällige Fehlerterm ist. Das Ziel einer Regressionsanalyse ist, die systematische Komponente f aus den gegebenen Daten $y_i, x_{i1}, ..., x_{ik}$ zu schätzen und von dem Fehlerterm ϵ zu isolieren (Fahrmeir et al., 2018).

4.1 Lineare Regression

Eine der bekanntesten statistischen Lernmethoden ist das lineare Regressionsmodell

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k + \epsilon, \tag{4.3}$$

wobei gilt, dass die Funktion f linear ist, sodass

$$E(y|x_1, ..., x_k) = f(x_1, ..., x_k) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \tag{4.4}$$

gilt. Wenn wir die Daten nach der Aspekt-Segmentierung einsetzen, dann erhalten wir die n Gleichungen

$$y_i = \beta_0 + \beta_1 x_{i1} + ...\beta_k x_{ik} + \epsilon_i \tag{4.5}$$

mit den zu schätzenden Parametern $\beta_0, ...\beta_k$. Man erzielt mit dieser Methode gute Ergebnisse, wenn die Zielvariable y stetig ist(Fahrmeir et al., 2018).
Das lineare Regressionsmodell lässt sich zusammenfassend auch in einer Matrixnotation schreiben,

$$y = X\beta + \epsilon \tag{4.6}$$

wobei y einen Vektor der Beobachtungen $1, ..., N$ beschreibt. Die Matrix X ist mit den Einträgen $x_{i,k}$ befüllt. Die Indizes stehen für die Beobachtungen i und den Regressor k. β definiert den Vektor von $k = 1, ..., K$ Einträgen, welche als zu schätzende Regressionskoeffizienten gelten. ϵ ist der Fehlervektor, ebenfalls mit der Länge N (Trevor et al., 2009). Es gibt eine Reihe von Methoden, die das lineare Modell lösen, aber die gängigste Methode ist die Schätzung der kleinsten Quadrate (Trevor et al., 2009):

$$\min_{\beta} \sum_{i=1}^{N} (y_i - f(x_i))^2 \tag{4.7}$$

4.2 Logistische Regression

Die logistische Regression ist trotz ihres Namens ein lineares Modell für Klassifizierungen und keine Regression. Sie wird in der Literatur auch Logit-Modell, Maximum-Entropie-Klassifikation (MaxEnt) oder loglinearer Klassifikator genannt. In diesem Modell werden die möglichen Klassen eines zu lösenden Problems mithilfe einer logistischen Funktion modelliert. Sie ist auch unter der Sigmoidfunktion bekannt (Yu et al., 2011). Die logistische Regression findet vor allem dann Verwendung, wenn die Zielvariable y kategorial ist und die Features bernoulliverteilt sind. Wie bei der linearen Regression ist das Ziel, die Zielvariable so zu modellieren, dass sie in Abhängigkeit zu den Features steht. Ein übliches logistisches Regressionsmodell

$$y_i = P(y_i = \pm 1) + \epsilon_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} + \epsilon_i \tag{4.8}$$

mit $\epsilon_i \sim N(0, \sigma^2)$ ist aber aus verschiedenen Gründen ungeeignet (Fahrmeir et al., 2018):

- Selbst wenn man auf die Normalverteilungsannahme für ϵ_i verzichtet, kann die Fehlervarianz $Var(\epsilon_i) = Var(y_i|x_i)$ nicht homoskedastisch, das heißt gleich σ^2, sein. Da die Zielvariablen y_i mit $\pi_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik}$ bernoulliverteilt sind, folgt, dass

$$Var(y_i) = \pi_i(1 - \pi_i) \tag{4.9}$$

  darüber hinaus von Features und den Parametern $\beta_0 + ...\beta_k$ abhängen und somit für alle i den gleichen Wert σ^2 besitzen kann.

- Des Weiteren lässt das lineare Modell für zum Beispiel $P(y_i = 1, 2, 3, 4, 5)$ (multinomiale logistische Regression) auch Werte $\pi_i < 1$ und $\pi_i < 5$ zu, welche nicht zulässig sind.

Der Einfachheit halber wird im Folgenden zunächst ein binäres Problem betrachtet. Eine genauere Erläuterung der multinomialen Regression erfolgt später.
Es wird das Modell

$$y_i = P(y_i = \pm 1) + \epsilon_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} + \epsilon_i \tag{4.10}$$

betrachtet, bei dem der Wertebereich der Funktion F im Intervall $[0, 1]$ liegt. Da es aus praxisbezogenen Gründen vorteilhaft ist, dass auch F streng monoton wächst, erhält man die logistische Verteilungsfunktion

$$F(\eta) = \frac{exp(\eta)}{1 + exp(exp(\eta)}, \tag{4.11}$$

mit der man das Logit-Modell

$$P(y_i = \pm 1) = \frac{exp(\eta)}{1 + exp(exp(\eta)} \tag{4.12}$$

mit dem *linearen Prädiktor*

$$\eta_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik}. \tag{4.13}$$

aufstellt.
Wie beim linearen Rezessionsmodell wird vorausgesetzt, dass die Zielvariablen y_i bei gegebenen Feature-Werten $x_i = (x_i, ..., x_{ik})$ bedingt unabhängig sind. Erhöht sich der Wert des Prädikators η_i um eine Einheit, so erhöht sich die Wahrscheinlichkeit für die Zielvariable y_i *nichtlinear* von $F(\eta)$ auf $F(\eta + 1)$. Aus diesem Grund können die Parameter $\beta_0, ..., \beta_k$ nicht mit der KQ-Methode geschätzt werden. Für die Lösung verwendet man die Maximum-Likelihood-Schätzung. Die multinomiale (mehrkategoriale) logistische Regression lässt sich wie folgt definieren (Fahrmeir et al., 2018):
Multinomiale Logistische Regression

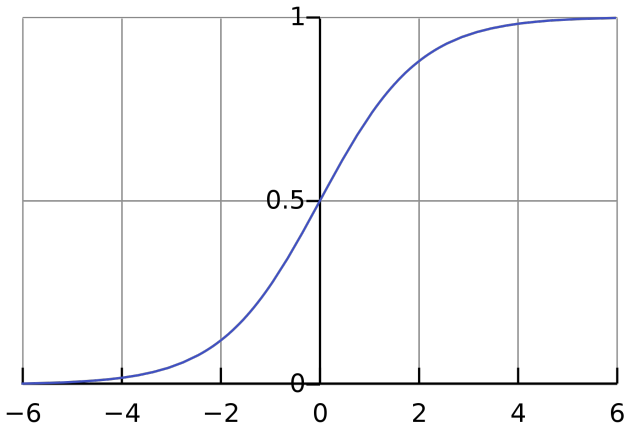
Daten:
Die Zielvariable $y_i \in \{1, ..., c\}$ ist kategorial und nominalskaliert. Zusätzlich sind Features x_i gegeben, die nicht von der Kategorie der Zielvariablen abhängen.
Modell:
Die Auftretenswahrscheinlichkeit für die Kategorie r ist bestimmt durch

$$P(y_i = r|x_i) = \pi_{ir} = \frac{\exp(\beta_o + \beta_1 x_{i1} + ... + \beta_k x_{ik})}{1 + \exp(\beta_o + \beta_1 x_{i1} + ... + \beta_k x_{ik})} = \frac{exp(x_i\beta)}{1 + \exp(x_i\beta)}, r = 1, ..., q. \tag{4.14}$$

Für die Referenzkategorie c gilt:

$$\pi_{ic} = 1 - \pi_{i1} - ... - \pi_{iq} = \frac{1}{1 + \sum_{s=1}^{q} exp(x_i\beta_s)} \tag{4.15}$$

Die Parameter $\beta_r = \beta_r 0, \beta_r 1, ..., \beta_r k)$ sind kategorienspezifisch, ebenso wie die linearen Prädiktoren $\eta_{ir} = x_i\beta_r = \beta_{r0} + x_{i1}\beta_{r1} + ... + x_{ik}\beta_{rk}, r = 1, ..., q$. An der Abbildung 4.1 erkennt man die Sigmoidfunktion. Bei dieser Funktion erkennt man, dass Wertebereich immer zwischen $[0, 1]$ liegt.



