

# Aspekt-basierte Sentiment Analysis

Fritjof Bornebusch, Glaucia Cancino, Alvine Nzeungang Fansou,  
Maryam Farajzadeh Jalali, Jamal Mohsen, Max Nitze, Christina Plump,  
Ronald Smith Djomkam Yotedje, Hubert Fred Tchambo, Toni, Henning Ziegler

Arbeitsgruppe Rechnerarchitektur  
Universität Bremen

{fritjof,cancino,alvine,maryam,mohsen,maxnitze,cplump,smith,  
tfred,toni,hziegler}@informatik.uni-bremen.de

**Abstract:** Sentiment Analysis (engl. für Stimmungsanalyse) wird verwendet, um Meinungen, Gefühle und Emotionen aus Texten zu extrahieren. In diesem Artikel wird ein Ansatz zur Sentiment Analysis auf Basis von Aspekten beschrieben. Dieser Ansatz kann unter anderem für die automatisierte Klassifizierung von Produktkommentaren, wie sie beispielsweise auf Bewertungsplattformen vorkommen, verwendet werden. Dieser Ansatz gliedert sich in vier wesentliche Teilaufgaben: (i) Extraktion der Aspekte, (ii) Bewertung der Polarität der Aspekte, (iii) Klassifizierung der Aspekte in vorgegebene Kategorien und (iv) Bewertung der Polarität der Kategorie. Zur Umsetzung des Ansatzes werden Methoden der maschinellen Sprachverarbeitung eingesetzt.

## 1 Einleitung

*Sentiment Analysis* bezeichnet die Untersuchung von Meinungen, Gefühlen und Emotionen, die in einem Satz oder einem Text ausgedrückt werden ([Liu10]). Das grundlegende Problem bei der Sentiment Analysis ist es zu erkennen, welche Polarität ein gegebener Satz hat ([KK07]), wobei die Polarität angibt, ob der Satz sinngemäß positiv oder negativ ist. Dies wird zum Beispiel von Firmen genutzt, um einzuschätzen, wie ihre Produkte von Kunden angenommen werden. Mit den entsprechenden Rückmeldungen entwickeln sie Strategien, um die Qualität ihrer Produkte zu verbessern ([Vec10]).

Bisherige Arbeiten in diesem Bereich haben sich unabhängig von vorhandenen Entitäten (engl.: *named entity*) ([TP12]) oder Aspekten ([ME10]) immer mit der Analyse ganzer Sätze oder auch Absätze beschäftigt.

Die *SemEval-2014* ist ein Workshop der sich mit der Verarbeitung natürlicher Sprache (engl. *Natural Language Processing (NLP)*) beschäftigt. Ein *Task* der im Zuge dieses Workshops ausgeschrieben wurde ist die *Aspect Based Sentiment Analysis (ABSA)*<sup>1</sup>. Das Ziel in diesem Task ist es, eine Sentiment Analysis durchzuführen, die darauf basiert, dass zunächst die Polaritäten von einzelnen Aspekten eines Satzes oder Absatzes bestimmt werden. Diese werden anschließend auf vorgegebene Kategorien erweitert. Dafür wurde das

---

<sup>1</sup><http://alt.qcri.org/semeval2014/task4/>

Problem in vier verschiedene Teilaufgaben unterteilt. Die erste Teilaufgabe extrahiert aus einem Satz alle vorhandenen Aspekte, die zweite bestimmt die Polaritäten der jeweiligen Aspekte, die dritte die Kategorien der einzelnen Sätze und die letzte die Polaritäten zu den ermittelten Kategorien. Um das System evaluieren zu können, wurden Trainingsdaten während der Entwicklung zur Verfügung gestellt.

## 2 Verwandte Arbeiten

Die in Python implementierte Bibliothek Natural Language Toolkit (NLTK)<sup>2</sup> wird von Laurent Luce ([Luc12]) eingesetzt, um Twittermeldungen als positiv oder negativ zu klassifizieren. Dies wird über den NLTK eigenen Klassifizierer realisiert, da dieser durch eigene Sätze, die entweder positiv oder negativ sind, trainiert werden kann.

Die Arbeit von Pang et al. ([PLV02]) befasst sich mit der Analyse von Filmbewertungen, um zu ermitteln, ob ein Film insgesamt als negativ oder positiv bewertet wurde. Hierfür werden Bayessche Filter, die Maximum Entropie Methode und Support Vector Machines verwendet. Es wird in dieser Arbeit mit den verschiedenen Klassifizierern gearbeitet.

