Universität Zürich[UZH]

Institut für Computerlinguistik

Aspektsentimentklassifikation in Rezensionen

Bachelorarbeit der Philosophischen Fakultät
der Universität Zürich

Referent: Prof. Dr. Martin Volk
Betreuer: Dr. Manfred Klenner

Verfasser:
Thomas David Murphy
Matrikelnummer 11-708-450
Technikumstrasse 37
3400 Burgdorf

1. Dezember 2016

Zusammenfassung

Bei der aspektbasierten Sentimentanalyse stehen zwei Aufgaben im Vordergrund: Die Aspektextraktion und die Aspektsentimentklassifikation. Mit der letzteren habe ich mich in der vorliegenden Bachelorarbeit befasst. Basierend auf Trainingsdaten der Domäne „Restaurants“ und ferner „Laptops“ des SemEval-Workshops 2016 habe ich Features extrahiert, um einen Klassifikator aus dem WEKA-Toolkit [Hall et al., 2009] zu trainieren, der Aspekte als positiv, neutral oder negativ klassifiziert. Bei der zehnfachen Kreuzvalidierung auf den Trainingsdaten erreicht mein System eine Genauigkeit von 76.5058%, was über einer Mehrheitsklassen-Baseline liegt. Auf den Testdaten erreicht mein System eine Genauigkeit von 71.2456%. Des Weiteren zeige ich in einem Vergleich auf, dass mein System domänenunabhängig funktioniert.

Danksagung

Ich bedanke mich ganz herzlich bei allen, die mich während der Erstellung meiner Bachelorarbeit unterstützt haben. Besonderer Dank gilt Herrn Dr. Manfred Klenner, welcher sich bereit erklärt hatte, meine Arbeit zu betreuen. Ich konnte mich jederzeit an ihn wenden und war für seine konstruktiven Ratschläge stets dankbar. Ebenfalls möchte ich mich bei Herrn Prof. Dr. Martin Volk bedanken, welcher sich als Referent zur Verfügung gestellt hat. Des Weiteren möchte ich dem ganzen Institut für Computerlinguistik der Universität Zürich meinen Dank aussprechen für die Unterstützung während des Studiums, sei dies als DozentInnen, TutorInnen, BetreuerInnen oder auch einfach zwischendurch bei offenen Fragen. Zuletzt möchte ich den Organisatoren des SemEval-Workshops 2016 für die Zurverfügungstellung der Workshopdaten danken.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsliste

ABSA	Aspektbasierte Sentimentanalyse
ACL	Association for Computational Linguistics
AKTSKI	Kürzel für das Forschungsteam von Pateria and Choubey [2016]
ARFF	Attribute-Relation File Format (verwendet in WEKA-Toolkit)
BOW	Bag-Of-Words
C4.5	Algorithmus um aus Trainingsdaten Entscheidungsbäume zu generieren
CoNLL	Conference on Natural Language Learning
CRF	Conditional Random Field
HMM	Hidden Markov Model
IBk	Implementation eines k-nearest-Neighbours-Klassifikators (in WEKA-Toolkit)
J48	Implementation von C4.5-Entscheidungsbäumen (in WEKA-Toolkit)
JJ	Adjective (Penn Treebank POS-Tag)
knn	k-nearest neighbours (Klassifikator)
LDA	Latent Dirichlet Allocation
MALLET	MAchine Learning for LanguagE Toolkit
ML	Machine Learning (maschinelles Lernen)
MPQA	Multi-Perspective Question Answering
NLP	Natural Language Processing
NLTK	Natural Language ToolKit
NP	Nominalphrase
OTE	Opinion Target Expression oder Opinion Target Extraction
pLSA	probabilistic Latent Semantic Analysis
POS	Part-Of-Speech
SA	Sentimentanalyse
SemEval	International Workshop for Semantic Evaluation
SVM	Support Vector Machine

VBD	Verb, past tense (Penn Treebank POS-Tag)
VBZ	Verb, 3rd person singular present (Penn Treebank POS-Tag)
WEKA	Waikato Environment for Knowledge Analysis (machine learning toolkit)
XML	eXtensible Markup Language

1. Einleitung

Sentimentanalyse (auch „Opinion Mining“) beschäftigt sich mit der Analyse der von Menschen geäusserten Meinungen über Entitäten. Unter Entitäten werden Produkte, Personen, Ereignisse, Themen, usw. sowie deren Attribute verstanden. In Zeiten der sozialen Medien, Online-Shopping und dergleichen ist eine ganze Fülle von subjektiven Meinungsäusserungen verfügbar. Potentielle Käufer eines Produkts sind daran interessiert, wie andere Käufer diese bewerten und neue Käufer bewerten bereitwillig ihre neueste Erwerbung. Firmen, Organisationen und Personen haben entsprechend ihren Zielen Interesse an diesen öffentlichen Meinungen. Die Sentimentanalyse versucht, diese Meinungen zu erkennen und weiterverwendbar zu machen.
Sentimentanalyse kann auf Dokumentebene durchgeführt werden, wobei ein Dokument die kleinste zu betrachtende Einheit darstellt. Aufgrund des Dokuments wird die Meinung über eine Entität entsprechend gewertet. Ebenfalls möglich ist die Analyse auf Satzebene, um detaillierte Ergebnisse zu erhalten. Doch oftmals ist auch diese Analyse von Meinungen für bestimmte Anwendungen zu ungenau. So ist bei einer Rezension eines Laptops nicht nur die Gesamtbewertung des Geräts relevant, sondern auch, was über einzelne Komponenten und Eigenschaften ausgesagt wird, wie im folgenden Beispiel ersichtlich:

> “The display on this computer is the best I’ve seen in a very long time, the battery life is very long and very convienent.”[1]

In diesem Beispiel sind zwei Eigenschaften des Laptopmodells ersichtlich: Einerseits besitzt es einen Bildschirm, welcher positiv bewertet wird und andererseits einen Akku, dessen Leistung hier ebenfalls als positiv wahrgenommen wird. Die aspektbasierte Sentimentanalyse (ABSA) beschäftigt sich mit ebensolchen Eigenschaften, genannt Aspekte, von Entitäten. Das Vorgehen in der aspektbasierten Sentimentanalyse kann grob in zwei Schritte eingeteilt werden: Aspektextraktion und Aspektsentimentklassifikation. Auch am „International Workshop for Semantic Evaluation“ (SemEval) wird seit 2014 eine Aufgabe zur aspektbasierten Sentimentanalyse gestellt. Hierbei geht es einerseits um die Extraktion von Aspekten und sogenannten

[1] Dieses und sämtliche weitere Beispiele entstammen den Trainingsdaten des SemEval-Workshops 2016, Task 5, Subtask 1.

„Opinion Target Expressions“ (OTE) und andererseits um die Klassifikation ebendieser Aspekte. Der letzteren Aufgabe habe ich mich in dieser Arbeit zugewendet. Es handelt sich hierbei um eine Klassifikationsaufgabe, welche ich mithilfe eines Klassifikators aus dem Toolkit WEKA [Hall et al., 2009] löste. Die Ziele meiner Bachelorarbeit lauteten:

1. Aufbau einer Pipeline zur Generierung von Features auf Basis der für den SemEval-Workshop 2016[2] zur Verfügung gestellten Trainingsdaten.
2. Anreicherung der Trainingsdaten mit morphologischen, syntaktischen und semantischen Informationen unter Zuhilfenahme verschiedener Ressourcen
3. Trainieren und Evaluieren eines Klassifikators zur Erkennung der Targetpolarität auf Basis der erhaltenen Features mithilfe supervisierten Lernens.

In Kapitel 2 gebe ich einen Überblick über die Forschung in der aspektbasierten Sentimentanalyse und zeige den aktuellen Forschungsstand. In Kapitel 3 präsentiere und charakterisiere ich die für den SemEval-Workshop 2016 zur Verfügung gestellten Daten. In Kapitel 4 präsentiere ich die von mir aufgebaute Pipeline und die ihr zugrunde liegenden Werkzeuge, Ressourcen und die damit verbundenen Überlegungen. Darauf folgt in Kapitel 5 die Evaluation meines Systems und dessen Einordnung relativ zu den Ergebnissen anderer Systeme. Schlussendlich folgt in Kapitel 6 die Schlussfolgerung.

[2] `http://alt.qcri.org/semeval2016/` [26.11.2016].

2. Forschungsstand

Im folgenden versuche ich, den aktuellen Forschungsstand der aspektbasierten Sentimentanalyse, insbesondere der Aspektsentimentklassifikation, wiederzugeben. Unterkapitel 2.1 gibt einen kurzen Überblick über die Sentimentanalyse im Allgemeinen, Unterkapitel 2.2 widmet sich der aspektbasierten Sentimentanalyse im Speziellen. Unterkapitel 2.3 geht auf die Aspektextraktion ein, welche nötig ist, um Aspektsentimentklassifikation zu betreiben und Unterkapitel 2.4 beschäftigt sich schliesslich mit der letzteren.

2.1. Sentimentanalyse

Die Sentimentanalyse versucht, positive und negative Meinungen oder Gefühle in Texten zu finden sowie die Einheiten, auf welche diese bezogen sind, welche auch „Target“ oder „Opinion Target Expressions“ (OTE) genannt werden [Liu, 2015, 3]. Erste Forschungsarbeiten zum Thema wurden anfangs der 2000er Jahre publiziert. Der Begriff „sentiment analysis“ wurde vermutlich erstmals in [Nasukawa and Yi, 2003] verwendet [Liu, 2015, 1]. Hu and Liu [2004] veröffentlichten eine erste Arbeit, in der sie die aspektbasierte Sentimentanalyse definierten. Ein anderer Begriff für die Sentimentanalyse ist „Opinion Mining“. In der Wissenschaft werden oftmals beide Varianten verwendet, in der Industrie am häufigsten „Sentimentanalyse“ [Liu, 2015, 3].

Die Forschung zum Thema hat in den 2000er Jahren rasant an Umfang gewonnen [Liu, 2015, 3]. Zurückzuführen ist dies auf das Wachsen des Internetkonsums und das Aufkommen von sozialen Medien. Waren vorher nicht viele subjektive Texte digital verfügbar, wuchs deren Menge nun schnell an. Der Erfolg der sozialen Medien korreliert mit dem Wachstum der Sentimentanalyse. Menschen publizieren freiwillig ihre Meinung und diese Daten sind oftmals öffentlich einsehbar. Gerade bei Produktrezensionen werden den Meinungen von anderen Kunden eine höhere Glaubwürdigkeit attestiert als den Informationen des Verkäufers [Schouten and Frasincar, 2016, 1]. Firmen können die im Internet verfügbaren Meinungen verwenden, um neue Produkte zu entwickeln oder die Verkäufe anzukurbeln. Ebenfalls kommt den Meinungen