Quelle: `https://en.wikipedia.org/wiki/Sigmoid_function#/media/File:Logistic-curve.svg`

Abbildung 4.1: Sigmoidfunktion $\frac{1}{1+1+exp(-x)}$

4.3 Latent Rating Regression (LRR) Modell

Das *Latent Rating Regression (LRR) Modell* ist für speziell für das Schätzen von Aspekt-Ratings konstruiert worden und ist für diese Arbeit die Ausgangslage für die eigenen Modelle aus Kapitel 5. Als Input benötigt das LRR-Modell die Gesamtratings r_d, den vorverarbeiteten Rezensionstext als Worthäufigkeits-Matrix W_{dij} mit der Aspekt-Segmentierung für jede Rezension d und die Zuordnung für jeden Aspekt i. Eine allgemeine Annahme, die das LRR-Modell trifft, ist folgende: Der Autor entscheidet sich vorher, welche Aspekte er im

Bewertungstext kommentieren möchte. Für jeden Aspekt legt er mithilfe von Stimmungswörtern β fest, wie stark der Aspekt gewichtet beziehungsweise bewertet wird. Schließlich gibt der Autor am Ende ein Gesamtrating r_d ab, welches eine gewichtete Summe von allen Aspekten i, die in der Bewertung erwähnt werden, zusammenfasst. Die Aspekt-Ratings s_i können durch die Wörter, die jedem Aspekt i zugeordnet sind, geschätzt werden. Diese Annahme wird in Kapitel 5 später in den Algorithmen als global bezeichnet. Nicht jeder Autor eines Bewertungstextes wird alle Aspekte kommentieren und nicht für jeden Autor spielen die verschiedenen Aspekte eines Produktes eine gleich große Rolle. Bei der Domain Laptop werden für einen Gamer andere Aspekte wichtiger sein als zum Beispiel für einen Arbeitnehmer, der in der Verwaltung tätig ist. Das LRR-Modell berücksichtigt dies und erlernt deswegen nicht nur das Aspekt-Rating s_i, sondern auch eine Gewichtung α_i für jeden Aspekt i.

$$s_i = \sum_{j=1}^{n} \beta_{ij} W_{dij}. \tag{4.16}$$

Das Rating s_i errechnet sich aus der Linearkombination von W_{di} und β_{di}, wobei man über alle Stimmungswörter j eines Aspekts i aufsummiert. Dabei ist W_{dij} Worthäufigkeits-Matrix eines festen Wortcorpus n. n ist die Anzahl der vorkommenden Wörtern nach der Textvorverarbeitung. Das Gewicht β_i drückt die Stärke der Stimmung aus. Die allgemeine Annahme über die Gesamtrating r_d, die von dem Autor der Rezension d getroffen wird, lässt sich mit Normalverteilung wie folgt ausdrücken:

$$r_d \sim N(\sum_{i=1}^{k} \alpha_{di} \sum_{j=1}^{n} \beta_{ij} W_{dij}, \delta^2) \tag{4.17}$$

wobei k die Anzahl der Aspekten ist, die mit einer Eingabe von Startwörtern bestimmt wird. Die grundlegende Idee des LRR-Modells ist, die Lücke zwischen dem beobachteten Gesamtrating und dem Bewertungstext mithilfe der latenten Variablen α und β zu schließen. α repräsentiert die unterschiedlichen Präferenzen der einzelnen Aspekte eines Konsumenten und β die Gewichtung der Stimmungswörter. Um die Abhängigkeiten zwischen den Aspekten zu erfassen, wird außerdem eine mehrdimensionale Normalverteilung als Prior für α genutzt, das heißt

$$\alpha_d \sim N(\mu, \Sigma), \tag{4.18}$$

wobei μ der Vektor der Erwartungswerte der eindimensionalen Komponenten und Σ die Kovarianzmatrix ist. Aus der Kombination der Formeln 4.17 und 4.18 ergibt sich das bayessche Regressionsproblem:

$$P(r|d) = P(r_d|\mu, \Sigma, \delta^2, \beta, W_d) = \int p(\alpha_d|\mu, \Sigma) p(r_d| \sum_{i=1}^{k} \alpha_{di} \sum_{j=1}^{n} \beta_{dij} W_{dij}, \delta^2) d\alpha_d \tag{4.19}$$

wobei r_d und W_d die beobachteten Daten aus der Rezension d sind und $\Theta = (\mu, \Sigma, \delta^2, \beta)$ das Quadrupel aus dem Textcorpus der Modellparameter. Die graphische Darstellung der *Latent Rating Regression* ist in der Abbildung 4.2 als Plate-Notation beschrieben.

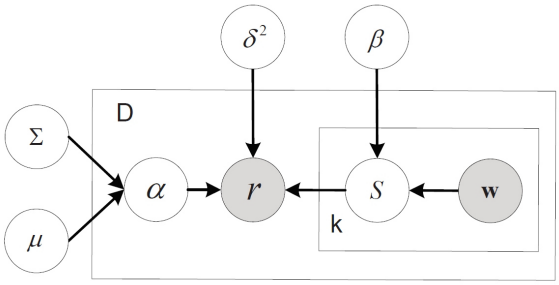


Abbildung 4.2: Plate-Notation Latent Rating Regression

Die äußere Box repräsentiert Rezensionen, während die innere Box die Zusammensetzung der latenten Aspektbewertungen und Wortbeschreibungen innerhalb einer Rezension darstellt. Die bayes'sche Regression wird mit einem EM-Algorithmus gelöst. Der E-Schritt berechnet, basierend auf dem aktuellen Modellparameter Θ, das Rating s_i und das Aspekt-Gewicht α_i. Der M-Schritt hingegen erlernt die Parameter des Modells, indem er über alle Bewertungen die Likelihood-Schätzung maximiert:

$$\Theta = \arg\max_{\Theta} \sum_{d \in D} \log p(r_d | \mu, \Sigma, \delta^2, \beta, W_d) \tag{4.20}$$

Die Schritte werden so lange wiederholt, bis die Likelihood am Ende gegen einen Grenzwert konvergiert (Wang et al., 2010).

Kapitel 5

Latente Aspekt-Rating Vorhersagemodelle

Dieses Kapitel führt einige Vorhersagemodelle ein, die für das Schätzen von Aspekt-Ratings verwendet werden. Zunächst wird der Ansatz der linearen Regression beschrieben, der auf allen Aspekt-Segmenten lernt. Das Gesamtrating r_d ist dabei die Zielvariable, weil es ein Ziel dieser Arbeit ist, auf Aspekt-Ratings zu verzichten. Die Worthäufigkeits-Matrix w_{dilj} beinhaltet die Features, wobei l die einzelnen Segmente oder auch Satzteile beschreibt. Das Lernen auf Satzteilebene wird im Folgenden als lokal bezeichnet. Auf dieser Ebene wird die Worthäufigkeits-Matrix direkt verwendet und nicht aufsummiert. Außerdem wird eine logistische Regression eingeführt, die auch auf allen Segmenten lernt. Diese Modelle werden im Folgenden als lokale lineare Regression und lokale logistische Regression bezeichnet. Bei den lokalen Modellen wird das fehlende Aspekt-Rating durch Mehrfachverwendung des Gesamtratings ausgeglichen. Dies wird auch als *weak Label* bezeichnet.
Im Gegensatz zu lokalen Modellen werden in diesem Kapitel auch globale Modelle konstruiert. Global heißt in diesem Kontext, dass eine Summe über die Aspekte i gebildet wird, um die fehlenden Aspekt-Ratings zu kompensieren. Die globalen Modelle lassen sich sowohl auf der Rezensionsebene als auch auf der Kategorieebene erstellen, da sich auf diesen Ebenen die Aspekte i aufsummieren lassen. Die Algorithmen werden anhand von formulierten Pseudocodes definiert, um eine genauere Vorstellung zu bekommen, wie welche Algorithmen später für die Evaluation verwendet werden.