Eine weitere Arbeit zu diesem Thema kommt von Kim et al. ([KH06]) und befasst sich mit der automatisierten Analyse von Polaritäten aus Online Bewertungen. Es wird der Ansatz gewählt, die Gründe für eine Meinung auszuwerten und dahingehend zu analysieren, ob diese eher eine Meinung oder Tatsache beschreiben. Auch hier wird die Maximum Entropy Methode verwendet.

Die in [PLV02] und [KH06] beschriebenen Methoden verwenden Klassifizierer, die mit Daten trainiert werden und anschließend zur Analyse der Testdaten eingesetzt werden. In unserer Arbeit werden keine Klassifizierer verwendet, allerdings könnten diese zu einer weiteren Verbesserung der Ergebnisse beitragen.

Bagheri et al. ([BSdJ13]) stellen in Ihrer Arbeit eine Möglichkeit vor, wie Aspekt Terme in Bewertungen gefunden werden können. Diese Arbeit basiert auf einer eigenen Entwicklung bestehend aus dem Lernen von Aspekt-Termen und dem Anwenden von heuristischen Regeln.

Unser Ansatz basiert auf der Verwendung von Bibliotheken. Die Verwendung von selbst-trainierte Klassifizierern wäre ebenfalls eine Möglichkeit, mit der die Ergebnisse verbessert werden könnten.

## 3 Grundlagen

Die zu bearbeitende Aufgabe basiert hauptsächlich auf NLP-Techniken. Damit ein Computer natürliche Sprache bearbeiten bzw. verstehen kann, sind mehrere Schritte notwendig. Zunächst einmal muss die grundlegende Struktur des vorliegenden Satzes verstanden

---

<sup>2</sup><http://www.nltk.org/>

Tabelle 1: Annotationen für den Satz „My dog likes eating sausages.“

Token	my	dog	likes	eating	sausages
Lemma	my	dog	like	eat	sausage
POS	PRP	NN	VBZ	VBG	NNS

werden (Parsing). Dies extrahiert jedoch nur das Gerüst des Satzes und nicht die tatsächliche Bedeutung desselben. Hierfür ist ein Verständnis der Semantik des Satzes (und damit natürlich zunächst einmal der einzelnen Wörter) von Nöten. Dieses Problem wird durch Bibliotheken (WordNet [Mil95], RiTa.WordNet<sup>3</sup>) gelöst, in denen die Bedeutungen aller Wörter und ihrer Synonyme aufgeteilt nach verschiedenen Wortgruppen enthalten sind. Mithilfe dieser Bibliotheken oder Wörterbücher können auch Beziehungen zwischen Wörtern erkannt werden. Hierauf geht der zweite Unterabschnitt dieses Kapitels ein. In einigen Teilgebieten des NLP, vor allem der Sentiment Analysis, ist weiterhin der Grundtenor des Satzes von Interesse. Hierbei steht die Frage im Vordergrund, ob der Satz eine positive oder negative Meinung darstellt.

Der folgende Abschnitt soll einige Begrifflichkeiten im Zusammenhang mit Natural Language Processing erläutern: Ein einfaches Wort, so wie es im Satz steht (das heißt, in seiner gebeugten Form) wird als *Token* bezeichnet. Hierunter fallen auch Satzzeichen und Zahlen. Wird ein Wort auf seinen Wortstamm zurückgeführt, so wird dieser als *Lemma* bezeichnet. Mit einem *Part-of-Speech Tag* wird angegeben, welche grammatikalische Eigenschaft das Wort einnimmt.

Parser geben die syntaktische bzw. grammatikalische Struktur von Sätzen unter anderem durch *Typed Dependencies* ([dMM08]) an. Diese geben Beziehungen zwischen Token des Satzes an. Der *dependent* ist dabei abhängig vom *governour* und die Relation zwischen diesen beiden Wörtern wird durch *relation* beschrieben.

**Beispiel 1** *My dog likes eating sausages.* Tabelle 1 zeigt die einzelnen Tokens, Lemmata und POS-Tags des obigen Satzes. Tabelle 2 enthält die entsprechenden *Typed Dependencies*, sowie ihre Auflösung nach *governour*, *dependent* und *relation*.

Der am meisten verbreitete Parser ist der Stanford Parser ([SBMN13], [KM03]). Er wird von der Natural Language Processing Group an der Stanford University entwickelt und basiert auf den Elementen der Penn Treebank.