eine Relevanz auf den Finanzmärkten zuteil [Schouten and Frasincar, 2016, 1].
Die Sentimentanalyse befindet sich als aktives Forschungsfeld in der Verarbeitung natürlicher Sprache („natural language processing“ - NLP), wurde aber auch in anderen Disziplinen wie „Data Mining“, „Web Mining“ und „Information Retrieval“ studiert, da auch diese Forscher mit Textdaten umgehen [Liu, 2015, 1].
Nutzen bringt die Sentimentanalyse Unternehmen und Organisationen, welche an öffentlichen und Kundenmeinungen interessiert sind. Ebenfalls haben Regierungen Interesse daran herauszufinden, wie die Meinung der Bevölkerung zu diversen Themen steht. Einzelpersonen greifen ebenfalls gerne auf öffentliche Meinungen zurück, zum Beispiel auf Rezensionen eines Produktes, das sie kaufen möchten.
Heutzutage werden die sozialen Medien immer stärker in die Entscheidungsfindung miteinbezogen. Jedoch ist es eine schwierige Aufgabe, aus der Fülle von verfügbaren Informationen die wichtigen herauszufiltern. Hier bietet sich die automatische Verarbeitung durch die Sentimentanalyse an. Was wird überhaupt unter Meinung verstanden? Eine Meinung ist gemäss Duden eine „persönliche Ansicht, Überzeugung, Einstellung o.Ä., die jemand in Bezug auf jemanden, etwas hat (und sein Urteil bestimmt)“.[3] Im Englischen kann „opinion“ als Urteil oder Vorstellung definiert werden, welches oder welche nicht auf Tatsachen oder Beweisen beruht (vgl. thefreedicitonary.com[4]). Es handelt sich bei einer Meinung also immer um eine subjektive Aussage im Gegensatz zu objektiven Aussagen. Das Sentiment (engl. „sentiment“), übersetzbar als Gefühl oder Empfindung[5], stellt eine weitere Beurteilung einer Aussage dar: Enthält sie ein Sentiment oder nicht? Somit kann eine Aussage in vier Felder fallen: Subjektiv oder objektiv, sowie mit oder ohne Sentiment [Schouten and Frasincar, 2016, 2]. Zudem kann ein Sentiment, wie auch ein Target, implizit oder explizit sein. Da eine Fülle von menschlichen Emotionen besteht, werden oftmals Sentimentpolaritäten, im Gegensatz zu Emotionen, zur Beurteilung verwendet. Die Polarität beschreibt die Ausrichtung des Sentiments und ist positiv, negativ oder neutral (vgl. Pang and Lee [2008]). Das Finden eines Sentiments kann mit dem Quadrupel (s, g, h, t) definiert werden [Liu, 2015, 18]. Dabei ist s das Sentiment, g das Target, für welches das Sentiment ausgedrückt wird, h ist der Halter des Sentiments und t ist die Zeit, zu welcher das Sentiment geäussert wurde. In den meisten Ansätzen wird jedoch nur das Paar (s, g) gesucht (vgl. Liu [2015, 19].

[3] `http://www.duden.de/rechtschreibung/Meinung` [26.11.2016].

[4] `http://www.thefreedictionary.com/opinion` [26.11.2016].

[5] `http://dict.leo.org/` [26.11.2016].

2.2. Aspektbasierte Sentimentanalyse

Für die meisten Applikationen ist die Sentimentanalyse auf Dokumentenebene oder Satzebene nicht ausreichend. Es wird eine genauere Strukturierung benötigt, um relevante Schlüsse aus den Meinungen zu ziehen. Dies ist möglich mit der aspektbasierten Sentimentanalyse (ABSA). Hierbei werden nicht ganze Dokumente oder Sätze auf ihre Polarität untersucht, sondern Aspekte von Entitäten. Bei einer Entität könnte es sich beispielsweise um ein Restaurant handeln und bei einem Aspekt um das Essen, welches in diesem Restaurant serviert wird.

> "The food is great and the environment is even better."

In diesem Satz finden sich das Essen („food") und die Umgebung („environment") als Aspekte des Restaurants. Beide werden einzeln bewertet, in diesem Falle positiv. Manche Ansätze greifen auf eine Liste von zuvor festgelegten Aspekten zu, andere finden Aspekte frei im Text [Schouten and Frasincar, 2016, 2]. Die Definition von Meinungen gemäss Liu (vgl. 2.1) kann für die aspektbasierte Sentimentanalyse folgendermassen erweitert werden: (e, a, s, h, t) [Liu, 2015, 22]. Es entsprechen s, h und t der vorhergehenden Definition. Neu sind e und a anstelle von g. Sie entsprechen der Entität (e) respektive dem Aspekt der Entität (a).
Nach Liu [2015, 27] kann die Sentimentanalyse generell die folgenden acht Aufgaben enthalten: Entitätsextraktion und -resolution, Aspektextraktion und -resolution, Extraktion und Resolution des Halters der Meinung, Extraktion und Standardisierung des Zeitpunkts der Meinungsäusserung, Aspektsentimentklassifikation, Generierung des Meinungsquintupels, Extraktion und Resolution des Meinungsgrunds und Extraktion und Resolution von „opinion qualifiers". In den meisten Anwendungen werden aber nicht alle acht Schritte durchgeführt. Für die aspektbasierte Sentimentanalyse sind vor allem Entitäts- und Aspektextraktion sowie die Aspektsentimentklassifikation wichtig. Schouten and Frasincar [2016, 2] unterscheiden drei Verarbeitungsschritte in der aspektbasierten Sentimentanalyse (ABSA): Dies sind die Identifikation, Klassifikation und Aggregation (vgl. Tsytsarau and Palpanas [2012]): Sentiment-Target-Paare identifizieren, Sentiment klassifizieren und für jeden Aspekt die Sentimentwerte aggregieren, um eine detaillierte Übersicht zu erhalten. Die folgenden zwei Unterkapitel beschäftigen sich mit der Identifikation von Aspekten (Aspektextraktion) respektive mit deren Klassifikation (Aspektsentimentklassifikation).

2.3. Aspektextraktion

Für die vollständige Extraktion von Aspekten müssen einerseits die Aspekte selbst, sowie deren zugehörige Entitäten gefunden werden. Zusammen werden diese beiden Teilaufgaben auch „Opinion Target Extraction“ genannt [Liu, 2015, 137]. Ein Target ist ein Wort oder eine Phrase im Satz, an dem ein Aspekt einer Entität festgemacht werden kann. Es ist diejenige Phrase respektive dasjenige Wort über welches eine Meinung geäussert wird. Eine Meinung hat immer ein Target, auch wenn dieses nur implizit im Satz genannt wird. Gemäss Liu [2015, 138] haben sich vier unterschiedliche Ansätze zur Aspektextraktion herauskristallisiert:
Bei der frequenzbasierten Aspektextraktion werden nur explizit vorhandene Aspekte extrahiert. Dabei handelt es sich um Nomen oder um Nominalphrasen (NPs). Mithilfe eines POS-Taggers werden Nomen und NPs erkannt und mithilfe eines Data-Mining-Algorithmus die Auftrittswahrscheinlichkeiten berechnet. Diejenigen Substantive oder NPs, die mit einer höheren Frequenz als ein bestimmter Minimalwert auftreten, werden als Aspekte behalten, die restlichen verworfen. Dies funktioniert, da man davon ausgeht, dass Menschen beim Beschreiben von Entitäten ein ähnliches Vokabular verwenden (vgl. Liu [2015, 138]).
Eine weitere Möglichkeit zur Extraktion von Aspekten ist die Auswertung von syntaktischen Relationen. Hu and Liu [2004] approximieren mit ihrem System die Dependenzrelation zwischen dem Sentimentwort und einem Substantiv oder einer NP, welche vom Sentimentwort modifiziert wird [Liu, 2015, 141]. Zhuang et al. [2006] extrahieren Aspekt-Sentiment-Paare. Ihr System verwendet Dependenz-Templates aus Trainingsdaten und benutzt diese, um in den Testdaten gültige Aspekt-Sentiment-Paare zu finden. Andere Systeme verwenden lexikalisch-syntaktische Muster, welche die Entitäts- und Attributsrelationen kodieren, um Aspekte zu extrahieren.
Des Weiteren gibt es verschiedene Ansätze zur Aspektextraktion mithilfe des supervisierten Lernens. Hierbei kommen bespielsweise Hidden-Markov-Modelle (HMM) oder Conditional Random Fields (CRF) zum Einsatz.
Jin et al. [2009] verwenden zum Beispiel ein HMM, um Produktaspekte und Sentimentausdrücke in Rezensionen zu extrahieren. Dabei integrieren sie linguistische Features in das Modell. CRF wurden von Jakob and Gurevych [2010] verwendet, um Targets oder Aspekte aus Sätzen zu extrahieren, welche einen Sentimentsausdruck enthalten. Als Features haben sie beispielsweise Tokens, POS-Tags, Wortdistanzen und Dependenzpfade verwendet [Liu, 2015, 152].
Eine weitere Möglichkeit besteht in der Aspektextraktion mithilfe von Topic Modelling. Durch Topic Modelling können sowohl explizite wie auch implizite Aspekte in einem Schritt extrahiert werden [Liu, 2015, 159].

Topic Modelling ist ein Ansatz, mit welchem Themen („topics“) aus einem grossen Korpus von Textdokumenten extrahiert werden. Bei diesen Topics handelt es sich um Wortcluster. Die Ausgabe eines Topic Models sind eine Menge von solchen Wortclustern oder eine Topic-Distribution für jedes Dokument. Jeder Wortcluster ist eine Wahrscheinlichkeitsverteilung über Wörter im Korpus. Im Zusammenhang mit der Sentimentanalyse sind solche Topics Aspektkategorien. Jedes Wort in einem Wortcluster entspricht einem Aspektwort oder Aspektausdruck. Ein Vorteil der Topic Models ist es, dass sowohl explizite wie auch implizite Aspekte gefunden werden können. Hinzu kommt, dass die Aspekte gruppiert werden, was für die Sentimentanalyse von Nutzen sein kann.
Es gibt zwei grundsätzliche Topic Models: „probabilistic Latent Semantic Analysis“ (pLSA) und „Latent Dirichlet Allocation“ (LSA) [Liu, 2015, 159]. Bei beiden handelt es sich um unsupervisierte Methoden. Titov and McDonald [2008] haben als erste für die aspektbasierte Sentimentanalyse pLSA und LDA direkt auf ein Korpus von Rezensionen angewendet [Liu, 2015, 163].

2.4. Aspektsentimentklassifikation

Gemäss Liu [2015, 91] kennt die Aspektsentimentklassifikation zwei Hauptansätze: das supervisierte Lernen und den lexikonbasierten Ansatz. Diese sollen im Folgenden behandelt werden. Daneben gibt es auch Mischformen und weitere Ansätze, wie zum Beispiel unsupervisierte Lernverfahren.