5.1 Lokale lineare Regression

In der Forschung von *Rated Aspect Summarization of Short Comments* wird ein lokales Gesamtrating eingeführt (Lu et al., 2009). Die Annahme ist, dass der Autor konsistente Stimmungen in den Rezensionen bewertet. Das bedeutet, dass er entweder von dem Produkt vollständig überzeugt ist und nur Positives kommentiert oder, wenn er von dem

Produkt nicht überzeugt ist, dann schreibt der Autor nur Negatives in die Bewertung. In den Daten ist zu beobachten, dass es viele Kommentare gibt, die diese Behauptung stützen, aber auf keinen Fall eine allgemeine Gültigkeit haben. Eine große Anzahl der Kommentare ist sehr kurz geschrieben und die Autoren bewerten in diesen nur ein oder zwei Aspekte, in welchen dann auch eine konsistente Stimmung vorliegt. Diese Annahme wird in dieser Arbeit als *weak Label* bezeichnet, da die Gesamtratings mehrfach verwendet werden. Dieses *weak Label* wird zu allen Satzteilen l einer Rezension d zugeordnet und mit r_{dil} bezeichnet. Die Verwendung des *weak Labels* wird auch mit der linearen Regression in folgender Form

$$r_{dil} = \sum_{j=1}^{n} \beta_{ij} W_{dilj} + \epsilon \tag{5.1}$$

evaluiert. Da kaum Wortwiederholungen nach der Entfernung der Stoppwörter in einem Satzteil vorkommen, sind die Features nun annähernd bernoulliverteilt. Aufgrund dessen wird auch eine Implementierung mit der logistischen Regression durchgeführt, welche das nächste Kapitel 4.2 umfasst. Die logistische Regression ist für die Bernoulli-Verteilung entwickelt worden (Fahrmeir et al., 2018). Wang et al. (2010) und Lu et al. (2009) zeigen in ihren Evaluationen, dass lokale Vorhersagenmodelle zu besseren Ergebnissen führen, obwohl diese Art der Interpretation des *weak Labels* nachweisbare Fehler enthält. Es gibt Rezensionen, die sowohl negative Aspekte als auch positive Aspekte bewerten. Angesichts der guten Evaluationsergebnisse sind diese lokalen Algorithmen, wie zum Beispiel der Algorithmus 2, implementiert. Da die Matrix w_{dilj} auf Satzebene sehr viele Nulleinträge enthält,

Algorithm 2 Lokal Linear Regression (segment level)

Input: Corpus as collection of phrases W_{dilj} (BoW), label is the overall Rating r_{dil}

Output: aspect rating s_i,

1: Match all segments l with their overall Rating r_{dil}
2: Fit the Linear Regression Model with $r_{dil} \sim \sum_{j=1}^{n} W_{dilj}$
3: Estimate each aspect rating s_i with $\sum_{d=1}^{m} \sum_{j=1}^{n} \beta_{ij} \frac{W_{dij}}{n}$
4: Output: aspect rating s_i

genau genommen alle Wörter, die nicht in den Rezensionen vorkommen, aber dennoch im gesamten Corpus, wird das LibSVM-Format verwendet (Joachims, 1999). Dieses Format reduziert eine Matrixzeile, indem es die vorkommenden Nullen nicht explizit speichert. Gespeichert werden nur die Wörter, die auch in einer Rezension vorkommen. Dieses Format ist wichtig, da sich das gesamte Corpus bei der Berechnung der Features im Speicher befindet muss. Andernfalls kommt es schnell zu *Out-of-Memory*-Fehlern. In der ersten Spalte des LibSVM-Formats befindet sich das *weak Label*.

5.2 Lokale logistische Regression

Ein weiteres Verfahren, welches in dieser Arbeit nach der Aspekt-Segmentierung Verwendung findet und im Anschluss evaluiert wird, ist die logistische Regression. Wie in 4.2 beschrieben, ist die logische Regression ein Klassifikator. Bei Vorhersage der Aspekt-Ratings in dieser Arbeit liegen die Zielvariablen zwischen 1 und 5, deswegen ist es wünschenswert, dass bei der Lösung auch $s = \{x \in \mathbb{Z} | x \in [1, 5]\}$ gilt. Bei dem Evaluationsdatensatz liegen die Aspekt-Ratings genau in diesem Wertebereich, deswegen liegt es nahe, ein Verfahren zu wählen, welches diesen Wertebereich garantieren kann. Ein Vorteil der multinomial logistischen Regression gegenüber der linearen Regression ist, dass sich ein Modell konstruieren lässt, in dem die Zielvariablen im Wertebereich $[1, 5]$ liegen. Zusammengefasst lässt sich folgende These formulieren:

5.2.1 These. *Wenn man mittels einer logistischen Regression garantieren kann, dass die Zielvariablen in einem festen Wertebereich liegen, dann ist der Fehler der mittleren quadratischen Abweichung geringer als bei der linearen Regression.*

Diese These gilt es in der Evaluation in Kapitel 6 für dieses konkrete Problem dieser Arbeit zu bestätigen oder zu widerlegen. Eine Schlussfolgerung für eine Allgemeingültigkeit kann nicht untersucht werden und würde den Rahmen dieser Arbeit übersteigen. Dennoch gilt es, diesen vermeintlichen Vorteil der logistischen gegenüber der linearen Regression zu überprüfen.
Ein weiterer Vorteil, den die logistische Regression bietet, ist, dass sie für bernoulliverteilte Daten konzeptioniert ist. Bei der Aspekt-Segmentierung werden die Rezensionstexte in Satzteile aufgeteilt und den verschiedenen Aspekt-Clustern zugeordnet. Im späteren Verlauf des Algorithmus' werden die Texte in BoW-Format und auf Rezensionsebene zusammenaddiert. Wenn man in der Vorverarbeitung der Texte alle Stop-Words entfernt, dann ist zu beobachten, dass in den Satzteilen kaum Wiederholungen vorkommen. Es entsteht bei den Features eine Art Bernoulliverteilung zum Lernen. Wie in 4.2 beschrieben, ist die logistische Regression für binäre Features erforscht worden und wird in der Literatur als ein geeignetere Verfahren nahegelegt (Trevor et al., 2009). Der lineare Prädikator lässt sich in der Matrixschreibeweise folgendermaßen veranschaulichen:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_l \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 0 \\ 0 & \ddots & 1 & \vdots \\ \vdots & & 0 & \\ 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_j \end{pmatrix} + \epsilon$$

wobei y_l die Zielvariablen sind und der Index l für die Satzteile steht. Es gilt, dass die Anzahl von l größer gleich als die Anzahl der Rezensionen d ist. Die Matrix W_{dilj}, die

die beobachte Daten enthält, hat l Satzteile als Zeilen und j Wörter als Spalten. β definiert die zu schätzende Stimmung eines Wortes. Die Wörteranzahl des ganzen Corpus ist nicht unerheblich groß und lässt die Matrix wie auch schon bei der linearen Regression schnell wachsen. Damit man das ganze Corpus beim Lernen im Speicher behält, wird auch hier das LibSVM-Format von Joachims (1999) verwendet. Wenn es immer noch zu Speicherproblemen kommt, dann kann man dieses Format auch in einen Cache laden. Die Datenlage ist so, dass eine Rezension ein Gesamtrating zum Lernen zur Verfügung stellt. Damit man auf Satzteilebene lernen kann, werden die Gesamtratings auf alle Satzteile, die auch aspekt-spezifisch zugeordnet sind, mehrmals verwendet. Wenn man mithilfe von der mittleren quadratischen Abweichung das Ergebnis evaluiert, dann kommen sowohl Wang et al. (2010) als auch Lu et al. (2009) mit dem lokalen Label zu den besseren Ergebnissen. Der Algorithmus 3 verwendet als Input die zuvor erwähnten Satzteile:

Algorithm 3 Lokal Logistic Regression (segment level)

Input: Corpus as collection of phrases W_{dilj} (BoW), label is the overall Rating r_{dil}

Output: Aspect rating $s_i, s \in \{1, 2, 3, 4, 5\}$

1: Match all segments l with their overall rating r_{dil}
2: Fit the Logistic Regression Model with $r_{dil} \sim \sum_{j=1}^{n} W_{dilj}$
3: Estimate each aspect rating s_i with $\sum_{d=1}^{m} \sum_{j=1}^{n} \beta_{ij} W_{dij}$
4: Output: aspect rating s_i

5.3 Globale lineare Regression

Die globale lineare Regression erfolgt nach der in Kapitel 3 beschriebenen Aspekt-Segmentierung und berechnet schlussendlich ein Rating s für alle Aspekte i. Zuerst legt man einen Schwellenwert und eine maximale Iteration fest. Der Algorithmus 4 benötigt den Rezensionstext aus der Aspekt-Segmentierung als *Bag-of-Words* (BoW) w_{dij}. Diese Aspekt-Ratings können durch die Wörter, die jedem Aspekt zugeordnet sind, vorhergesagt werden. Da die Daten nur Gesamtratings über die Rezensionen d enthalten, werden diese für alle Aspekte i auf Rezensionsebene aufsummiert. Eine Annahme für eine globale Verwendung des Gesamtratings ist, dass sich Autoren beim Verfassen von Rezensionen am Ende Gedanken machen, welches Gesamturteil sie in einem Rating zusammenfassen wollen. Die genaue Annahme für die Verwendung des Gesamtratings ist in Kapitel 4.3 genauer erläutert. Sowohl der Bewertungstext als auch die Gesamtratings sind den Rezensionen d zugeordnet und werden auf dieser Ebene verwendet. Das Modell lässt sich wie folgt trainieren:

$$r_d = \sum_{i=1}^{k} \sum_{j=1}^{n} \beta_{ij} W_{dij} + \epsilon, \tag{5.2}$$

Algorithm 4 Global linear Regression (review level)