Im nächsten Schritt muss ein Verständnis für die semantische und lexikalische Bedeutung des Satzes entwickelt werden. Dies geschieht mithilfe der Bibliotheken WordNet [Mil95] und RiTa.WordNet. Die Bibliothek WordNet ermöglicht es, alle Bedeutungen eines Wortes zu überblicken, sortiert nach Worttyp – *noun*, *verb*, *adjective*. Weiterhin sind für jeden Wortsinn alle Synonyme und Beschreibung aufgelistet, sowie Über- und Unterbegriffe. RiTa.WordNet bietet eine Schnittstelle zu WordNet, mithilfe derer auch Informationen aus WordNet erhalten werden können, die nicht direkt abzufragen sind. Eine sehr wichtige dieser Funktionen ist der *Path based similarity algorithm* ([ZY08]). Dieser Algorithmus durchläuft Oberbegriffe der zu vergleichenden Wörter, bis ein gemeinsamer gefunden

<sup>3</sup><http://www.rednoise.org/rita/>

Tabelle 2: Typed Dependencies für den Satz „My dog likes eating sausages.“

dependency	governour	dependent	relation
poss(dog, My)	dog	my	poss
nsubj(likes, dog)	likes	dog	nsubj
root(ROOT, likes)	ROOT	likes	ROOT
xcomp(likes, eating)	likes	eating	xcomp
dobj(eating, sausages)	eating	sausages	dobj

wurde. Aus der Anzahl der nötigen Schritte, um von Wort zu Wort zu kommen, wird die Ähnlichkeit der beiden Wörter berechnet – je kürzer, desto ähnlicher.

Die Sentiment Analysis soll Wörtern, Sätzen und/oder Abschnitten eine Polarität zuordnen. Mit SentiWordNet [GGT13] gelingt dies auf Wortebene. Es wird der Thesaurus aus WordNet benutzt. Jedem Wort werden Gewichtungen für die drei Polaritäten *positive*, *negative* oder *objective* zugeordnet, wobei sich die drei Werte zu 1 addieren müssen.

Eine mögliche Implementierung, mit deren Hilfe der Grundtenor des gesamten Satz oder einzelner Teilsätze ermittelt werden kann, ist der *Sentiment Tree*. Diese Struktur wurde für die Analyse von Filmbewertungen an der Stanford Universität entwickelt und in die Stanford CoreNLP Bibliothek integriert (vgl. [SPW<sup>+</sup>13]). Der von dieser Bibliothek zur Verfügung gestellte Sentiment Tree basiert auf der *Sentiment Treebank*, die aus rund zwölftausend annotierten *Parse-Trees* besteht. Das dem Sentiment Tree zugrunde liegende *Recursive Neural Tensor Network*, welches über die Sentiment Treebank trainiert wurde, kann fünf Polaritätsklassen erkennen: *sehr negativ*, *negativ*, *neutral*, *positiv* und *sehr positiv*. Für die Erkennung der Aspekt-Term-Polaritäten fassen wir diese zu den drei wesentlichen Klassen *negativ*, *neutral* und *positiv* zusammen.

## 4 Implementierung

Zu Beginn werden die eingegebenen Sätze vorbereitet um anschließend auf ihnen eine Sentiment Analysis durchgeführt. Dazu werden unter anderem Annotierungen vorgenommen, die jeder weitere Algorithmus benötigt. Diese werden mit dem Satz zusammen für die weitere Verarbeitung gespeichert.

Die Implementierung gliedert sich, in Anlehnung an die vier Teilaufgaben der *SemEval-2014*, in vier Teilbereiche: Die *Aspect Term Extraction* (Abschnitt 4.1) und die *Aspect Term Polarity* (Abschnitt 4.2), die sich mit der Sentiment Analysis auf Aspekt-Ebene beschäftigen, sowie der *Aspect Category Extraction* (Abschnitt 4.3) und der *Aspect Category Polarity* (Abschnitt 4.4), die dies auf Kategorie-Ebene durchführen.

Zum Testen der Verfahren wurden die im Rahmen der *SemEval-2014* zur Verfügung gestellten Trainingsdaten verwendet. Die Daten enthalten jeweils etwa 3000 englischsprachige Sätze aus Laptop- und Restaurantbewertungen, bei denen per Hand annotiert wurden.

Das Messkriterium, um die Ergebnisse mit den Trainingsdaten zu vergleichen, ist die *F-measure*. Diese setzt sich als harmonisches Mittel aus der *Precision*, also der Anzahl der richtigen an den gesamten Elementen, und dem *Recall*, dem Verhältnis von richtigen zu falschen Elementen, zusammen.