2.4.1. Supervisiertes Lernen

Im Unterschied zu den dokument- und satzbasierten Ansätzen muss bei der Aspektsentimentklassifikation das „Opinion Target“ in die Feature-Generierung miteinbezogen werden, da festgestellt werden muss, auf was, also auf welchen Aspekt, sich eine Meinung bezieht. Ein erster Ansatz besteht darin, ein Set von Features zu generieren, welche abhängig sind vom Target. In dieser Hinsicht unterscheidet sich die Aspektsentimentklassifikation von der dokument- oder satzbasierten Klassifikation, bei welchen die Features vom Target unabhängig sind. Jiang et al. [2011] benutzen syntaktische Parsebäume, um vom Target abhängige Features zu extrahieren.
Eine andere Möglichkeit besteht darin, den Geltungsbereich jedes Sentimentausdrucks zu bestimmen und somit herauszufinden, ob das Target in diesem Bereich liegt, also vom Sentimentausdruck beeinflusst wird.
Andererseits kann die Distanz zwischen dem Target und jedem Wort-Feature berech-

net werden. Dies haben Boiy and Moens [2009] gemacht, wobei sie drei unterschiedliche Gewichte verwendet haben: Erstens die Tiefendifferenz, wobei das Gewicht umgekehrt proportional zur Differenz zwischen der Tiefe des Wortfeatures und der Entität im Parsebaum ist. Zweitens die Pfaddistanz, wobei das Gewicht eines Wortes umgekehrt proportional zur Länge des Pfades zwischen Feature und Entität im Parsegraphen ist. Das letzte Mass ist die simple Distanz, bei welcher das Gewicht eines Wortes umgekehrt proportional zu seiner Distanz zur Entität im Satz ist.
Yu et al. [2011] verwenden die Felder „pros“ und „cons“ von Rezensionen, um Sentimentterme zu finden. Dazu verwenden sie das Lexikon von Wilson et al. [2005] und die Information, ob das Wort im jeweiligen Kontext in „pros“ oder „cons“ vorkommt. Die gewonnenen Merkmale werden anschliessend verwendet, um eine Support Vector Machine (SVM) zu trainieren, welche in der Lage ist, Sentimentausdrücke als positiv oder negativ zu klassifizieren. In einer Rezension wird für jeden Aspekt derjenige Ausdruck gefunden, welcher sein Sentiment enthaltet. Anschliessend wird der SVM-Klassifikator verwendet, um die Polarität des Aspekts zu bestimmen.
Blair-Goldensohn et al. [2008] benutzen einen Maximum-Entropy-Klassifikator zusammen mit lexikonbasierten Methoden. Sie benutzen Informationen aus einem Lexikon als zusätzliche Features. Es werden die rohe Punktzahl aus dem Sentimentslexikon sowie davon abhängige Masse verwendet. Als weiteres Merkmal wird, falls vorhanden, die Sterne-Bewertung einer Rezension als weiterer Anhaltspunkt für die Klassifikation eingesetzt.
Lu et al. [2011] verwenden keinen binären Klassifikator (positiv oder negativ), sondern benutzen ein Support-Vector-Regression-Modell, um für Aspekte eine Sentimentbewertung zu erhalten. Damit ist es möglich, die Bewertung mit einer reellen Nummer zwischen 0 und 5 auszudrücken, was eine Annäherung an die oft verwendete Fünf-Stern-Bewertungsskala ist.
Vorteil des Ansatzes mit supervisiertem Lernen ist, dass der Algorithmus automatisch aufgrund von diversen Features lernen kann. Nachteil ist, dass das Verfahren auf Trainingsdaten angewiesen ist, welche manuell annotiert werden müssen. Des Weiteren ist es für das Lernverfahren schwierig, Phänomene zu lernen, welche nicht oft in den Trainingsdaten auftreten.

2.4.2. Lexikonbasierter Ansatz

Auch dieser Ansatz unterscheidet sich in der aspektbasierten Sentimentanalyse durch den Miteinbezug des Targets. Beim lexikonbasierten Ansatz wichtig ist vor allem eine Sentiment-Aggregierungs-Funktion. Zur Berechnung wichtig können auch die Distanzen zwischen Entitäten, respektive Aspekten, und polaren Ausdrücken sein.

Für jedes polare Wort gilt es, seinen jeweiligen Geltungsbereich („scope“) zu finden. Falls die Zielentität oder der Zielaspekt in diesem Bereich liegt, so kann diese Information dazu verwendet werden, die syntaktischen Beziehungen zwischen polaren Ausdrücken und Targets zu finden.
Ein reiner lexikonbasierter Ansatz ist derjenige von Ding et al. [2008]. Er basiert auf der Arbeit von Hu and Liu [2004]. In einem ersten Schritt werden Sentimentsausdrücke markiert. Dabei kann es sich um Phrasen oder Wörter handeln. Als nächstes werden Valenzshifter angewendet. Valenzshifter (auch „sentiment shifters“) sind Wörter oder Phrasen, welche Orientierung von Polaritäten ändern können [Liu, 2015, 93]. Zu diesen Shiftern gehören zum Beispiel Negationswörter wie „not“ oder „never“. Ein solcher Valenzshifter kann in einer Phrase wie „not good“ beispielsweise die Polarität umkehren. Die Polarität von „not good“ ist somit nicht positiv, sondern negativ. Als dritter Schritt werden „but“-Sätze behandelt. Es handelt sich um Sätze in der Form “X is Y, but A is B”. Oft zeigen solche „but“-Sätze an, dass die Polaritäten in den beiden Teilsätzen unterschiedlich sind [Liu, 2015, 94]. Dies muss jedoch nicht immer der Fall sein. Im Satz “Car-x is good, but car-y is better.”gibt es keinen Wechsel der Meinung [Liu, 2015, 94]. Schlussendlich werden die Sentiment-Scores berechnet. Hierbei wird eine Funktion verwendet, welche das „Schlussresultat“ des kompletten Satzes berechnet. Ist das Resultat positiv, so ist auch die Meinung zum Aspekt positiv.
Um diesen Algorithmus effektiver zu machen, schlägt Liu [2015, 95] vor, anstelle von Wortdistanzen den Geltungsbereich jedes einzelnen Sentimentausdrucks zu bestimmen. Dies kann anhand verschiedener Relationen zwischen Target und Sentimentausdruck geschehen. Einerseits können syntaktische Dependenzen verwendet werden. Meist handelt es sich hierbei um Adjektiv-Substantiv- oder Verb-Adverb-Dependenzen. In anderen Fällen kann ein Sentimentausdruck selbst als Target gewertet werden. Adjektive beschreiben meist spezifische Attribute oder Entitäten. So verweist „teuer“ beispielsweise auf den Aspekt „Preis“ und „schön“ beschreibt den Aspekt „Aussehen“. Schlussendlich können auch semantische Relationen analysiert werden. Hierbei handelt es sich aber um generell schwieriger zu erkennende Relationen.
Moghaddam and Ester [2010] benutzen ein Set von Adjektiven von Epinions.com, wobei jedes Adjektiv einer bestimmten Sterne-Bewertung zugeordnet ist. Unbekannte Adjektive werden mithilfe des Synonym-Graphen von WordNet bestimmt.
Zhu et al. [2009] segmentieren Sätze, falls mehrere Aspekte darin vorkommen. Somit kann die Polarität einzeln pro Aspekt bestimmt werden. Mithilfe eines Lexikons werden Aspekt-Polarität-Paare erstellt, welche die gesamte Polarität eines Aspekts in der Rezension wiedergeben.
In den meisten Fällen ist zur Erkennung von Relationen ein syntaktischer Parse-

baum vonnöten. Auch möglich ist die Kombination von supervisiertem Lernen mit der lexikonbasierten Methode, wie es beispielsweise Blair-Goldensohn et al. [2008] vorgenommen haben (vgl. weiter oben in 2.4.1).
Vorteil des lexikonbasierten Ansatzes ist, dass solche Systeme oft domänenunabhängig sind [Liu, 2015, 97]. Des Weiteren kann das System nach Belieben erweitert und verbessert werden. Auf der anderen Seite ist der Aufbau von Lexika, Mustern und Regeln sehr zeitintensiv.

3. SemEval-Task und -Daten

3.1. SemEval

Bei SemEval handelt es sich um den „International Workshop on Semantic Evaluation“, der in seiner heutigen Form das erste Mal 2007 durchgeführt wurde, ab 2012 jährlich. Bei diesem Workshop geht es um die computergestützte Semantikanalyse. Entstanden ist er aus dem Senseval-Workshop, der sich mit Wordbedeutungsevaluation beschäftigte. Dieser wurde 1998, 2001 und 2004 durchgeführt, im Jahre 2007 fand dann das erste Mal der SemEval-Workshop statt. Nach 2010 und 2012 fand SemEval schliesslich jährlich statt.
Beim SemEval-Workshop 2014 wurde das erste Mal eine Aufgabe zur aspektbasierten Sentimentanalyse gestellt. Hierbei mussten folgende Teilaufgaben gelöst werden:

1. Aspekttermextraktion
2. Aspektterm-Polaritätsschätzung
3. Aspektkategorie-Erkennung
4. Aspektkategorie-Polaritätsschätzung

Zur Verfügung standen Daten aus den beiden Domänen Restaurants und Laptops. Es wurden 163 Eingaben von 32 Teams registriert.
Im Jahr 2015 wurde die Aufgabe erneut gestellt. Dieses Mal wurde ein neues Set von Guidelines und Spezifikationen zu den Aspekten und Polaritäten eingehalten und diese einheitlich in Tupeln (Aspekt und Entität) codiert, um einen expliziteren Unterschied zwischen Entitäten und deren Aspekten sicherzustellen [Pontiki et al., 2015, 487]. Darüber hinaus wurde eine neue Teilaufgabe gestellt, in der es um „Out-of-domain-ABSA“ geht: Die Teilnehmer konnten ihr System in bisher nicht gesehenen Domänen testen.

3.2. Aufgabenstellung

Auch 2016 wurde die Aufgabe zur aspektbasierten Sentimentanalyse durchgeführt. Für diese wurden insgesamt 39 Datasets zur Verfügung gestellt, davon 19 für das Training und 20 für das Testing [Pontiki et al., 2016, 21]. Die Texte stammen nun aus sieben Domänen und sind in acht Sprachen gehalten: Englisch, Arabisch, Chinesisch, Niederländisch, Französisch, Russisch, Spanisch und Türkisch. Die Aufgabe (Task 5) beinhaltet drei Teilaufgaben: Aspektbasierte Sentimentanalyse auf Satzebene, aspektbasierte Sentimentanalyse auf Textebene und aspektbasierte Sentimentanalyse für Out-of-domain-Daten. Die erste Teilaufgabe ist in drei sogenannte Slots aufgeteilt. In Slot 1 sollen Aspektkategorien gefunden, in Slot 2 der Targetausdruck extrahiert und in Slot 3 die Sentimentpolarität bestimmt werden. Mit Slot 3 dieser Teilaufgabe habe ich mich in dieser Bachelorarbeit befasst. Hierbei ist der Targetausdruck bereits gegeben und für diesen muss anhand des zugehörigen Satzes entschieden werden, ob die Polarität des Targets positiv, neutral oder negativ ist.
Die 39 Datasets stammen aus sieben Domänen: Restaurants, Laptops, Mobiltelefone, Digitale Kameras, Hotels, Museen und Telekommunikation. Die ersten sechs bestehen ausschliesslich aus Kundenrezensionen, die Domäne Telekommunikation besteht aus Tweets.
Es wurden insgesamt 245 Eingaben von 29 Teams für die Aufgabe zur aspektbasierten Sentimentanalyse eingereicht. Die meisten Eingaben waren für Subtask 1 (216). Für Subtask 2 gab es 29 Eingaben.

3.3. Daten

Für den für meine Bachelorarbeit relevanten Subtask 1 (Slot 3) bestehen die Trainingsdaten aus je einem XML-Dokument für die Restaurantrezensionen und einem für die Laptop-Rezensionen. Die Restaurantdaten bestehen aus 350 Rezensionen mit insgesamt 2000 Sätzen. Eine Rezension besteht somit durchschnittlich aus 5.71 Sätzen. In diesen sind gesamthaft 2507 „Opinion Target Expressions“ (OTE) enthalten. 1151 Sätze beinhalten nur 1 OTE, 381 Sätze haben 2, 126 haben 3 und 50 haben 4 oder mehr OTE. In 627 Fällen ist die OTE „NULL“, das heisst, dass das Target nicht explizit im Satz genannt wird. 292 Sätze enthalten keine OTE. Da in meinen Experimenten nur der unmittelbare satzinterne Kontext betrachtet wird, spielen diese Sätze in der vorliegenden Arbeit keine Rolle. Unterschiedliche OTE gibt es insgesamt 722. Viele davon kommen nur einmal in den Daten vor, andere bis zu 38 mal.