Input: Corpus as collection of reviews w_{dij} (BoW), label is the overall rating r_d

Output: Aspect Rating s_i,

1: Match the reviews with their overall rating r_d
2: Fit the Linear Regression Model with $r_d \sim \sum_{i=1}^{k} \sum_{j=1}^{n} W_{dij}$
3: Estimate each aspect rating s_i with $\sum_{d=1}^{m} \sum_{j=1}^{n} \beta_{ij} \frac{W_{dij}}{n}$
4: Output: aspect rating s_i

wobei der Index j für die einzelnen Wörter steht und β_{ij} die zu schätzenden Stimmungen für die einzelnen Wörter beschreibt. Der Autor gibt am Ende seiner Rezension ein Gesamtrating ab, welches eine gewichtete Summe von allen Aspekten, die in der Bewertung erwähnt werden, zusammenfasst.
Der Algorithmus 5 summiert für das Trainieren zusätzlich die Rezensionen d auf, was der Kategorieebene entspricht. Als Zielvariable verwendet das Modell das gemittelte Gesamtrating $\overline{r}$ über alle Rezensionen.
Abschließend lässt sich folgende These formulieren, die in der Evaluation überprüft wird:

Algorithm 5 Global linear Regression (category level)

Input: Corpus as collection of reviews w_{dij} (BoW), label is the overall rating r_d

Output: Aspect Rating s_i,

1: Match the reviews with their overall rating r_d
2: Fit the Linear Regression Model with $\overline{r} \sim \sum_{d=1}^{m} \sum_{i=1}^{k} \sum_{j=1}^{n} W_{dij}$
3: Estimate each aspect rating s_i with $\sum_{d=1}^{m} \sum_{j=1}^{n} \beta_{ij} W_{dij}$
4: Output: aspect rating s_i

5.3.1 These. *Wenn man das umfangreichere Corpus auf Satzeilebene l mit dem weak Label verwendet, dann performen die Vorhersagemodelle besser als die globalen Modelle, die die Aspekte i aufsummieren.*

Auch diese These wird in dem Experiment in Kapitel 6 untersucht.

5.4 Gewichtete globale lineare Regression

Eine weitere latente Variable α, die das Aspekt-Gewicht beschreibt, wird von Wang et al. (2010) aufgestellt. Dieses Aspekt-Gewicht, wie in dem theoretischen Kapitel 4.3 vorgestellt, soll die Lücke zwischen den beobachteten Gesamtratings und dem Bewertungstext mithilfe der latenten Variablen α und β schließen. Im Laufe der Arbeitsphase hat sich herausgestellt, dass der EM-Algorithmus des LRR-Modells (Kapitel 4.3) zu keinen konvergierenden Ergebnissen führt. In dem Kapitel 6 wird dies detaillierter beschrieben. Trotzdem findet

die Idee des Aspekt-Gewichts in dieser Arbeit weiter Verwendung, indem α_i nicht gelernt wird, sondern als Konstante in die Berechnung mit eingeht.

$$r_d = \sum_{i=1}^{k} \alpha_{di} \sum_{j=1}^{n} W_{dij}\beta_{ij} + \epsilon \tag{5.3}$$

Die Idee, die hinter der latenten Variable α steckt ist, dass die Autoren von Rezensionen unterschiedliche Präferenzen haben. Nicht jeder Autor wird alle Aspekte als gleich wichtig empfinden, da jeder Mensch andere Vorlieben hat. Die meisten Autoren schreiben in den Rezensionen darüber am ausführlichsten, was ihnen am wichtigsten ist. Dieses gilt sowohl für den positiven als auch für den negativen Fall. Dabei wird α_i durch

$$\alpha_{di} = \sum_{j=1}^{n} \frac{W_{dij}}{W_{dj}} \tag{5.4}$$

definiert. $\sum_{j=1}^{n} W_{dj}$ ist die Anzahl aller Wörter einer Rezension d und $\sum_{j=1}^{n} W_{dij}$ die Wortanzahl eines zugeordneten Aspekts in einer Rezension d. Eine Eigenschaft, die sich aus diesem Verfahren ergibt, ist $\sum_{i=1}^{k} \alpha_i = 1$. Das α auf Kategorieebene dagegen lässt sich ähnlich formulieren. Im Algorithmus 7 wird das α dagegen auf Kategorieebene verwendet und summiert zusätzlich alle Rezensionen auf. Eine These, die sich über diese Behauptung aufstellen lässt, ist folgende:

5.4.1 These. *Die Präferenzen von Autoren bezüglich verschiedener Aspekte in einer Rezension spiegeln sich in der zugeordneten Wortanzahl eines Aspektes wieder. Diese Gewichtung der Summe in den globalen Modellen ist wichtig, da es einen Einfluss der Aspekte darstellt.*

Ohne α geht bei den globalen Modellen nichts aspekt-spezifisches in die Schätzung mit ein. Die These 5.4.1 wird später in der Evaluation mit dem Algorithmen 6 und 7 überprüft.

Algorithm 6 Global Linear Regression with α (review level)

Input: Corpus in BoW as collection of reviews $\sum_{j=1}^{n} w_{dij}$, label is the overall rating r_d, together as *libsvm format*

Output: aspect rating s_i,

1: Match the reviews with their overall rating r_d
2: Calculate aspect weight $\alpha_d i = \sum_{j=1}^{n} \frac{w_{dij}}{w_{dj}}$
3: Fit the Linear Regression Model with $r_d \sim \sum_{i=1}^{k} \alpha_{di} \sum_{j=1}^{n} W_{dij}$
4: Estimate each aspect rating s_i with $\sum_{d=1}^{m} \sum_{j=1}^{n} \beta_{ij} \frac{W_{dij}}{n}$
5: Output: aspect rating s_i

Algorithm 7 Global Linear Regression with α (category level)

Input: Corpus in BoW as collection of reviews $\sum_{j=1}^{n} w_{dij}$, label is the overall rating r_d, together as *libsvm format*

Output: aspect rating s_i,

1: Match the reviews with their overall rating r_d
2: Calculate aspect weight $\alpha_i = \sum_{d=1}^{m} \sum_{j=1}^{n} \frac{W_{dij}}{W_{dj}}$
3: Fit the Linear Regression Model with $\overline{r} \sim \sum_{i=1}^{k} \alpha_i \sum_{d=1}^{m} \sum_{j=1}^{n} W_{dij}$
4: Estimate each aspect rating s_i with $\sum_{d=1}^{m} \sum_{j=1}^{n} \beta_{ij} W_{dij}$
5: Output: aspect rating s_i

Kapitel 6

Experiment

Das Experiment dieser Master-Thesis dient dazu, die Modelle, die mittels der Algorithmen aus Kapitel 5 erstellt wurden, auszuwerten. Dieses Kapitel führt eine mögliche Evaluierung mithilfe des Tripadivor-Datensatzes ein und stellt die Besonderheiten vor. Dabei werden die zuvor formulierten Thesen untersucht und ausgewertet. Des Weiteren präsentiert dieses Kapitel die verwendeten Wortlisten aus der Aspekt-Segmentierung.

6.1 Evaluationsmetrik

In dem Experiment wird die mittlere quadratische Abweichung verwendet, um bei den Modellen den Fehler zu bestimmen. Δ^2_{hotel} ist der Mean-Square-Error (MSE) für die Modelle, welcher die Qualität der Schätzung anhand des Abstandes misst (Sarkar et al., 2017). Formal gesehen nehmen wir an, dass g_{di} die *Ground-Truth-Ratings* für die Aspekte A_i definiert. Dieses Δ^2_{hotel} bildet die quadratische Differenz zwischen den geschätzten Aspekt-Ratings s_i und dem Durchschnitt der *Ground-Truth-Rating* $\overline{g_i}$. Man befindet sich bei beiden Ratings auf Hotelebene. Dies wird definiert als:

$$\Delta^2_{hotel} = \sum_{h=1}^{H} \sum_{i=1}^{k} s_{hi} - \overline{g}_{hi})^2/(k) * (H). \tag{6.1}$$

Vereinfacht ausgedrückt misst Δ^2_{hotel} den Fehler zwischen *Ground-Truth-Ratings* und geschätzten Ratings s_h für jeden Aspekt i auf Hotelebene H. Dabei ist h die das einzelne Hotel. Für die Evaluation der Wortlisten berechnet man für jeden Aspekt

$$\Delta^2_{aspect} = \sum_{h=1}^{H} (s_{hi} - \overline{g}_{hi})^2/(H), \tag{6.2}$$

um diese quantitativ zu evaluieren.
Eine Idee, um die Ergebnisse der Schätzungen einzuordnen, ist die Verwendung des *weak Labels*. Zunächst wird der Durchschnitt aller vorkommenden Gesamtratings

$$\overline{r}_i = \frac{\sum_{d=1}^{m} r_d}{m} \tag{6.3}$$

in einem Hotel berechnet, wobei m die Anzahl der vorkommenden Rezensionen ist. $\overline{r}_i$ wird auf Hotelebene gebildet und i-mal wiederholt, um die mittlere quadratische Abweichung

$$\Delta^2_{gesamtrating} = \sum_{h=1}^{H} \sum_{i=1}^{k} (\overline{r}_{ih} - \overline{g}_{ih})^2 / (k) * (H) \tag{6.4}$$

über alle Hotels zu berechnen.