#### 4.1 Aspect Term Extraction

Im ersten Teil der Aufgabe 4 der *SemEval-2014* sollen in Reviews zu einem bekannten Thema (in diesem Fall *restaurant* und *laptop*) genannte Aspekte erkannt werden. Hierfür wurden zwei Ansätze verwendet: *NounFrequency* und *TermAggregation*.

**NounFrequency.** Nach [HL04] ist eine gute Grundlage, um Aspekte zu erhalten, eine Liste von Nomen zu verwenden. Da dies zu einer sehr ungenauen *precision* führen würde, ist eine Untersuchung auf Häufigkeit von Nöten. Dafür werden in der Vorverarbeitung alle Nomen nach Häufigkeit sortiert in eine Liste geschrieben. Die ersten 20% werden als mögliche Aspekte betrachtet. Die 20% sind ein willkürlich gewählter Wert, der sich auf den Trainingsdaten als am besten erwiesen hat.

Dieser Ansatz führt zu zwei Problematiken: Zum einen beschreibt der Stanford Parser Wörter, die komplett in Großbuchstaben geschrieben sind, direkt als *proper nouns* (NNP(s)), womit die Häufigkeitsliste verfälscht wird (vgl. Beispiel 2). Zum anderen führen Rechtschreibfehler innerhalb der Reviews zu einer verfälschten Liste und zu falschen Häufigkeiten (vgl. Beispiel 3).

**Beispiel 2** Der Satz sei: „*DO NOT BUY IT.*“ Dann werden alle vier Wörter als NNP beschrieben, dementsprechend erreichen vier Wörter, die keine Nomen sind, sowohl die Liste als auch die Liste aller Nomen eines Satzes.

**Beispiel 3** Der Satz sei: „*Performace was great on this one.*“ *Performace* soll hier offensichtlich *performance* heißen. Letzteres hat eine relativ hohe Häufigkeit, der inkorrekte Term jedoch nicht, er wird also fälschlicherweise nicht ausgegeben.

Ein Lösungsansatz für diese Probleme ist es Wörter, die ausschließlich aus Großbuchstaben bestehen vor dem Parsieren in Kleinbuchstaben abzuändern. Worte, die dann noch nicht in der Häufigkeitsliste auftauchen könnten dann einer Rechtschreibprüfung unterzogen werden, wenn sie zum Beispiel in WordNet nicht aufzufinden sind.

Eine weitere Problematik, die durch den Ansatz der häufigsten Wörter entsteht, ist, dass Wörter, die sehr häufig vorkommen (die oberen 2%), häufig nicht Aspekte sind, sondern das Oberthema bzw. Synonyme desselben oder semantisch ähnliche Wörter. Dies gilt auch für andere Wortgruppen, welche in Tabelle 3 mit Beispielen zusammengefasst sind. Diese werden aus der Häufigkeitsliste ausgeschlossen.

**Beispiel 4** Sei das Thema *Laptop* und der Satz „*Hard drives and battery life are alright, but the touchpad doesn't react as well as hoped and the display becomes unreadable as*

Tabelle 3: Gefilterte Wortgruppen mit Beispielwörtern

	Beispiel
person	husband, wife, mother, boyfriend
time	date, year, month, monday-sunday
location	NYC, Manhattan, street, Avenue
mistagged	everything, something, none, some, any

*soon as its not raining outside.*“ Aspekte wären in diesem Fall: Hard drives, battery life, touchpad und display.

Der beschriebene grundlegende Ansatz mit seinen kleinen bis großen Modifikationen liefert bereits eine gute Basis des Ganzen. Jedoch – wie auch in Beispiel 4 zu sehen ist – gibt es auch Aspekte, die aus mehreren Wörtern bestehen (z.B. hard drives). Hierfür werden hintereinander vorkommende Aspekte zusammengefasst. Dieser kleine Schritt bringt bereits eine große Verbesserung.

Tabelle 4 fasst die Ergebnisse für die trainingsdaten zusammen. Diese liegen mit 30% bzw. 40% F-measure weit unter den möglichen Werten (vgl. [ZY08]). Ein Verbesserungspunkt ist, mehr mit den Typed Dependencies zu arbeiten (vgl. Algorithmus *AggregatedWords*).