Die Daten sind als XML formatiert. Jede Rezension ist ein eigener Knoten mit eindeutiger ID. Darin sind die Sätze erhalten, welche ebenfalls über eine eindeutige ID verfügen. In den Satzknoten ist jeweils im Element <text> der Inhalt des Satzes enthalten. Im darauf folgenden <Opinions>-Element sind die OTE, Aspektkategorie, Polarität sowie eine Referenz auf die Position im Text („von... bis...“) als Attribute eines <Opinion>-Knotens enthalten. Sind im Satz mehrere OTE zu finden, beinhaltet der <Opinions>-Knoten entsprechend mehrere <Opinion>-Elemente.

```
<Review rid="1123262">
    <sentences>
        <sentence id="1123262:0">
            <text>I think I've had some the best meals of my life at minnow.</text>
            <Opinions>
                <Opinion target="meals" category="FOOD#QUALITY" polarity="positive" from="31" to="36"/>
            </Opinions>
        </sentence>
        <sentence id="1123262:1">
            <text>The seafood is amazing, there's a good wine list, and the ever-changing menu always offers some great surprises.</text>
            <Opinions>
                <Opinion target="seafood" category="FOOD#QUALITY" polarity="positive" from="4" to="11"/>
                <Opinion target="wine list" category="DRINKS#STYLE_OPTIONS" polarity="positive" from="39" to="48"/>
                <Opinion target="menu" category="FOOD#STYLE_OPTIONS" polarity="positive" from="72" to="76"/>
            </Opinions>
        </sentence>
        <sentence id="1123262:2">
            <text>The combination of super-fresh ingredients in the dishes are unusual but really delicious.</text>
            <Opinions>
                <Opinion target="ingredients" category="FOOD#QUALITY" polarity="positive" from="31" to="42"/>
            </Opinions>
        </sentence>
        <sentence id="1123262:3">
            <text>Worth the trip from Manhattan.</text>
            <Opinions>
                <Opinion target="NULL" category="RESTAURANT#GENERAL" polarity="positive" from="0" to="0"/>
            </Opinions>
        </sentence>
    </sentences>
</Review>
```

Abbildung 1.: Beispiel für eine Rezension in den Restaurant-Trainingsdaten

Die Beschreibung des Aufbaus der Daten gilt auch für die Domäne der Laptops. In der Domäne „Laptops“ sind es 450 Rezensionen mit insgesamt 2500 Sätzen, die die Trainingsdaten bilden. Einziger Unterschied ist hier, dass keine OTE angegeben werden. Stattdessen werden nur die Entität-Apekt-Paare angegeben, von denen 2909 Vorkommen gezählt werden. 454 Sätze enthalten kein Entität-Aspektpaar.
Die englischen Trainingsdaten für die Domänen Restaurants und Laptops stammten von der SemEval-ABSA-Aufgabe von 2015. Die bestehenden Trainings- und Testdaten wurde mit kleineren Korrekturen zusammengeführt und nun als neues Trainings-Set verwendet. Für das Testen wurden neue Daten gesammelt und annotiert. Die Annotation der Restaurant-Daten wurde von einem Linguisten durchgeführt, die Annotation der Laptop-Daten von fünf Informatikstudenten. Beide Annotationen wurden anschliessend von einem zweiten Linguisten kontrolliert und, falls nötig, korrigiert. Konflikte wurden zwischen beiden Annotatoren kollaborativ gelöst [Pontiki et al., 2016, 23]. Verfügbar sind die Daten via META-SHARE[6].

[6] `http://metashare.ilsp.gr:8080/repository/search/?q=semeval+2016` [26.11.2016]

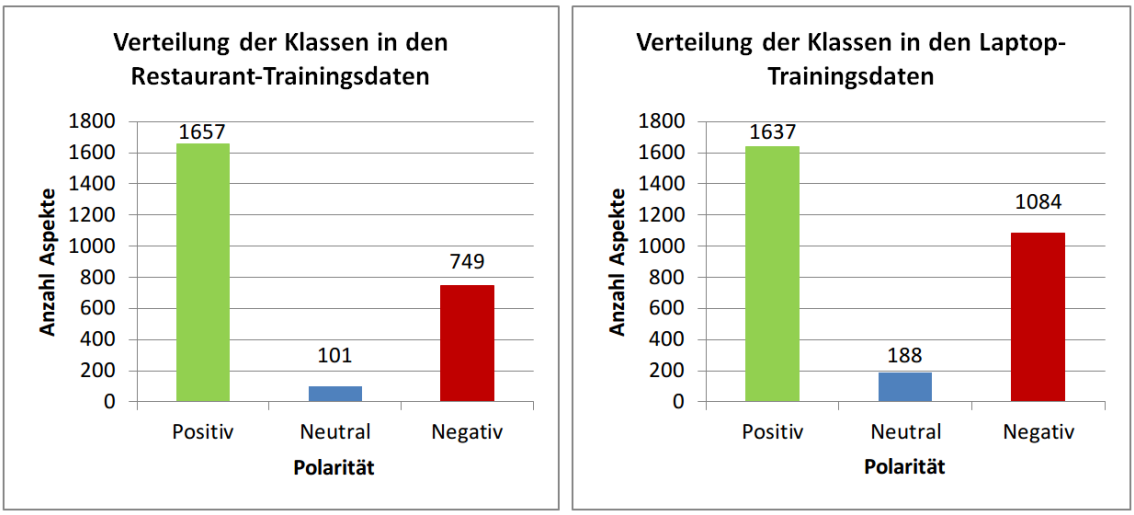

Abbildung 2.: Verteilung der Polaritäten in den Trainingsdaten

Im Folgenden wird auf das Annotationsschema eingegangen[7]. Dieses basiert auf demjenigen aus dem Jahre 2015. Bei den Restaurant-Daten wird das Entität-Aaspekt-Paar („category“), die „Opinion Target Expression“ (OTE), die Polarität sowie der Offset angegeben. Bei den Entitäten und Aspekten kann aus folgenden Möglichkeiten ausgewählt werden:

Entitäten der Domäne Restaurants:
FOOD, DRINKS, SERVICE, AMBIENCE, LOCATION, RESTAURANT.

Attribute der Domäne Restaurants:
GENERAL, PRICES, QUALITY, STYLE&OPTIONS, MISCELLANEOUS.

Bei den Laptop-Daten hingegen werden wie bereits erwähnt nur die Entität-Aspekt-Paare und die Polarität angegeben. Jedoch stehen bei den Laptops mehr Entitäten und Attribute zur Verfügung:

Entitäten der Domäne Laptops:
LAPTOP, DISPLAY, CPU, MOTHERBOARD, HARD_DISC, MEMORY, BATTERY, POWER_SUPPLY, KEYBOARD, MOUSE, FANS_COOLING, OPTICAL_DRIVES, PORTS, GRAPHICS, MULTIMEDIA_DEVICES, HARDWARE, OS, SOFTWARE, WARRANTY, SHIPPING, SUPPORT, COMPANY.

Aspekte der Domäne Laptops:

[7] vgl. die offiziellen Annotationsrichtlinien: `http://alt.qcri.org/semeval2016/task5/data/uploads/absa2016_annotationguidelines.pdf` [26.11.2016].

GENERAL, PRICE, QUALITY, OPERATION_PERFORMANCE, USABILITY, DESIGN_FEATURES, PORTABILITY, CONNECTIVITY, MISCELLANEOUS.

4. Klassifikationssystem

In meiner Pipeline greife ich auf verschiedene Ressourcen zurück: Tokenizer, Tagger und Lemmatizer des NLTK [Bird et al., 2009], den MaltParser [Nivre and Hall, 2005], das MPQA Subjectivity Clues Lexikon [Wilson et al., 2005] und das Opinion Lexicon von Bing Liu [Liu et al., 2005].

Nr.	**Name**	**Input**	**Output**
10	SentsExtractor	SemEval-XML	Sätze in Rohtext
11	NLTKTagger	Sätze in Rohtext	Sätze in CoNLL-Format getaggt und lemmatisiert
12	MaltParser	CoNLL-Format	CoNLL-Format inkl. Dependenzrelationen
20	ParseImporter	SemEval-XML und CoNLL-Datei	XML mit <parse>-Knoten
21	CoNLLtoXML	-	-
30	PolarityLookup	XML	XML mit <polar>-Attributen
40	FeatureExtractor	XML	ARFF-Datei
41	FeatureUtilities	-	-
42	FeatureFunctions	-	-

Tabelle 1.: Aufbau der Pipeline

4.1. Pre-processing

Die Pipeline ist in verschiedene Skripte aufgeteilt, die jeweils einen Arbeitsschritt übernehmen. Jedes Skript generiert einen eigenen Output, welcher zwischengespeichert wird. Somit können Verarbeitungsschritte einzeln ausgeführt werden, ohne die Outputs von anderen zu beeinflussen. Zudem wird so die Fehlersuche vereinfacht und es ist nicht nötig, bei einzelnen Änderungen die ganze Pipeline ablaufen zu lassen.
Die Pipeline beginnt damit, die für den SemEval-Workshop 2016 zur Verfügung gestellten Sätze aus den XML-Trainingsdateien zu extrahieren. Sie werden anschliessend zwischengespeichert in einer Textdatei, wobei eine Zeile einem Satz und eine

Textdatei einer XML-Datei entspricht.
Die gewonnenen Sätze werden anschliessend tokenisiert und getaggt. Ebenfalls werden sämtliche Wörter lemmatisiert. Diese Schritte werden durch Zuhilfenahme von Modulen des Natural Language Toolkits NLTK [Bird et al., 2009] durchgeführt. Der Output wird in das CoNLL-Format konvertiert.[8] Dies ist nötig, da der folgende Arbeitsschritt dieses Format voraussetzt. Die Sätze werden als nächstes geparst. Dies wird durch den Malt Parser [Nivre and Hall, 2005] übernommen. Der Parser fügt die Dependenzstruktur ein und ordnet Dependenzrelationen zu. Tabelle 2 zeigt einen kurzen Satz in CoNLL-Format und Erläuterungen zu den einzelnen Spalten.

ID	FORM	LEMMA	CPOSTAG	POSTAG	FEATS	HEAD	DEPREL	PHEAD	PDEPREL
1	Avoid	avoid	VB	VB	_	0	null	_	_
2	this	this	DT	DT	_	3	det	_	_
3	place	place	NN	NN	_	1	dobj	_	_
4	!	!	.	.	_	1	punct	_	_

ID Tokenzähler
FORM Wortform
LEMMA Lemma
CPOSTAG „coarse-grained“ POS-Tag (nicht verwendet und entspricht deshalb POSTAG)
POSTAG POS-Tag
FEATS Syntaktische oder morphologische Merkmale (nicht verwendet)
HEAD Kopf dieses Tokens (immer ID-Wert des Kopfes oder 0 im Fall, dass dieses Token Kopf ist)
DEPREL Dependenzrelation zum Kopf (ROOT wenn HEAD=0)
PHEAD Projektiver Kopf (nicht verwendet)
PDEPREL Dependenzrelation zu PHEAD (nicht verwendet)

Tabelle 2.: Beispiel des Satzes “Avoid this place!” in CoNLL-Format

In einem nächsten Schritt wird die so erhaltene CoNLL-Struktur in einem eigenen Knoten in die ursprüngliche XML-Datei eingefügt. Pro Token wird ein <w>-Knoten erstellt, welcher die CoNLL-Struktur als Attribute und das Token als Inhalt enthält. Anschliessend werden polare Wörter markiert. Die Informationen zur Polarität werden als Attribute in die <w>-Knoten aufgenommen. Die Informationen zur Polarität von Wörtern stammen aus zwei Lexika. Das erste sind die „MPQA Subjectivity Clues“. Es handelt sich hierbei um ein Polaritätslexikon, welches in Wilson et al. [2005] beschrieben wird. Das Lexikon enthält über 8000 „subjectivity clues“. Unter diesem Begriff werden Wörter und Phrasen verstanden, welche eine subjektive Verwendung aufweisen können [Wilson et al., 2005, 349]. Das vorliegende Lexikon enthält nur aus einem Wort bestehende Einträge. Es sind sowohl positive und negative, als auch neutrale Wörter vorhanden.
Das zweite Lexikon ist das „opinion lexicon“ von Bing Liu. Gemäss Webseite[9] wurde

[8] `http://ilk.uvt.nl/conll/#dataformat` [26.11.2016]

[9] `https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html` [26.11.2016].

es seit 2004, angefangen mit Hu and Liu [2004], kontinuierlich aufgebaut. Es besteht aus zwei Wortlisten: Eine davon enthält Wörter mit positiver Polarität, die andere Wörter mit negativer Polarität. Insgesamt enthalten die Listen ca. 6800 Wörter. Ebenfalls enthalten sind Wörter, welche bewusst falsch geschrieben sind. Diese sollen in Online-Texten häufig vorkommen. Somit ist die Erkennung von Polaritäten auch im Falle dieser falsch geschriebenen Wörter möglich.
Als letzter Vorbereitungsschritt werden Sätze mit mehr als einem Target behandelt: Diese werden dupliziert und jeder Instanz ein einziges Target zugeordnet. So wird sichergestellt, dass die Sätze für jedes Target separat analysiert werden können. Zudem werden die Wörter, die im Target enthalten sind, im <w>-Knoten als Target markiert, damit diese Information für künftige Schritte zur Verfügung steht.