6.2 Evaluation der Ergebnisse

Um die Ergebnisse vergleichbar zu machen, werden für alle Algorithmen die gleichen Datenvorbereitungsschritte angewandt. Mit den richtigen Vorbereitungen kann man in der Regel die Güte der Algorithmen deutlich verbessern. Diese stehen in diesem Experiment nicht im Fokus, da diese die Anzahl der Ergebnisse deutlich erhöhen. Das Kapitel 2.2 erläutert die Verfahren, die in diesem Experiment verwendet werden, um die Rezensionstexte für die Algorithmen aufzubereiten. Es sind folgende Vorbereitungsschritte zum Einsatz gekommen:

1. Schreibe alle Wörter in Kleinbuchstaben.
2. Entferne alle Stoppwörter[1].
3. Bilde den Wortstamm für jedes Wort.

6.2.1 Evaluationsdatensatz

Der Tripadvisor-Datensatz von Wang[2] bietet bis zu neun vordefinierte Aspekte als Tripadvisor-Datensatz. Bei dem Tripadvisor-Datensatz sind die Aspekte also durch die Aspekt-Ratings vorgegeben. Dennoch sind Autoren auf dieser Seite nicht dazu verpflichtet, alle neun Aspekt-Ratings abzugeben, da sie auf der Seite nur als optionale Felder aufgeführt sind. Zusätzlich kommt erschwerend hinzu, dass mittlerweile nicht mehr die ursprünglichen neun Aspekte bewertet werden können, da im Laufe der Zeit einige Aspekte von den Betreibern der Seite gestrichen worden sind, was dazu führt, dass der Datensatz nicht mehr konsistent bezüglich aller neun Aspekt-Ratings ist. Aktuell kann man beispielsweise nur noch die Aspekte *Sleep Quality*, *Service* und *Cleanliness* bewerten. Eine weitere Veränderung, die Schwierigkeiten mit sich bringt ist, dass die Aspekte *Check In/Front Desk* und *Business Service* zu einem *Service* Aspekt zusammengefasst worden sind und *Sleep Quality* hinzugefügt worden ist. Die alten Rezensionen sind auf der Seite immer noch verfügbar, aber stellen kein großes Corpus für die Evaluation bereit.
Die Rohdaten können also für die Evaluation nicht komplett benutzt werden, da die meisten Rezensionen nicht alle Aspekte bewertet haben. Das hat zur Folge, dass der Datensatz

[1] Onix text retrieval toolkit stopword list. http://www.lextek.com/manuals/onix/stopwords1.html.
[2] http://times.cs.uiuc.edu/ wang296/Data/

viele Hotels beinhaltet, die kein *Ground-Truth-Rating* besitzen. Um ein aussagekräftige Evaluation zu erhalten wird, zunächst die Anzahl der Aspekte reduziert. Ein Test, in dem sieben Aspekte definiert worden sind, führt nicht zum Erfolg, da die Aspekte *Check In-/Front Desk* und *Business Service* im Nachhinein von dem Betreiber der Website entfernt worden. Der Bootstrapping-Algorithmus funktioniert grundsätzlich mit sieben oder auch neun Aspekten, nur ist eine aussagekräftige Evaluation nicht gewährleistet, da zu wenige Aspekt-Ratings für *Check In/Front Desk* und *Business Service* vorliegen. Im Anhang kann man die verwendeten Startwörter für die sieben Aspekte nachschlagen.
Eine weitere Reduzierung auf fünf Aspekte, die die Teilmenge aus den alten und neuen Rezensionen ist, führt zu einem verwendbaren Datensatz für die Evaluation. In diesem Experiment sind folgende Startwörter für den Bootstrapping-Algorithmus verwendet worden:

Aspekt	**Startwörter**
Value	value, price, quality, worth
Room	room, suite, view, bed
Location	location, locate, traffic
Cleanliness	clean, dirty, maintain, smell
Service	service, manager, food, buffet, breakfast

Tabelle 6.1: Startwörter

6.2.2 Corpus

Der komplette Datensatz enthält 12.773 Hotels, wobei die meisten Hotels nur sehr wenig bis gar keine Rezensionen enthalten. Es ist als zweite Maßnahme für die Evaluation wichtig, die Daten zu sortieren und geeignete auszuwählen. Der bestmögliche Datensatz mit einer Zuordnung von allen fünf *Ground-Truth-Ratings* zur jeder Rezension ist nicht vorteilhaft, da dies die Daten auf eine zu kleine Textmenge für die Aspekt-Clusterungen reduziert. Diese erzielen nur mit einer großen Menge an Texten gute Ergebnisse.
Um den Datensatz dennoch zu benutzen, wird in dieser Arbeit die Lösung angeführt, dass die Evaluation auf Hotelebene stattfindet. Es ist dann für viele Hotels möglich einen Durchschnitt für die *Ground-Truth-Ratings* zu berechnen, sofern sie mindestens ein *Ground-Truth-Rating* für jeden Aspekt besitzen. Diese Ratings können benutzt werden, um die Metrik der mittleren quadratischen Abweichung zu ermitteln.
In diesem Kapitel werden zwei Versuche mit zwei verschiedenen Datengrößen durchgeführt. Das erste Corpus umfasst folgende Werte für das Experiment:

Anzahl der Hotels	1.753
Anzahl der Rezensionen	159.794
Satzteile	526.790
Wortanzahl	275.735

Tabelle 6.2: Corpus: Datensatz 1

Das Corpus besteht aus 1753 Hotels, zu denen wiederum 159794 Rezensionen enthalten sind. Ohne den Vorbereitungsschritt der Stammbildung erhöht sich die Wortanzahl auf 301837. Diese Auswahl entstand durch eine zufällige Stichprobe und enthält auch Hotels mit nicht nur einem *Ground-Truth-Rating*. Wo kein Durchschnittswert berechnet werden kann, dann wird stattdessen das Gesamtrating verwendet. Diese Vorgehensweise findet sich in den Versuchen von Wang et al. (2010) wieder, die auch diesen Datensatz zur Evaluierung verwenden.
In dem zweiten Versuch dieser Arbeit werden die Daten so reduziert, dass man garantieren kann, dass mindestens ein *Ground-Truth-Rating* pro Hotel zur Verfügung steht, um einen Durchschnitt zu berechnen. Das ist wichtig, damit die Evaluation zu fairen Ergebnissen kommen kann.

Anzahl der Hotels	223
Anzahl der Rezension	265.408
Satzteile	373.1741
Wortanzahl	241.739

Tabelle 6.3: Corpus: Datensatz 2

Dieses Corpus besteht aus 223 Hotels, wozu wiederum 265408 Rezensionen enthalten sind. Ohne den Vorbereitungsschritt der Stammbildung erhöht sich das Wortcorpus auf 301837.

6.3 Evaluation der Aspekt-Clusterung

In diesem Abschnitt werden die Aspekt-Cluster des Bootstrapping-Algorithmus' zunächst qualitativ evaluiert. Es werden die passenden einzelnen Wörter in der Tabelle fett markiert. Für diese Evaluation ist kein Stemming verwendet worden, damit man die Wörter besser erkennen kann. Später wird mit der Δ^2_{aspect}-Metrik (6.2) eine quantitativ Evaluation durchgeführt. Die Messung wird mit der *Globale lineare Regression mit* α auf Hotelebene gemessen. Tabelle 6.4 zeigt Cluster mit den Aspekten *Value, Room, Location, Cleanliness* und *Service*.