Tabelle 4: Ergebnisse des Noun Frequency Algorithmus

Datensatz (Domain)	Precision	Recall	F-measure
Laptop	0,25	0,38	0,30
Restaurant	0,34	0,45	0,39

**Aggregated Words.** Der Algorithmus erkennt zunächst Aspekte, die mehrere Wörter enthalten. Danach werden erst die Einzelaspekte identifiziert. Eine tragende Rolle spielen hierbei die Typed Dependencies, wobei nur die Relationen  $nn^4$  und  $amod^5$  verwendet wurden.

**Beispiel 5** Die relevanten Typed Dependencies des Satzes „It has good speed and plenty of hard drive space“ sind:  $amod(speed, good)$ ,  $amod(space, hard)$  und  $nn(space, drive)$ . Als Aspekte erkannt würden nun: hard drive space und good speed.

Der erste Schritt besteht daraus, Wortgruppen zum gleichen *governour* zu finden. Sind diese alle gefunden, werden alle übrigen Nomen als Aspekte identifiziert. Hier entsteht nun eine ähnliche Problematik wie im Algorithmus *Noun Frequency*: Es werden zu viele Nomen erkannt. Dies wird dadurch behoben, dass Listen von *unwichtigen* und *wichtigen* Nomen und Adjektiven angelegt werden und mithilfe dieser Listen die Ergebnisse weiter gefiltert werden. Weiterhin werden alle Wörter, die zu ausgewählten Wortkategorien aus WordNet gehören, gefiltert. Dies führt zu einer besseren *precision*.

<sup>4</sup>noun compound modifier: Beziehung zwischen zwei Nomen

<sup>5</sup>adjectival modifier: Beziehung zwischen Adjektiv und Nomen

## 4.2 Aspect Term Polarity

Die *Aspect Term Polarity* bestimmt für gegebene Aspekte eines Satzes die dazugehörigen Polaritäten.

Diese Ideen wurden mit Hilfe des *Sentiment Trees* verfeinert. Der Sentiment Tree ist für die Bestimmung der Polarität eines kompletten Satzes ausgelegt und muss für die Bestimmung der Aspektpolaritäten entsprechend ausgewertet werden. Erläutert wird die Funktionsweise des Algorithmus anhand des Beispiel-Satzes „The keyboard is too slik.“ mit dem Aspekt *keyboard*.

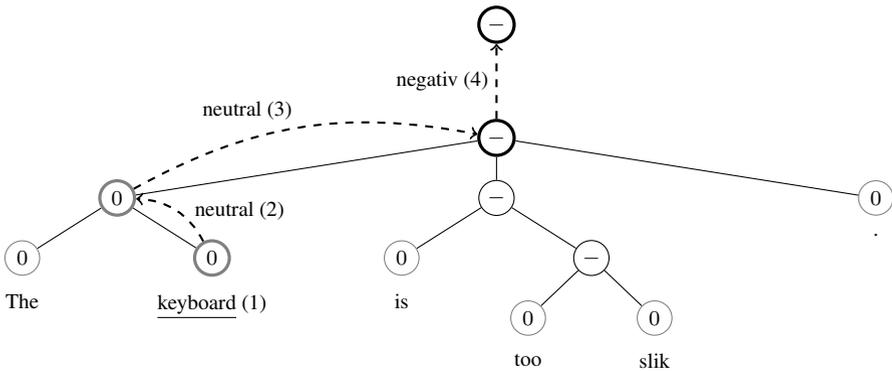


Abbildung 1: Beispiel des Sentiment-Tree-Algorithmus für den Satz „The keyboard is too slik.“

Zunächst wird der Sentiment Tree für den Satz erzeugt (siehe Abbildung 1). Daraufhin wird der Knoten, der das Hauptwort des Aspekts enthält, lokalisiert. Ausgehend von diesem Knoten wird der Baum bis zum Wurzelknoten durchsucht. Wird ein Knoten mit einer nicht-neutralen Polarität (also *negative* oder *positive*) gefunden, wird diese als Polarität für den entsprechenden Aspekt angenommen. Nur wenn der Wurzelknoten erreicht wird, ohne dass eine nicht-neutrale Polarität erkannt wurde, wird der Aspekt mit *neutral* bewertet.

Im Beispiel wird ausgehend vom *Keyboard-Knoten* (1) nach einer Polarität gesucht (2). Da dessen Elternknoten keine nicht-neutrale Polarität enthalten, wird der Wurzelknoten untersucht (3). Dieser besitzt eine negative Polarität (4), wodurch der Aspekt *keyboard* insgesamt eine negative Polarität bekommt.

In Tabelle 5 sind die Ergebnisse des Algorithmus, die mit den Trainingsdaten erreicht wurden, nach Polarität getrennt aufgeschlüsselt.