4.2. Featureextraktion

Insgesamt werden 32 Features aus den Daten extrahiert. Sie sind in folgende Kategorien einteilbar:

1. Zahlen und Häufigkeiten
   Hier werden einfache Auftreten und Häufigkeiten gezählt. Dazu gehören die Anzahl positive sowie negative Wörter. Weiter wird auch die Satzlänge verwendet. Basierend auf den vorhergehenden Features werden ebenfalls das Verhältnis von positiven zu negativen Wörtern und das Verhältnis von positiven respektive negativen Wörtern zur Satzlänge berechnet.

2. Binäre Features
   Diese Features werden auf 1 gesetzt, wenn ein bestimmter Sachverhalt im Satz gegeben ist. So gibt es ein Flag, für positive und negative Wörter: Enthält der Satz mindestens ein positives respektive negatives Wort, so wird das entsprechende Flag auf 1 gesetzt. Für die fünf häufigsten in den Trainingsdaten vorkommenden Dependenzrelationen besteht ebenfalls ein Flag. Diese sind „nsubj“, „det“, „prep“, „pobj“ und „amod“. Kommt eine solche Dependenzrelation im Satz vor, wird das entsprechende Flag auf 1 gesetzt.

3. Wörter vor und nach dem Target
   Diese Features beschreiben die Umgebung des Targets. Es werden die POS-Tags des dem Target vorhergehenden und folgenden Wortes extrahiert. Ebenfalls werden diese angrenzenden Wörter auf ihre Polarität hin untersucht: Sind diese polar, so wird die Polarität als Feature gewertet, ansonsten wird das jeweilige Wort als nicht-polar klassifiziert.

Nr.	Feature	Datentyp	Anzahl Werte
l01	getPositiveWords	Integer	1
l02	getNegativeWords	Integer	1
l03	getPosNegWordRatio	Float	1
l04	getSentLength	Integer	1
l05	getPosWordToSentRatio	Float	1
l06	getNegWordToSentRatio	Float	1
l07	hasPosWord	Boolean	1
l08	hasNegWord	Boolean	1
c01	getPrevPOSTag	String	1
c02	getNegPOSTag	String	1
c03	getPrevPolJJ	String	1
c04	getNextPolJJ	String	1
c05	getCopulaRelation	String	1
c06	deprelFeats	Boolean	5
c07	polDepRel	String	9
c08	punctBetweenTargetPol	Boolean	1
t01	getTargetPolarity	String	1
t02	getTargetPosition	Integer	1
t03	getTargetHeadRelation	String	1
t04	getTargetPolDistance	Integer	1
n01	hasNeg	Boolean	1
n02	negDepRel	String	1

Tabelle 3.: Extrahierte Features

Ein weiteres Feature dieser Kategorie untersucht die Sätze auf folgende Konstruktion: „Target (VBD od. VBZ) X* JJ“, also Target, gefolgt von Hilfsverb, eventuell gefolgt von einer beliebigen Anzahl beliebiger Wörter, gefolgt von einem polaren Adjektiv. Das Feature erhält den Polaritätswert dieses Adjektivs, ansonsten wird mit NONE das Nichtvorhandensein dieser Konstruktion oder das Fehlen von Polarität markiert.

4. Distanzen
   Zu diesen Features gehört die Position des Targets im Satz. Ausserdem wird die Distanz vom Target zum ersten polaren Wort gemessen. Dabei sind vier Kategorien möglich: „near“, „middle“, „far“, „ultra“, wobei „near“ einem Abstand von bis zu zwei Wörtern, „medium“ von drei bis zu fünf Wörtern, „far“ von sechs bis zu acht Wörtern und „ultra“ von mehr als acht Wörtern entspricht.

5. Dependenzrelationen
   Ein Feature dieser Gruppe ist die Dependenzrelation des Kopfs des Targets. Weiter werden die Dependenzrelationen von maximal 9 polaren Wörtern ex-

trahiert, drei pro Polaritätsklasse. Sind weniger als 9 polare Wörter vorhanden, so werden die übrigen Stellen mit „NONE“ angegeben.

6. Negationen
   Hierbei kommt ein binäres Feature zum Zuge. Ein Flag wird auf 1 gesetzt, falls im Satz ein Wort mit Dependenzrelation „neg“ vorkommt. Ein weiteres Feature ist die Suche nach dem Kopf eines Wortes mit Dependenzrelation „neg“. Handelt es sich hierbei um ein polares Wort, so enthält das Feature die umgekehrte Polarität dieses Wortes.

Schlussendlich wird die Polarität des Aspekts („positive“, „neutral“ oder „negative“) hinzugefügt. Extrahiert wird diese aus dem XML-Knoten in den Trainingsdaten, welcher neben Entität-Aspekt-Paaren und Targets des Satzes auch die Targetpolarität beinhaltet.

4.3. Klassifikation

Zur Klassifikation habe ich das WEKA-Toolkit verwendet [Hall et al., 2009]. Beim „Waikato Environment for Knowledge Analysis“ handelt es sich um eine Sammlung von Algorithmen für Data Mining der „Machine Learning Group“ der Universität Waikato. Einerseits können Daten direkt, via Kommandozeile oder grafische Nutzeroberfläche, oder durch Aufruf via Java-Code verarbeitet werden. WEKA unterstützt diverse Aspekte des Data Minings und des maschinellen Lernens: Preprocessing, Clustering, Klassifikation, Regression, Visualisierung und Featureselektion. Ich verwende WEKA, da es einfach und schnell eingesetzt werden kann. Die grafische Nutzeroberfläche liefert nützliche Einstellungsmöglichkeiten und Visualisierungen für Training und Testing. Es können verschiedene Klassifikationsmethoden gut evaluiert und verglichen werden.
WEKA verlangt die zu verarbeitenden Daten als sogenannte ARFF-Dateien („Attribute-Relation File Format“). In einer ARFF-Datei wird jeweils eine Relation gespeichert, im vorliegenden Fall die Relation „Polarität“. Im Header werden der Name der Relation und die nötigen Attribute sowie deren Namen und Typ spezifiziert, in meinem Fall die in 4.2 beschriebenen Features. Im Abschnitt „data“ folgt dann pro Zeile eine Instanz. Es handelt sich pro Zeile um sämtliche Attributwerte zu einer einzelnen „Opinion Target Expression“ (OTE), jeweils durch Kommata getrennt.
Für die Klassifikation stehen diverse maschinelle Lernverfahren zur Verfügung. In dieser Arbeit habe ich verschiedene Klassifikatoren ausprobiert. Kapitel 5.2 enthält die Ergebnisse dieses Vergleichs.

5. Evaluation

Die Evaluation wurde beim SemEval-Workshop in zwei Phasen durchgeführt. In Phase A mussten die Teilnehmer die Aspektkategorien und die „Opinion Target Expressions“ (OTE) erkennen. Dies entspricht Subtask 1 Slot 1 und 2 (vgl. 3.2). In Phase B wurden die Goldannotationen für diese Teile herausgegeben und anschliessend mussten die Sentimentpolaritätswerte, also Slot 3 (vgl. 3.2), ausgegeben werden. 28 Systeme wurden für die Sentimentpolaritätklassifikationsaufgabe (Slot 3) gemäss Liste von Pontiki et al. [2016, 25] ausgewertet.
Da ich mich in meiner Arbeit mit Slot 3 befasst habe, verwende ich zur Evaluation meines Klassifikationssystems die SemEval-Golddaten aus Phase B. In einem ersten Schritt (Unterkapitel 5.1) habe ich die Resultate meines Systems auf den Trainingsdaten analysiert. In Unterkapitel 5.2 zeige ich die Ergebnisse des Vergleichs verschiedener Klassifikationsmethoden in WEKA auf und zeige, warum meine Wahl des Klassifikators auf „SimpleLogistic“ gefallen ist. Anschliessend führe ich in Unterkapitel 5.3 einen Vergleich meines Systems mit einem „Bag-Of-Words“-Ansatz durch, welcher zeigt, dass mein System domänenunabhängig funktioniert. In Unterkapitel 5.4 evaluiere ich mein Klassifikationssystem auf den für den SemEval-Workshop 2016 zur Verfügung gestellten Testdaten. Schlussendlich folgt eine Diskussion sowie ein Vergleich der Resultate beim SemEval-Workshop 2016 (Unterkapitel 5.5).

5.1. Ergebnisse des SimpleLogistic-Klassifikators auf den Trainingsdaten

Während der Entwicklung meines Systems evaluierte ich dessen Leistung anhand einer 10-fachen Kreuzvalidierung auf den Trainingsdaten der Restaurantdomäne. Als Baseline diente mir ein hypothetischer Klassifikator, der sämtliche Targets als der Mehrheitsklasse zugehörig klassifiziert. Die Trainingsdaten umfassten 2507 Targets. Davon gehören 1657 der Klasse positiv an. Es handelt sich hierbei um die Mehrheitsklasse, womit der Baseline-Klassifikator sämtliche Targets als positiv gewertet hat. Dies ergibt eine Genauigkeit von 66.09%. In Tabelle 4 ist ersichtlich, wie mein Sys-

tem, unter Verwendung des SimpleLogistic-Klassifikators von WEKA, im Vergleich zu dieser Baseline abgeschnitten hat.

System	**Genauigkeit in Prozent**
Baseline - Mehrheitsklasse	66.0949
WEKA - SimpleLogistic	76.5058

Tabelle 4.: Genauigkeit des Systems im Vergleich zur Baseline

Meinem System war es möglich, diese Baseline um über 10% zu übertreffen.

5.2. Vergleich verschiedener Klassifikatoren

Während dem Aufbau meines Systems habe ich verschiedene Klassifikatoren ausprobiert, welche in WEKA zur Verfügung stehen. Tabelle 5 zeigt eine Übersicht zu diesen Klassifikatoren. Sämtliche wurden auf den definitiven 32 Features trainiert, welche auch für das Training des SimpleLogistic-Klassifikators verwendet wurden. Zur Evaluation wurde für jeden Klassifikator 10-fache Kreuzvalidierung auf den Traingsdaten verwendet.