Value	Room	Location	Cleanliness	Service
parking	**bed**	**bus**	polite	**drinks**
continental	**small**	**hollywood**	nicely	**best**
cheaper	**shower**	**sights**	**carpets**	**shuttl**
deal	**tv**	**locate**	**rooms**	general
half	booked	perfect	friendly	**bar**
saved	told	**distance**	accommodating	turndown
save	**balcony**	**accommodating**	spacious	**waiters**
range	**space**	spacious	**pools**	**poolside**
recommend	arrived	**pools**	simple	**maid**
6	5	7	3	7
0,087	0,098	0,119	0,393	0,075

Tabelle 6.4: Wortlisten Bootstrapping-Algorithmus Datensatz 2

Man kann erkennen, dass die quantitative und qualitative Evaluation bei dem Aspekt *Cleanliness* übereinstimmen. Beide kommen zu dem Schluss, dass der Cluster über den Aspekt *Cleanliness* nicht so gut gebildet wird. Eine Verbesserung der Vorhersagemodelle, wäre es, bessere Startwörter für den Aspekt *Cleanliness* zu definieren. Dieser Cluster beinhaltet viele Adjektive, die jeder Kategorie zugeordnet werden können. Die anderen Aspekt-Cluster bilden dagegen ein vernünftiges Ergebnis. Vergleicht man aber das Ergebnis mit dem LDA-Modell in der Tabelle 6.5, dann erkennt man an diesen Wortlisten einen etwas klareren Cluster.

Value	Room	Location	Cleanliness	Service
recommend	**bed**	**minut**	**monorailroom**	**hotel**
price	**bathroom**	squar	**clean**	**monorailservice**
good	**room**	**subwai**	small	good
pay	**shower**	**shop**	nice	experience
star	**small**	perfect	comfortable	**business**
money	**large**	**cab**	**stale**	**suite**
worth	**comfortable**	**attract**	spacious	property
rate	**big**	**ride**	large	expect
high	**tv**	**monorail**	upgrade	part
9	9	7	3	4

Tabelle 6.5: Wortlisten LDA Datensatz 2

6.4 Evaluation der Vorhersagemodelle für Aspekt-Ratings

Es ist äußerst aufwändig und kostspielig, die Vorhersage der Aspekt-Ratings ohne maschinelle Hilfe auszuwerten, da ein Experte alle Kommentare lesen und annotieren müsste, um diese für eine aussagekräftige Evaluation nutzen zu können (Lu et al., 2009). Stattdessen wird der Datensatz von Tripadvisor mit den Aspekt-Ratings $\hat{s_i}$ verwendet. Im weiteren Verlauf werden diese als die *Ground-Truth-Ratings* bezeichnet. Diese Evaluation der Vorhersagemodelle erfolgt nach der Aspekt-Segmentierung, die mit dem Bootstrapping-Algorithmus aus Kapitel 3.2.1 erstellt worden sind. In diesem Kapitel geht es um die Berechnung von Aspekt-Ratings, die mithilfe von Vorhersagemodellen geschätzt werden. Es werden die Vorhersagemodelle (5) verwendet, die die Aspekt-Ratings mithilfe des Gesamtratings schätzen. Es liegt also für das Trainieren der Algorithmen kein vollständiger Datensatz für das überwachte Lernen vor, da der Input aus Rezensionstexten und Gesamtratings besteht.
In der folgenden Evaluation sind die Daten nicht klassisch in Trainingsdatensätze und Testdatensätze unterteilt. Die Unterteilung erfolgt stattdessen auf den verschiedenen Ebenen. Die Schätzung der Aspekt-Ratings s_i erfolgt auf Hotelebene. Auf dieser Ebene sind alle Rezensionen d in W_{dij} als BoW-Modell zusammengefasst und die Wörter j werden den verschiedenen Aspekten i zugeordnet. Das Trainieren erfolgt auf der einen Seite lokal (Satzteilebene) und auf der anderen Seite global (Rezensionsebene oder Hotelebene). Die globalen Verfahren schließen die Lücke, indem die Summe über die Aspekte i gebildet wird. Im ersten Teil dieser Evaluation der Aspekt-Ratings werden die Algorithmen von Wang et al. (2010) auf den ersten Datensatz von 1753 Hotels angewandt und diskutiert. In diesem Experiment wird zwischen dem vorhergesagten Aspekt-Rating und dem *Ground-Truth-Rating* mit Δ^2_{hotel} (6.1) evaluiert.

Methoden	Δ^2_{hotel}
Latent Rating Regression (LRR) Modell mit gelerntem α und mit L_{aux}	1,120
Latent Rating Regression (LRR) Modell mit gelerntem α (original)	5,275

Tabelle 6.6: Ergebnisse der Vorhersagemodelle

Das LRR-Modell konvergiert weder mit dem originalen Algorithmus (4.3) noch mit dem in der Implementierung verwendeten Hilfsterm. Dieser lässt sich wie folgt formulieren:

$$L_{aux}(r_d, \alpha_d, s_d) = \pi(s_{di} - r_d)^2, \tag{6.5}$$

wobei π ein vordefinierter Hyperparamter ist, der verwendet wird, um den Einfluss zu steuern. r_d definiert das Gesamtrating und s_{di} ist das geschätzte Aspekt-Rating. In dieser Arbeit sind eine Reihe von Hyperparametern getestet worden, die alle nicht dazu geführt haben, dass der EM-Algorithmus konvergiert. Die Feinabstimmungen der Hyperparameter

sind hilfreich, um die Initialisierung zu verbessern. Eine Optimierung des MSE konnte weder mit dem Java Optimierungsframework *lbfgs*[3] noch mit einer Implementierung in Python erreicht werden. In Python ist der Optimierer vom Scipy[4] Framework verwendet worden. Während der Laufzeit ist zu beobachten, dass sich beide Modelle von Wang et al. (2010) der mittleren quadratische Abweichung (MSE) des Gesamtratings annähern. Diese Modelle konvergieren nicht gegen den MSE, der an den vorhergesagten Aspekt-Ratings und *Ground-Truth-Ratings* gemessen wird. Daraus lässt sich schließen, dass man den MSE, der an den vorhergesagten Aspekt-Ratings gemessen wird, nicht direkt mit dem LRR-Modell optimieren kann.
Des Weiteren wird in dieser Evaluation mit der Metrik Δ^2_{hotel} (6.1) gemessen und gegenüber der formulierten Einordnungsmetrik Δ^2_{gr} (6.4) durchgeführt. Δ^2_{gr} ist der einfachste Fall, da man nur die Gesamtratings mit denen der Aspekt-Ratings gleichsetzt und diese dann gegen die *Ground-Truth-Ratings* misst. Die erste Versuchsreihe über den Datensatz von 1753 Hotels ist in der folgenden Tabelle zusammengefasst:

Vorhersagemodelle	Δ^2_{hotel}
Global Latent Rating Regression (LRR) Modell mit konstantem α	0,468
Lokale logistische Regression	0,548
Lokale lineare Regression	0,406
Globale lineare Regression ohne α (review level)	0,409
Globale lineare Regression mit α (review level)	0,395
Globale lineare Regression ohne α (hotel level)	0,305
Globale lineare Regression mit konstantem α (hotel level)	0,209
$\Delta^2_{gesamtrating}$	**0,161**

Tabelle 6.7: Ergebnisse der Vorhersagemodelle Datensatz 1

Die genaueren Erläuterungen zu den Lernmethoden sind in Kapitel 5 weiter ausgearbeitet. Die lokalen Modelle verwenden das Gesamtrating auf Satzteilebene und die globalen Modelle lassen sich als die Summe der einzelnen Aspekt-Ratings für jedes Hotel oder jede Rezension formulieren. Diese Summen werden mit dem konstanten α (5.4) gewichtet, wobei auf Hotelebene alle Rezensionen aufsummiert werden.
Das *global Latent Rating Regression (LRR) Modell mit konstantem* α konvergiert auch nicht. Der Wert 0,468 wird mit Hilfe des Hilfsterms L_{aux} erreicht, mit dem man eine gute Initialisierung erreichen kann, aber während der Laufzeit werden die Werte der Metrik Δ^2_{hotel} schlechter.
Die lokale lineare Regression und die lokale logistische Regression lassen sich nicht mit α

[3] `http://www.netlib.org/opt/lbfgs_um.shar`
[4] `https://www.scipy.org/`

gewichten, da die Features aspektspezifisch zum Lernen vorliegen. Da die Satzteile kaum wiederholte Wörter beinhalten, liegen die Features annähernd binär vor. Die logistische Regression wird in der Literatur für binäre Features als eine geeignete Wahl erläutert (Fahrmeir et al., 2018). Der errechnete Fehler kann bei der *lokalen logistischen Regression* mit 0,548 beziffert werden. Hierzu ist zu sagen, dass die Aspekt-Ratings ganzzahlig in dem Wertebereich von 1 bis 5 liegen und die *Ground-Truth-Ratings* nach der Durchschnittsberechnung auf drei Stellen nach dem Komma gerundet sind. Der Vergleich zwischen einer Regression und einem Klassifikator mittels der Δ^2_{hotel} ist nicht fair, da der Klassifikator, wie er hier verwendet wird, nur ganze Zahlen erlernen kann. Die lokale lineare Regression misst dagegen einen Fehler von 0,406 und ist in diesem Vergleich besser. Beim Datensatz 6.2 ist der Vergleich mit dem durch Δ^2_{gr} gemessenen Ergebnis nicht ganz fair, da es einige Hotels beinhaltet, die kein *Ground-Truth-Rating* enthalten und stattdessen das gemittelte Gesamtrating verwendet wird. In diesem Fall misst der Fehler mit Δ^2_{gr} null und bevorteilt dieses Ergebnis. Um zu aussagekräftigen Messungen durch Δ^2_{gr} zu kommen, wird ein zweiter Datensatz mit 223 Hotels und 265.408 Rezensionen verwendet. Dieser beinhaltet keine gemittelte Gesamtratings auf Hotelebene und es lassen sich auch aussagekräftigere gemittelte *groud-truth Ratings* berechnen. Außerdem hat man die Möglichkeit den Unterschied zu untersuchen, wenn die Datenlage deutlich besser wird.

Vorhersagemodelle	Δ^2_{hotel}
Global Latent Rating Regression (LRR) Modell mit α	0,461
Lokale logistische Regression	0,436
Lokale lineare Regression	0,301
Globale lineare Regression ohne α (review level)	0,320
Globale lineare Regression mit α (review level)	0,324
Globale lineare Regression ohne α (hotel level)	0.234
Globale lineare Regression mit α (hotel level)	**0.154**
$\Delta^2_{gesamtrating}$	0,165

Tabelle 6.8: Ergebnisse der Vorhersagemodelle Datensatz 2

Es ist zu erkennen, dass - bis auf das LRR-Modell - alle Modelle bessere Ergebnisse erzielen. Anhand der beiden Ergebnistabellen 6.7 und 6.8 werden die Thesen aus den vorigen Kapiteln diskutiert.

These 5.2.1. *Wenn man mittels einer logistischen Regression garantieren kann, dass die Zielvariablen in einem festen Wertebereich liegen, dann ist der Fehler der mittleren quadratischen Abweichung geringer als bei der linearen Regression.*

Die These lässt sich widerlegen, da die logistische Regression in beiden Versuchen nach den LRR-Modellen zu den schlechtesten Ergebnissen führt. Dies lässt sich vor allem durch die fehlende Fairness zu erklären.

These 5.3.1 *Wenn man das umfangreicheres Corpus auf Satzeilebene l mit dem weak Label verwendet, dann performen die Vorhersagemodelle besser als die globalen Modelle, die die Aspekte i aufsummieren*

Auch diese These lässt sich nicht bestätigen, da insbesondere die globalen Modelle auf Hotelebene am besten performen. Wenn man nur die lokale lineare Regression im Vergleich zu den globalen Modellen auf Rezensionsebene untersucht, kommt man zu dem Schluss, dass die lokale Variante mit den *weak labels* zu minimal besseren Ergebnisse führt. Die in dieser Arbeit formulierte Behauptung, lässt sich durch die folgende These überprüfen:

These 5.4.1 *Die Präferenzen von Autoren bezüglich verschiedener Aspekte in einer Rezension spiegeln sich in der zugeordneten Wortanzahl eines Aspektes wieder. Diese Gewichtung der Summe in den globalen Modellen ist wichtig, da es einen Einfluss der Aspekte darstellt.*

Diese These trifft auf Hotelebene zu, da die Modelle mit einer Gewichtung deutlich bessere Ergebnisse erlernen als ohne. Im zweiten Datensatz macht die Gewichtung den entscheidenden Unterschied und führt zu einem deutlich besseren Ergebnis von 0,154. Es schlägt den einfachen Ansatz der Δ^2_{gr}-Messung. Es schlägt die Ausgangslage des LRR-Modells von Wang et al. (2010) deutlich.
Auf Rezensionsebene erzielen die globalen Modelle bezüglich der Gewichtung α widersprüchliche Ergebnisse. Dies lässt sich eventuell dadurch erklären, dass es viele sehr kurze Rezensionen gibt, die nicht alle Aspekte bewertet haben. Diese Aspekte der kurzen Rezensionen werden dann deutlich stärker gewichtet als die längeren Rezensionen, da diese alle Aspekte bewerten.

Kapitel 7

Fazit und Ausblick

In diesem Kapitel wird ein Fazit der Erbegnisse dieser Master-Thesis gezogen. Es wird die ausgehende Forschungsfrage aus Kapitel 1.2 beantwortet. Anschließend wir eine kurze Diskussion über die anfangs erstellte Pipeline geführt und wie diese in den Kontext des überwachten und unüberwachten Lernens einzuordnen ist. Abschließend wird es einen Ausblick geben, der über mögliche weitere Arbeiten informiert. Es werden weitere Modelle genannt, die sich als weiterführende Modelle erweisen können.

7.1 Fazit

Diese Master-Thesis stellte am Anfang die Frage: »*Ist es möglich, aspekt-basierte Ratings anhand von automatisierten Vorhersagemodelle zu erlernen, welche Rezensionstexte und Gesamtratings verwenden, um Produkte mittels aspekt-spezifischer Ratings detaillierter vergleichen zu können?*«
Um diese Frage zu beantworten, ist zunächst eine Pipeline entwickelt worden, um die Komplexität dieser Frage in folgende Schritte zu unterteilen:

1. Crawlen der Produktbewertungen,
2. Datenvorbereitung,
3. Aspekt-Clusterung,
4. Aspekt-Segmentierung,
5. Vorhersagemodelle,
6. Evaluation der Ergebnisse.

Dabei wurde jeder einzelne Schritt unter Berücksichtigung der Datenlage, die aus Gesamtratings und Rezensionstexten besteht, implementiert. Auf Grund der fehlenden Labels für aspekt-basiertes überwachtes Lernen, wurde der Fokus auf einen Bootstrapping-Algorithmus gelegt, der auf der einen Seite die fehlenden Labels kompensiert, indem er

unüberwacht ist, und auf der anderen Seite eine Kontrolle mithilfe von Startwörtern über die Cluster bietet. Für die Beantwortung der Forschungsfrage sind einige einfache Vorhersagemodelle aufgestellt worden, um die fehlenden Aspekt-Ratings von Produkten zu schätzen. Zuvor hat sich herausgestellt, dass das LRR-Modell von Wang et al. (2010) und LARA-Modell (Wang et al., 2011) keine konvergierenden Ergebnisse liefert. In diesem Zuge wurden einfache Vorhersagemodelle erstellt und diese mit dem LRR-Modell verglichen.
Bei der Erstellung der Vorhersagemodelle hat sich eine aspekt-spezifische Gewichtung α als nützlich herausgestellt, da es den entscheidenden Vorteil hat, eine aspekt-spezifische Gewichtung zu beinhalten, welche beim Aufsummieren der aspekt-spezifischen Wörter verloren geht. Zudem sind die lokalen und globalen Annahmen der Zielvariable durch eine Evaluation verglichen worden. Auf der einen Seite verwenden die lokalen Modelle das Gesamtrating für alle fehlenden Zielvariablen mehrfach und auf der anderen Seite bilden die globalen Modelle die Summe über alle Aspekte. Mit α lässt sich die Summe über alle Aspekte gewichten. Dies erzielt die besten Ergebnissen in dieser Master-Thesis.
Wie in Kapitel 6.2.1 erklärt, ist der Tripadvisor-Datensatz für die Evaluation unvollständig, denn man hat nicht für jede Rezension alle *Ground-Truth-Ratings* zur Verfügung. Dies ist der Grund, warum man auf Hotelebene mit Durchschnittswerten evaluieren muss. Die Evaluation auf Rezensionsebene ist nicht ohne weiteres möglich. Diese Vorgehensweise entspricht nicht ganz dem Standard. Dies ist aber der fehlenden Konsistenz der Daten geschuldet. Eine andere Möglichkeit der Evaluation ist, dass man die fehlenden *Ground-Truth-Ratings* mit dem passenden Gesamtrating ausfüllt, um dann auf Rezensionsebene zu evaluieren. Dies hat aber den Nachteil, dass die verwendete Einordnungsmetrik Δ^2_{gr} bevorteilt wird, da sich dann beide Werte entsprechen und die Differenz gleich null ist.
Abschließend kann man die Forschungsfrage wie folgt beantworten: Bei guter Datenlage ist es möglich, Aspekt-Ratings anhand von Rezensionstexten und Gesamtratings zu schätzen, welche dann verwendet werden können, um Produkte aspekt-spezifisch zu vergleichen.