Klassifikator	**Positiv**			**Neutral**			**Negativ**			**Acc.**[4]
	P[1]	R[2]	F_1[3]	P[1]	R[2]	F_1[3]	P[1]	R[2]	F_1[3]	
SimpleLogistic	0.805	0.882	0.842	0.200	0.010	0.019	0.663	0.607	0.634	76.5058
RandomForest	0.804	0.881	0.841	0.243	0.089	0.130	0.668	0.583	0.623	76.0271
NaiveBayes	0.839	0.806	0.822	0.125	0.010	0.018	0.604	0.732	0.662	75.1496
BayesNet	0.839	0.798	0.818	0.083	0.010	0.018	0.594	0.729	0.655	74.5911
IBk knn=5	0.773	0.896	0.830	0.167	0.020	0.035	0.655	0.502	0.568	74.2720
DecisionTable	0.780	0.884	0.829	0.000	0.000	0.000	0.638	0.531	0.580	74.2720
J48	0.815	0.829	0.822	0.500	0.010	0.019	0.595	0.652	0.622	74.2720
Logistic	0.808	0.844	0.826	0.066	0.050	0.056	0.636	0.595	0.615	73.7535
RandomTree	0.782	0.817	0.799	0.138	0.129	0.133	0.566	0.515	0.539	69.8843

[1] Precision, [2] Recall, [3] F_1-Measure, [4] Genauigkeit in Prozent

Tabelle 5.: Vergleich verschiedener Klassifikatoren

Der SimpleLogistic-Klassifiaktor basiert auf linearen Logitmodellen (vgl. Landwehr et al. [2005]). Verwendet wird LogitBoost mit einfachen Regressionsfunktionen als Basis-Lerner, um die Logitmodelle anzupassen. Um die optimale Anzahl LogitBoost-Iterationen herauszufinden, wird kreuzvalidiert. Dadurch wird automatisch eine Featureselektion vorgenommen. Beim Ausprobieren hat sich herausgestellt, dass der SimpleLogistic-Klassifikator mit den verwendeten Features die beste Genauigkeit

erreicht. Mitunter durch die automatische Featureselektion kann er auf den Trainingsdaten gute Ergebnisse erzielen. In der Mehrheitsklasse, den positiven Aspekten, kann der Klassifikator 1462 Instanzen korrekt klassifizieren, was 88.2% Recall entspricht. Auch erreicht der SimpleLogistic-Klassifikator im vorliegenden Vergleich mit 0.842 das höchste F-Mass in der positiven Klasse. In der negativen Klasse erkennt er 455 Instanzen, was 60.8% Recall entspricht. Für die neutrale Klasse findet er nur einen Aspekt, womit er nur ca. 1% aller neutralen Sätze korrekt klassifiziert. Knapp zwei Drittel der neutralen Sätze werden als positiv gewertet, die restlichen als negativ. Zurückzuführen sind diese Unterschiede nach Klassen auf die Unausgewogenheit der Trainingsdaten, was das Verhältnis zwischen positiven, neutralen und negativen Aspekten betrifft (vgl. Abb. 2 in Unterkapitel 3.3).
Am zweitbesten ist die Leistung des RandomForest-Klassifikators. Dieser basiert, entsprechend seinem Namen, auf einem „Wald“ von „Random Trees“ (vgl. Breiman [2001]). Pro „Random Tree“ wird ein Baum konstruiert, bei welchem für jeden Knoten k zufällig gewählte Attribute beachtet werden. Mit 69.88% ist ein einzelner „Random Tree“ wesentlich schlechter als die „Forest-Variante“ mit mehreren Bäumen, welche eine Genauigkeit von 76.03% erreicht. Im Gegensatz zum SimpleLogistic- erkennt der RandomForest-Klassifikator acht neutrale Aspekte mehr. Für die neutrale Klasse ist dies das zweitbeste Resultat in diesem Vergleich, einzig der RandomTree-Klassifikator konnte noch mehr neutrale Aspekte korrekt klassifizieren. In diesem Vergleich erreicht der RandomTree-Klassifikator mit 66.8% die höchste Präzision in der negativen Klasse.
Das drittbeste Resultat wird mit dem NaiveBayes-Klassifikator erzielt. Dabei erreicht er mit 73.2% korrekt klassifizierten Aspekten den besten Recall in der negativen Klasse. Rund 0.5 Prozentpunkte schlechter ist die Genauigkeit des BayesNet-Klassifikators, welcher auf einem Bayes-Netzwerk beruht[10]. Die Genauigkeit sinkt im Vergleich zum NaiveBayes-Klassifikator in allen Klassen. Die Präzision der beiden Klassifikatoren in der positiven Klasse ist mit 83.9% die höchste in diesem Vergleich. Allerdings stehen diesen Werten niedrigere Recalls im Vergleich zum SimpleLogistic-Klassifikator gegenüber. In der negativen Klasse erreichen die beiden Bayes-Klassifikatoren höhere F-Masse als der SimpleLogistic-Klassifikator.
IBk ist die WEKA-Implementation eines „k-nearest neighbour“-Klassifikators. Es handelt sich um einen „lazy“ Klassifikator, welcher die k-nächsten Nachbarn im Featurespace betrachtet, im vorliegenden Fall die nächsten fünf Nachbarn. Es handelt sich um „Instance learning“, bei welchem die Trainingsdaten nicht vor, sondern erst während des Klassifizierens verrechnet werden (vgl. Aha et al. [1991]). Dieser Klassifikator bewertet mit 1484 Instanzen 89.6% der positiven Aspekte korrekt, womit

[10] Angabe in WEKA: `http://www.cs.waikato.ac.nz/~remco/weka.pdf`. Zugriff nicht verfügbar [26.11.2016].

er in der positiven Klasse einen höheren Recall als der SimpleLogistic-Klassifikator, jedoch eine tiefere Präzision aufweist.
Mit dem Decision-Table-Klassifikator wird ein simpler Entscheidungstabellenklassifikator aufgebaut, welcher mit Mehrheitsentscheid funktioniert. In der Klasse der positiven Aspekte liefert er mit einer Genauigkeit von 88.35% ein ähnliches Resultat wie der SimpleLogistic-Klassifikator, ist aber schlechter in der negativen Klasse. Neutrale Aspekte konnte er keine erkennen. Diese beiden Sachverhalte können wohl auf die Funktionsweise des Klassifikators mit Mehrheitsentscheid zurückgeführt werden, aufgrund der Unausgewogenheit der Klassen in den Trainingsdaten.
J48 ist die Implementation von C4.5-Entscheidungsbäumen (vgl. Quinlan [1993]). Es sind sowohl beschnittene als auch unbeschnittene Entscheidungsbäume möglich („pruned“ respektive „unpruned“). „Pruning“ wird eingesetzt, um Overfitting zu vermeiden, weshalb in diesem Fall beschnittene Bäume eingesetzt werden. In der negativen Klasse erreicht dieser Klassifikator einen besseren Recall als der SimpleLogistic-Klassifikator, in der neutralen Klasse ist er in etwa gleichauf, kann aber nur weniger gute Ergebnisse für die positive Klasse aufweisen.
Der Logistic-Klassifikator basiert auf einem multinominalen logistischen Regressionsmodell und verfügt über einen „Ridge Estimator“ (vgl. Le Cessie and Van Houwelingen [1992]). Es handelt sich um eine komplexere Implementierung als die des SimpleLogistic-Klassifikators. Entsprechend rechen- und zeitintensiv ist das Training. Der Logistic-Klassifikator erzielt nur in der neutralen Klasse, mit rund 5% korrekt klassifizerten neutralen Aspekten, bessere Ergebnisse als der SimpleLogistic-Klassifikator. In den anderen Klassen produziert er schlechtere Ergebnisse.
Insgesamt weist der RandomTree-Klassifikator die schlechteste Genauigkeit der verglichenen Methoden auf. Auffällig ist jedoch, dass er mit 13 korrekt klassifizierten Aspekten der neutralen Klasse für diese die besten Präzisions-, Recalls- und F-Mass-Werte liefert. Dies wird auf die Funktionsweise des RandomTree-Klassifikators zurückzuführen sein, da durch die arbiträre Auswahl von Attributen an den Knoten ein Vorteil für die in den Trainingsdaten untervertretene Klasse der neutralen Aspekte entstehen könnte.

5.3. Experiment zur Domänenunabhängigkeit des Klassifikationssystems

Ich wollte herausfinden, ob mein Klassifikator domänen-unabhängig ist, da domänenunabhängige Systeme den Vorteil haben, auf Daten aus beliebigen Domänen angewendet werden zu können. Dies sollte eigentlich der Fall sein, da ich in meinem Sys-

tem ausschliesslich domänen-unabhängige Ressourcen verwende. Wäre mein System domänen-abhängig, so müsste bei der Anwendung auf einer anderen Domäne die Leistung stark sinken, da es auf diese Domäne nicht angepasst ist. Ein domänen-unabhängiges System sollte hingegen auf verschiedenen Domänen in einem ähnlichen Bereich abschneiden.
Um diese These zu testen, habe ich meinen Klassifikator mit einem Naive-Bayes-Klassifikator verglichen, welcher mit Mallet trainiert wurde. Mallet ist ein Java-Package für die statistische Verarbeitung natürlicher Sprache, Dokumentklassifikation, Clustering, Topic Modelling, Informationsextraktion und andere Anwendungen des maschinellen Lernens [McCallum, 2002]. Mallet verwendet einen „Bag-Of-Words"-Ansatz, um aufgrund der Trainingsdaten Features zu generieren, womit die Klassifikation von Dokumenten vorgenommen werden kann. Zum Klassifizieren stehen zum Beispiel Naive Bayes, Maximum Entropy oder Decision Trees zur Verfügung. Mallet verarbeitet Dokumente zu Features. In meinem Fall entspricht ein „Dokument" genau einem Satz einer Rezension. Darin enthaltene Sätze werden tokenisiert und die Tokens zu einem „Bag-Of-Words" hinzugefügt. Für jeden Satz wird ein solcher „Bag-Of-Words" generiert, welcher als Features sämtliche Tokens beinhaltet. Kommt ein Token im Satz vor, so wird das Flag dieses Tokens auf „1" gesetzt. Die Flags von Tokens, die nicht im Satz vorkommen, bleiben auf „0" gesetzt. Mallet verwendet diese „Bags-Of-Words" anschliessend, um einen Klassifikator zu trainieren. Der „Bag-Of-Words" wird nur anhand der Trainingsdaten erstellt. Angesichts dessen handelt es sich bei Mallet um ein domänenspezifisches System. Denn würden zum Training Daten einer anderen Domäne verwendet, so würde sich auch das Vokabular im „Bag-Of-Words" unterscheiden. Wenn davon ausgegangen wird, dass Daten der gleichen Domäne ähnliche Wörter verwenden, muss angenommen werden, dass bei der Anwendung auf Daten einer anderen Domäne die Leistung des Systems sinken würde.

Für die Verwendung von Mallet habe ich die Sätze der Trainingsdaten als einfache Zeichenketten aus den XML-Dateien extrahiert und in separaten Textdateien sortiert nach Klasse (positiv, neutral, negativ) zur Verarbeitung durch Mallet gespeichert. Ein Satz entspricht einem Mallet-„Dokument". Diese Verarbeitung habe ich mit allen vier Dateien vorgenommen: Restaurants Training und Test, sowie Laptops Training und Test. Mit den jeweiligen Trainingsdaten habe ich je einen Naive-Bayes-Klassifikator trainiert.
Mit WEKA bin ich folgendermassen vorgegangen: Bei den Daten zur Domäne „Laptops" sind die „Opinion Target Expressions" (OTE) nicht annotiert – es sind also bloss die Aspektkategorien und die zugehörige Polarität vorhanden. Aus diesem Grund funktionieren diejenigen Features meiner Pipeline, welche Bezug auf die

Nr.	Feature	Datentyp	Anzahl Werte
l01	getPositiveWords	Integer	1
l02	getNegativeWords	Integer	1
l03	getPosNegWordRatio	Float	1
l04	getSentLength	Integer	1
l05	getPosWordToSentRatio	Float	1
l06	getNegWordToSentRatio	Float	1
l07	hasPosWord	Boolean	1
l08	hasNegWord	Boolean	1
c06	deprelFeats	Boolean	5
c07	polDepRel	String	9
t01	getTargetPolarity	String	1
n01	hasNeg	Boolean	1
n02	negDepRel	String	1

Tabelle 6.: Verwendete Features für den WEKA-Mallet-Vergleich

„Opinion Target Expression“ (OTE) nehmen, nicht auf den Laptop-Daten. Um den Vergleich mit dem Mallet-Klassifikator trotzdem vornehmen zu können, habe ich dementsprechend diese Features in diesem Experiment nicht verwendet. Ohne die targetspezifischen sind es insgesamt noch 25 Features. Dies galt sowohl für das Training der Restaurant- wie auch der Laptop-Daten.
Ich trainierte einen SimpleLogistic-Klassifikator in WEKA mit den Restaurant-Trainingsdaten und einen mit den Laptop-Trainingsdaten. Tabelle 6 gibt eine Übersicht über die Features. Anschliessend habe ich den „Restaurant-Klassifikator“ auf den Laptop-Testdaten und umgekehrt den „Laptop-Klassifikator“ auf den Restaurant-Testdaten evaluiert. In Mallet habe ich das gleiche gemacht. Tabelle 7 zeigt die Ergebnisse dieses Experiments.