7.2 Diskussion

Betrachtet man die Pipeline im Ganzen, dann ist sie weder rein überwacht noch unüberwacht, sondern hat interessante Verbindungen zu beiden Seiten. Einerseits sind die Zielfunktionen der Vorhersagemodelle eines überwachten Modells sehr ähnlich, da die beobachteten Gesamtratings den Labels entsprechen. Im Gegensatz zu einem regulären überwachten Modell sollen die Vorhersagemodelle kein Modell für vorhergesagte Aspekt-Ratings sein; stattdessen sind die Modelle viel eher daran interessiert, die versteckten Aspekt-Ratings zu schätzen, die sich aus den beobachteten Gesamtratings und Rezensiontexten ergeben. Die globalen Modelle können aber auch für die Vorhersage von Gesamtratings verwendet werden. Aspekt-Ratings werden mit den globalen Vorhersagemodellen nicht direkt vorhergesagt. Mit den geschätzten Aspekt-Ratings kann man aber recht einfach traditionelle

überwachte Vorhersagemodelle erstellen, die die Aspekt-Ratings vorhersagen. Dies ist der wesentliche Unterschied zwischen gewöhnlichen Vorhersagemodellen und den in dieser Arbeit verwendeten Vorhersagemodellen (Wang et al., 2010). Andererseits verhält sich der verwendete Bootstrapping-Algorithmus für die Aspekt-Clusterung wie ein unüberwachtes Modell, da er keine aspekt-spezifischen Labels benötigt und dennoch am Ende der Pipeline latente Aspekt-Ratings entdeckt. Ein großer Nachteil dieser Pipeline ist, dass die Modelle neu gelernt werden müssen, wenn neue Daten hinzukommen und diese dementsprechend aktualisiert werden sollen.

7.3 Ausblick

Die implementierte Pipeline bietet einige Möglichkeiten zur Optimierung und Erweiterung. Zum einem kann man in dem Schritt der Aspekt-Clusterung unüberwachte Methoden implementieren, die vielversprechende Modelle sein können und ohne Startwörter auskommen (Titov und McDonald, 2008; Jo und Oh, 2011; Zhao et al., 2010). Hier wäre es ein erster Schritt, mit dem hier verwendeten LDA-Modell eine Schnittstelle zu implementieren, welche die Topics mit den Aspekten zusammenbringt (engl. Mapping). Zum anderen könnte man weitere Bootsrapping-Algorithmen mit der hier formulieren Δ^2_{aspect}-Metrik evaluieren (Andrzejewski und Zhu, 2009; Chen et al., 2013; Shu et al., 2017). Diese Evaluation lässt sich sowohl für unüberwachte als auch halb-überwachte Modelle durchführen. Des Weiteren können die verwendeten Startwörter des Bootstrapping-Algorithmus' genauer dahingehend überprüft werden, ob sie sich auf die Δ^2_{aspect}-Metrik auswirken und sich so bessere Cluster bilden lassen. Dies könnte hilfreich sein, wenn kein Domain-spezifisches Vorwissen vorhanden ist. Außerdem ist die Verwendung der logistischen Regression als Klassifikator gegenüber den Regressionen nicht fair und müsste zum Beispiel nach dem Schätzen fairer gestaltet werden. Ein Ansatz wäre beispielsweise, dass man nach dem Schätzen eine Funktion mit den Aspekt-Ratings als Input aufruft, welche die Ergebnisse der logistischen Regression in einem reellen Zahlenbereich abbildet.
Des Weiteren wäre es interessant zu wissen, wie die vorgestellten Modelle Aspekt-Ratings vorhersagen. Dies könnte von großem Vorteil sein, um eventuell hinzukommende Daten zu aktualisieren und zu nutzen. Es gibt noch viele Herausforderungen, wie beispielsweise den Umgang mit Synonymen und Sarkasmus, die in dieser Arbeit gar nicht beleuchtet wurden und Potenzial für weitere Forschung bieten. Da diese Arbeit gezeigt hat, dass bei guter Datenlage aspekt-basierte Auswertung funktionieren kann, wäre es eine Möglichkeit, eine Website zu erstellen, auf welcher alle Produkte, die im Internet veräußert werden, anhand der geschätzten Ratings verglichen werden können.

Anhang A

Weitere Informationen

A.1 CD Inhalt

Die dieser Master-Thesis beiliegende CD enthält folgende Inhalte:

1.) Diese Master-Thesis im PDF-Format.

2.) Ergebnissdatein mit allen Aspekt-Ratings

3.) Original LARA-Framework Wang et al. (2010)

4.) Modifiziertes LARA-Framework Wang et al. (2010)

5.) LARA-Framework in Python

6.) LDA und LSI (Gensim Framework)

A.2 Startwörter und Corpus

Aspekt	**Startwörter**
Value	value, price, quality, worth
Room	room, suite, view, bed
Location	location, traffic, minute, restaurant
Cleanliness	clean, dirty, maintain, smell
Service	service, food, breakfast, buffe
Check In/Front Desk	stuff, check, help, reservation
Business service	business, center, computer, internet

Tabelle A.1: Startwörter für 7 Aspekte

Anzahl der Hotels	881
Anzahl der Rezension	92006
Satzteile	295635
Wortkorpus	191343

Tabelle A.2: Korpus mit 7 Aspekten

Abbildungsverzeichnis

Literaturverzeichnis

Andrzejewski, D. und Zhu, X. (2009). Latent dirichlet allocation with topic-in-set knowledge, *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*.

Blei, D. M., Ng, A. Y. und Jordan, M. I. (2003). Latent dirichlet allocation, *Journal of machine Learning research* .

Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M. und Ghosh, R. (2013). Exploiting domain knowledge in aspect extraction, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Fahrmeir, L., Kneib, T. und Lang, S. (2018). Regressionsmodelle, *Regression: Modelle, Methoden und Anwendungen* S. 19–58.

Hu, M. und Liu, B. (2004). Mining and summarizing customer reviews, *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Jo, Y. und Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis, *Proceedings of the fourth ACM international conference on Web search and data mining*.

Joachims, T. (1999). Svmlight: Support vector machine, *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund* **19**(4).

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, *Ijcai*, Vol. 14, Montreal, Canada, S. 1137–1145.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*, Cambridge University Press.

Liu, B., Hu, M. und Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web, *Proceedings of the 14th international conference on World Wide Web*, ACM, S. 342–351.

Liu, Q., Liu, B., Zhang, Y., Kim, D. S. und Gao, Z. (2016). Improving opinion aspect extraction using semantic similarity and aspect associations., *AAAI*, S. 2986–2992.

Lu, Y., Zhai, C. und Sundaresan, N. (2009). Rated aspect summarization of short comments, *Proceedings of the 18th international conference on World wide web*, ACM, S. 131–140.

Sarkar, D., Bali, R. und Sharma, T. (2017). Practical machine learning with python: A problem-solver's guide to building real-world intelligent systems.

Shu, L., Xu, H. und Liu, B. (2017). Lifelong learning crf for supervised aspect extraction, *arXiv preprint arXiv:1705.00251* .

Titov, I. und McDonald, R. (2008). Modeling online reviews with multi-grain topic models, *Proceedings of the 17th international conference on World Wide Web*.

Trevor, H., Robert, T. und JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction.

Vijayarani, S., Ilamathi, M. J. und Nithya, M. (2015). Preprocessing techniques for text mining-an overview, *International Journal of Computer Science & Communication Networks* **5**(1): 7–16.

Wang, H., Lu, Y. und Zhai, C. (2010). Latent aspect rating analysis on review text data: a rating regression approach, *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Wang, H., Lu, Y. und Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision, *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Yang, Y. und Pedersen, J. O. (1997). A comparative study on feature selection in text categorization, *Icml*, Vol. 97, S. 412–420.

Yu, H.-F., Huang, F.-L. und Lin, C.-J. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models, *Machine Learning* **85**(1-2): 41–75.

Zhao, W. X., Jiang, J., Yan, H. und Li, X. (2010). Jointly modeling aspects and opinions with a maxent-lda hybrid, *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 21. Februar 2019

Kristof Wilke