Trainingsdaten	Testdaten	SimpleLogistic WEKA	Naive Bayes Mallet
Restaurants	Restaurants	71.013	75.437
Restaurants	Laptops	70.736	53.059
Laptops	Laptops	65.293	69.538
Laptops	Restaurants	70.314	52.270

Tabelle 7.: Vergleich der Genauigkeit zwischen WEKA und Mallet (in Prozent)

Aus Tabelle 7 ist ersichtlich, dass die Genauigkeit des WEKA-Klassifikators auf meinen Features in allen Fällen im ungefähr gleichen Bereich liegt. Beim „Restaurant-Klassifikator“ beträgt der Unterschied der Genauigkeit zwischen Test auf Restaurant-

Daten und Test auf Laptop-Daten nur 0.277 Prozentpunkte. Somit kann davon ausgegangen werden, dass die Domäne der Trainingsdaten keinen Einfluss auf die Performanz hat. Beim „Laptop-Klassifikator“ fällt das Resultat in der Domäne mit 65.293% ein wenig niedriger aus. Beim Test des „Laptop-Klassifikators“ auf den Restaurant-Testdaten fällt jedoch auf, dass das Resultat sogar besser als auf den domäneninternen Daten ausfällt. Darum kann auch hier davon ausgegangen werden, dass die Domäne keinen Einfluss auf die Resultate hat.
Beim „Restaurant-Klassifikator“ von Mallet ist ein sehr grosser Leistungsabfall ersichtlich, wenn der Test auf den domäneninternen Restaurant-Daten mit dem Test auf den Laptop-Daten verglichen wird: die Genauigkeit mit den Laptop-Testdaten ist 22.378 Prozentpunkte tiefer. Dies zeigt, dass die Domäne hier einen grossen Einfluss auf das Resultat hat. Beim „Laptop-Klassifikator“ von Mallet ist ebenfalls eine relevante Differenz von 17.268% zu erkennen. Auch hier wirkt sich die Domäne auf das Resultat aus.
Es zeigt sich somit, dass mein Klassifikator domänenunabhängig funktioniert. Dies bringt den Vorteil, dass er auf Daten verschiedener Domänen angewendet werden kann, im Gegensatz zu domänenspezifischen Systemen, bei welchen auf anderen Domänen eine Verminderung der Performanz zu erwarten ist.

5.4. Evaluation des Klassifikationssystems auf den Testdaten

Eine Baseline wurde von den Organisatoren des SemEval-Workshops 2016 zur Verfügung gestellt. Diese besteht aus einem Support-Vector-Machine-Klassifikator (SVM) (vgl. Pontiki et al. [2016, 24]). Als Features werden n Unigramme verwendet, welche aus demjenigen Satz extrahiert werden, in dem das Entität-Aspekt-Paar und die „Opinion Target Expression“ (OTE) vorkommen. Zudem wird für das Entität-Aspekt-Paar ein Integerwert verwendet[11]. Dieser wird entsprechend des Entität-Aspekt-Paares eines Satzes ebenfalls als Feature verwendet. Schlussendlich wird das Label „positiv“, „neutral“ oder „negativ“ hinzugefügt. Der Klassifikator wurde auf den Trainingsdaten trainiert und auf den Testdaten evaluiert. Ergeben hat dies eine Genauigkeit von 76.484% [Pontiki et al., 2016, 24]. Da der Baseline-Klassifikator ausschliesslich auf Daten aus dem Trainingsset zurückgreift, handelt es sich dabei um ein domänenabhängiges System.

[11] Jedem möglichen Entität-Aspekt-Paar wird zuvor eine eindeutige Zahl zugeordnet, welche dieses explizit identifizert. Beispielsweise „1“ für „RESTAURANT#GENERAL“, „2“ für „RESTAURANT#AMBIENCE“, etc.

Aspektklasse	**Training**		**Test**	
	Anzahl Aspekte	Anteil in Prozent	Anzahl Aspekte	Anteil in Prozent
Positiv	1657	66.10	611	71.13
Neutral	101	4.03	44	5.12
Negativ	749	29.87	204	23.75
Insgesamt	2507	100.0	859	100.0

Tabelle 8.: Trainings- und Testdaten im Vergleich

Für meinen Testdurchlauf wird der in Unterkapitel 4.3 beschriebene SimpleLogistic-Klassifikator, welcher auf den definitiven 32 Features trainiert wurde, auf das Testset angewendet. Dieses besteht aus 90 Rezensionen mit insgesamt 676 Sätzen und beinhaltet 859 „Opinion Target Expressions“ (OTE) und Entität-Aspekt-Paare. Davon sind 611 positiv, 44 neutral und 204 negativ. Im Vergleich zum Trainingsset ergibt dies die in Tabelle 8 ersichtliche Verteilung. Daran kann gesehen werden, dass im Testset das Bias zugunsten der positiven Klasse noch grösser ist. Anzumerken ist, dass in den offiziellen Testdaten in den <text>-Knoten vermutlich Konvertierungsfehler bei Sonderzeichen wie „&“ und „ ’ ”vorhanden sind. Um die Kompabilität mit den Trainingsdaten, in welchen diese Zeichen richtig kodiert sind, herzustellen, habe ich diese Fehler mithilfe eines Skripts („testdata_corrector.py“) behoben und die richtig kodierten Zeichen eingefügt. Somit werden die Sonderzeichen in den Testdaten gleich wie in den Trainingsdaten dargestellt.

System	**Genauigkeit in Prozent**
SemEval 2016 - Baseline	76.4840
WEKA - SimpleLogistic	71.1292

Tabelle 9.: Resultate auf den Testdaten

Wird der SimpleLogistic-Klassifikator auf dieses Testset angewendet, so ergibt dies eine Genauigkeit von 71.1292%. Das sind 5.38 Prozentpunkte weniger als bei der 10-fachen Kreuzvalidierung über dem Trainingsset. Darüber hinaus liegt das Resultat 5.35 Prozentpunkte unterhalb der SemEval-Baseline, was in Tabelle 9 ersichtlich ist. Wie weiter oben angemerkt, handelt es sich bei der SemEval-Baseline um ein domänenabhängiges System. Bei der Evaluation hat sich gezeigt, dass dieses domänenabhängige System auf den Testdaten bessere Resultate liefern kann als mein domänenunabhängiges.

5.5. Diskussion und Vergleich der Resultate

Wie in Unterkapitel 5.4 beschrieben, hat mein System im Vergleich mit der SemEval-Baseline auf den Testdaten schlechter abgeschnitten. Dies könnte auf die Domänenabhängigkeit der Baseline zurückgeführt werden, die, wie weiter oben beschrieben, rein auf den Trainingsdaten basiert. Somit könnte die Baseline besser auf die Daten abgestimmt gewesen sein als mein System, welches domänenunabhängig funktioniert.
Die folgenden Punkte könnten weitere Gründe für das schlechtere Resultat auf den Testdaten im Vergleich zur Kreuzvalidierung auf den Trainingsdaten liefern: Es könnte an der automatischen Featureauswahl des SimpleLogistic-Klassifikators liegen, welche anhand einer fünffachen Kreuzvalidierung auf den Trainingsdaten erfolgt und deswegen auf diesen basiert. Die Analyse der Testdaten in Kapitel 5.4 ergibt zudem eine schlechtere Ausgewogenheit der Klassen „positiv“, „neutral“ und „negativ“, was ebenfalls eine Auswirkung auf die Genauigkeit haben könnte. Wie in Kapitel 3.1 beschrieben bestehen die Trainingsdaten aus den Trainings- und Testdaten der Aufgabe von 2015. Die Testdaten hingegen wurden neu gesammelt und annotiert [Pontiki et al., 2016, 23]. Dies könnte bedeuten, dass die Testdaten sich stärker von den Trainingsdaten unterscheiden als diese untereinander und es somit, trotz gleicher Domäne, zu einem Leistungsabfall kommt, da andere Phänomene als in den Trainingsdaten auftreten könnten. Ebenfalls könnten die Features zu wenig robust sein, um gewünschte Ergebnisse auf ungesehenen Daten zu erreichen.

System	**Genauigkeit in Prozent**
XRCE [Brun et al., 2016]	88.1260
SemEval 2016 - Baseline	76.4840
AKTSKI [Pateria and Choubey, 2016]	71.7110
WEKA - SimpleLogistic	71.1292
BUAP (unconstrained)	60.8850

Tabelle 10.: Auswahl von Resultaten auf den Testdaten

Bei den 28 für die Aufgabe eingereichten Systemen am SemEval-Workshop 2016 reichten die Resultate von 88.126% bei Team „XRCE“ [Brun et al., 2016] bis 60.885% bei Team „BUAP“ (vgl. Tabelle 4 in Pontiki et al. [2016, 26]). Es konnten sowohl eingeschränkte („constrained“) als auch uneingeschränkte („unconstrained“) Systeme eingereicht werden. Eingeschränkte Systeme, davon wurden 11 eingereicht, dürfen nur die zur Verfügung gestellten Trainingsdaten verwenden, während uneingeschränkte auch auf andere externe Ressourcen zurückgreifen können [Pontiki et al., 2016, 26]. Bei den restlichen 17 handelt es sich um ebensolche uneingeschränkten

Systeme. Bei den meisten Systemen wird supervisiertes Lernen zur Klassifikation verwendet, es wurden aber beispielsweise auch unsupervisierte Ansätze (Alvarez-López et al. [2016]) oder ein Ansatz mit Diskursanalyse (Schouten and Frasincar [2016]) eingesetzt.
Das beste Resultat, eine Genauigkeit von 88.1260%, erreichte das XRCE-Team [Brun et al., 2016]. Dabei handelt es sich um ein eingeschränktes System („constrained“). Das System kann in drei Teile geteilt werden: Extrahieren linguistischer Features, Sequenzlabeling und Klassifikation. Die Autoren verwenden den syntaktischen Parser XIP [Aït-Mokhtar et al., 2002]. Dieser beinhaltet eine ganze Verarbeitungskette, welche Tokenisierung, morphosyntaktische Analyse, POS-Tagging, Eigennamenerkennung, Chunking und Dependenzrealtionsextraktion beinhaltet. Darauf aufbauend verwenden die Autoren eine Komponente, welche semantische Informationen über die Aspekttargets und ihre Polarität extrahieren. Für die Aufgabe wurden syntaktische Dependenzen, lexikalische Information über Wortpolaritäten und semantische Klassen und Subkategorisierungsinformationen mit dem Parser kombiniert. Der Parser und dessen Erweiterungen werden verwendet, um Features für die nachfolgenden maschinellen Lernverfahren zu extrahieren.
Für die Schätzung der Sentimentpolarität verwenden die Autoren die gleiche Pipeline wie für die Aspektkategorienklassifikation, jedoch wird hier die höchste Polaritätswahrscheinlichkeit dem Term oder dem Satz zugeordnet und gemischte Fälle werden ignoriert. Zudem verwenden die Autoren die zuvor gefundenen Kategorien als Features und ersetzen die „Opinion Target Expressions“ (OTE) mit der generischen Aspektkategorie.
Die meisten Fehler macht das System bei der Klassifikation der neutralen Sätze. Dies führen die Autoren auf die seltenen Beispiele der neutralen Klasse in den Trainingsdaten zurück. Zudem sind die Polaritätslexika auf die positive und negative Klasse ausgerichtet.
Das Team von AKTSKI [Pateria and Choubey, 2016] erreichte ein Resultat von 71.711%. Verwendet wurde ein SVM-Klassifikator. Das Team benutzte die Scikit-learn-Implementation [Pedregosa et al., 2011], welche auf den SemEval-Daten trainiert wird. Aufbereitet werden die Daten mithilfe von NLTK [Bird et al., 2009]. Zudem werden ein WordNet-Package[12], SentiWordNet [Baccianella et al., 2010], Bing Liu's Opinion Lexicon [Liu et al., 2005] und das MPQA Subjectivity Lexicon [Wilson et al., 2005] verwendet, sowie der Stanford Dependency Parser [De Marneffe et al., 2006]. Es handelt sich also um ein uneingeschränktes System („uncinstrained“), mit welchem nebst den Trainingsdaten auch andere externe Ressourcen verwendet werden können (vgl. Pontiki et al. [2016, 26]). Sie suchen Teilsätze, welche direkt mit

[12]Princeton University "About WordNet." WordNet. Princeton University. 2010. `http://wordnet.princeton.edu` [26.11.2016].

dem Target verbunden sind und somit dieses beeinflussen könnten. Dies wird anhand des Dependenzgraphen, welcher vom Parser ausgegeben wird, gemacht. Wenn kein explizites Target im Satz vorhanden ist, dann benutzen die Autoren eigene Target-Werte: Sie schauen nach, welche Worte, die Sentiment enthalten, mit einem Substantiv oder Pronomen zusammenhängen. Dann wird die Polarität mit einem Lexikon ermittelt und die damit zusammenhängenden Substantive als Target ausgewählt. Ebenfalls werden so „food“, „drinks“, „service“, „waiter“, „price“, „staff“, „ambiance“ als Target markiert, wie auch „they“, „she“, „he“. Stoppwörter werden anschliessend aussortiert, Nummern und Symbole (ausser „!“) ebenfalls.
Des Weiteren verwenden Pateria and Choubey [2016] zwei weitere Methoden (Methoden 1 und 2): Dabei geht es darum, Zusammenhänge zwischen einzelnen Sätzen einer Rezension auszunutzen. Hierbei spielen zwei Eigenschaften eine zentrale Rolle, welche die Autoren „flow“ und „trans“ nennen [Pateria and Choubey, 2016, 321]. Flow besteht, wenn die Sentimente in den Sätzen einer Rezension die gleiche Polarität aufweisen und Trans besteht, wenn die Sentimente innerhalb der Rezension wechseln, z.B. aufgrund von Kontrastwörtern wie „but“, „however“, „though“, etc. Für Methode 1 wird ein neues polares Lexikon D erstellt. Die Trainingsdaten werden in drei Teile entsprechend der Polaritätslabel aufgeteilt. Anschliessend werden Sentimentausdrücke gesucht anhand der Lexika, Bi- und Trigramme bestehend aus Sentimentausdrücken und Negationen, sowie Ausdrücke, welche in einer Wortliste mit neutralen Wörtern nachgeschlagen werden. Für jeden Teilsatz wird dann ein Featureset erstellt, welches allen positiven, neutralen und negativen Ausdrücken, welche im Teilsatz und in D vorkommen, sowie im Teilsatz vorkommende Kontrastwörter beinhaltet. Anschliessend werden die Features eines Teilsatzes mit diesem neuen Featureset der zwei vorhergehenden Sätze erweitert. Beim ersten und zweiten Satz einer Rezension werden diese Features leer gelassen. Damit soll der Klassifikator feststellen können, inwieweit die verschiedenen Teilsätze der Rezension in eine polare Richtung ausschlagen. Je stärker dieser Effekt ist, desto weniger ambig ist die Polarität. Methode 2 verhält sich ähnlich der Autoregression[13]. Auch hier wird das neue Lexikon D verwendet.
Das Resultat von 71.711% hat eine Differenz von ca. 10% im Vergleich zur Evaluation auf den Trainingsdaten, welche eine Genauigkeit von 82% ergeben hat [Pateria and Choubey, 2016, 322]. Gemäss den Autoren könnte dies daran liegen, dass das Lexikon D von Methoden 1 und 2 auf Basis der Trainingsdaten erstellt wurde und deshalb weniger gut auf den nicht gesehenen Testdaten funktioniert. Ebenfalls wurde Methode 2 an die Trainingsdaten angepasst, wobei auch hier eine Leistungseinbusse auf ungesehenen Daten erwartet werden kann. Die Autoren führen auch an, dass

[13] `http://paulbourke.net/miscellaneous/ar/` [26.11.2016].

das Resultat durch robustere Features oder allenfalls mit Methoden des „Ensemble Learnings“ zur Verbesserung des Klassifikationsmodells verbessert werden könnte.

6. Schlussfolgerung

In meiner Bachelorarbeit habe ich mich mit der Klassifikation von Aspekten innerhalb von Rezensionen im Rahmen der aspektbasierten Sentimentanalyse beschäftigt. Das Ziel meiner Arbeit war es, ein System zu entwickeln, welches basierend auf den am SemEval-Workshop 2016 zur Verfügung gestellten Trainingsdaten der Domäne „Restaurants“ und ferner der Domäne „Laptops“ Features extrahiert, womit ein Klassifikator in einem supervisierten Lernverfahren trainiert werden kann, um die Polarität von Aspekten bestimmen zu können. Dazu werden die Trainingsdaten durch eine Pipeline verarbeitet. Die Sätze werden tokenisiert, getaggt und mit Lemmata versehen sowie geparst. Anschliessend werden mithilfe von externen Ressourcen polare Wörter markiert. Schliesslich können Features aus den Daten extrahiert werden, mit welchen ein Klassifikator in WEKA [Hall et al., 2009] trainiert wird.
Bei der zehnfachen Kreuzvalidierung auf den Trainingsdaten erreicht mein System mit dem SimpleLogistic-Klassifikator eine Genauigkeit von 76.5058%, was über einer Mehrheitsklassen-Baseline liegt. Auf den Testdaten erreicht mein System eine Genauigkeit von 71.2456%. Meine Arbeit zeigt, dass mit relativ einfachen Features akzeptable Resultate erreicht werden können. Das System könnte erweitert werden durch weitere wie auch robustere Features. Auch vorstellbar wären komplexere Verfahren zur detaillierteren Analyse der Polarität von Aspekten.
Der Vergleich mit einem domänenabhängigen Klassifikationssystem hat ergeben, dass meine Klassifikation domänenunabhängig funktioniert. Dies hat den Vorteil, dass das System auch auf andere Domänen angewendet werden kann. Andererseits sind domänenabhängige Systeme speziell auf eine bestimmte Domäne abgestimmt, um ihre Leistung auf dieser Domäne zu steigern. Die SemEval-Baseline, welche aufgrund ihrer Funktionsweise domänenabhängig ist, konnte auf der Domäne „Restaurants“ eine bessere Genauigkeit erreichen.
Die Lexika, welche von meinem System verwendet werden, könnten um domänenabhängige polare Wörter ergänzt werden. Damit könnte eine Leistungssteigerung auf der Domäne „Restaurants“ erreicht werden. Andererseits würde das System so domänenabhängig und entsprechend auf anderen Domänen eine schlechtere Performanz zeigen.

Bibliografie

Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, 6(1):37–66.

Aït-Mokhtar, S., Chanod, J.-P., and Roux, C. (2002). Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(2-3):121–144.

Alvarez-López, T., Juncal-Martınez, J., Fernández-Gavilanes, M., Costa-Montenegro, E., and González-Castano, F. J. (2016). Gti at semeval-2016 task 5: Svm and crf for aspect detection and unsupervised aspect-based sentiment analysis. *Proceedings of SemEval*, pages 306–311.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G. A., and Reynar, J. (2008). Building a sentiment summarizer for local service reviews. In *WWW workshop on NLP in the information explosion era*, volume 14, pages 339–348.

Boiy, E. and Moens, M.-F. (2009). A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brun, C., Perez, J., and Roux, C. (2016). Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 277–281. Association for Computational Linguistics.

De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240. ACM.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA. ACM.

Jakob, N. and Gurevych, I. (2010). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1035–1045. Association for Computational Linguistics.

Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

Jin, W., Ho, H. H., and Srihari, R. K. (2009). A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 465–472. Citeseer.

Landwehr, N., Hall, M., and Frank, E. (2005). Logistic model trees. *Machine Learning*, 59(1-2):161–205.

Le Cessie, S. and Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied statistics*, pages 191–201.

Liu, B. (2015). *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Liu, B., Hu, M., and Cheng, J. (2005). Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA. ACM.

Lu, B., Ott, M., Cardie, C., and Tsou, B. K. (2011). Multi-aspect sentiment analysis with topic models. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 81–88. IEEE.

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Moghaddam, S. and Ester, M. (2010). Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1825–1828. ACM.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2Nd International Conference on Knowledge Capture*, K-CAP '03, pages 70–77, New York, NY, USA. ACM.

Nivre, J. and Hall, J. (2005). Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*, pages 13–95.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Pateria, S. and Choubey, P. (2016). Aktski at semeval-2016 task 5: Aspect based sentiment analysis for consumer reviews. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 318–324. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Marıa Jiménez-Zafra, S., and Eryigit, G. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. *Proceedings of SemEval*.

Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., and Androutsopoulos, I. (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pages 486–495.

Quinlan, J. R. (1993). C4.5: Programming for machine learning. *Morgan Kauffmann*, page 38.

Schouten, K. and Frasincar, F. (2016). Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Titov, I. and McDonald, R. (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 111–120, New York, NY, USA. ACM.

Tsytsarau, M. and Palpanas, T. (2012). Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yu, J., Zha, Z.-J., Wang, M., and Chua, T.-S. (2011). Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1496–1505. Association for Computational Linguistics.

Zhu, J., Wang, H., Tsou, B. K., and Zhu, M. (2009). Multi-aspect opinion polling from textual reviews. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1799–1802. ACM.

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

A. Liste von Skripten

Pipeline zur Verarbeitung der Rezensionen

Für Details siehe Kapitel 4.

- `./pipe/`
  - *p00_pipeline.py*
  - *p10_sentsextractor.py*
  - *p11_nltktagger.py*
  - *p12_maltparser.py*
  - *p20_parseimporter.py*
  - *p21_CoNLLToXML.py*
  - *p30_polaritylookup.py*
  - *p40_featureextractor.py*
  - *p41_featureutilities.py*
  - *p42_featurefunctions.py*

Skript zur Verarbeitung der Rezensionen für MALLET

Für Details siehe Kapitel 5.3.

- `./pipe/mallet_extraction/`
  - *mallet_extractor.py*

Korrektur der Formatierung von speziellen Zeichen in den Testdaten

Für Details siehe Kapitel 5.4.

- `./pipe/testdata_correction/`
  - *testdata_corrector.py*