Anhand der Ergebnisse ist erkennbar, dass die F-Measure für positive Polaritäten leicht besser ist als für negative Polaritäten, wobei der Unterschied bei den Restaurant-Daten größer ausfällt. Allgemein sind die Ergebnisse aber relativ gut. Besonders problematisch gestaltet sich zur Zeit die Erkennung von neutralen Aspekten, da der Sentiment Tree fast immer eine Polarität ungleich *neutral* liefert. Um trotzdem eine bessere Erkennung von neutralen Polaritäten zu gewährleisten, wurden verschiedene Modifikationen getestet. Zum Beispiel sollten nur bestimmte Knotentypen zur Bestimmung der Polaritäten heran-

Tabelle 5: Ergebnisse des Sentiment Tree Algorithmus

Datensatz (Domain)	Precision	Recall	F-measure
Laptop			
- positive	0,66	0,67	0,67
- negative	0,51	0,73	0,60
- neutral	0,33	0,08	0,12
Restaurant			
- positive	0,78	0,70	0,74
- negative	0,38	0,74	0,50
- neutral	0,40	0,11	0,17

gezogen werden. Ebenso sollte der Algorithmus nur eine festgelegte Anzahl an Elternknoten betrachten, um sicherzustellen, dass die Polarität mit dem Aspekt zusammenhängt und nicht durch einen unabhängigen Satzteil hervorgerufen wird. Dies brachte jedoch nicht die gewünschte Verbesserung.

### 4.3 Aspect Category Extraction

Ab Aufgabenteil 3 wird nur noch mit einem Thema gearbeitet – *restaurant*. Hierfür sind folgende Kategorien vorgegeben, in die die Aspekte sortiert werden sollen: *food*, *service*, *price*, *ambience* und die neutrale Kategorie *anecdotes/miscellaneous*. Unser Ansatz lässt sich in zwei Fälle unterteilen, je nach Vorkommnis der Aspekte.

#### 1. Satz enthält Aspektterme:

Mit Hilfe des Beispielsatzes „Even though its good seafood, the prices are too high.“ soll gezeigt werden, wie der Algorithmus funktioniert. Die Aspekte des Satzes sind *seafood* und *price*.

- Entspricht ein Aspekt dem Namen einer Kategorie, wird diese direkt übernommen. Im Beispielsatz wäre dies *price*.
- Als Sonderfall wird überprüft, ob der Aspekt ein Gericht bezeichnet. Dies führt zur Zuordnung der Kategorie *food*.
- Nicht zugeordnete Aspekte werden auf ihre Ähnlichkeit bezüglich der Kategorien überprüft. Ist die *path length* (Path based similarity algorithm) geringer als 0,4 (hat sich auf den Trainingsdaten als bester Wert erwiesen), wird der Aspekt der Kategorie zugeordnet. In diesem Beispiel wäre die Kategorie *food*.

#### 2. Satz enthält keine Aspektterme:

Schritt 3 des ersten Falles wird für alle Nomen des Satzes (und nicht für alle Aspekte) durchgeführt. Der Schwellwert wird hier jedoch auf 0,19 gesenkt, um die Anzahl der erkannten Kategorien zu begrenzen. Falls kein Ähnlichkeitswert den Schwellwert unterschreitet, wird die Kategorie *anecdotes/miscellaneous* gewählt.

Tabelle 6 stellt unsere Ergebnisse dar. Das Ergebnis ist für Basisfälle gut. Sobald ein Satz Mischungen von Kategorien enthält oder weder Aspekte noch Nomen besitzt, besteht an den Ergebnissen jedoch Verbesserungsbedarf.

Tabelle 6: Ergebnisse des Kategorieerkennungs-Algorithmus

Datensatz (Domain)	Precision	Recall	F-measure
Restaurant	0,56	0,44	0,49

#### 4.4 Aspect Category Polarity

Dieser Abschnitt beschreibt die letzte Teilaufgabe, die die Polarität einer Aspekt-Kategorie in einem Satz bestimmt. Den vorgegebenen Kategorien (vgl. Abschnitt 4.3) werden die Wert *positive*, *negative*, *neutral* oder *conflict* zugeordnet. Die Polarität einer Kategorie wird über die Polaritäten der einzelnen Aspekte dieser Kategorie bestimmt. Da in diesem Aufgabenteil nur die Polaritäten von Kategorien bestimmt werden, müssen die Ergebnisse der zweiten und dritten Teilaufgabe vorliegen. Auf Basis der Trainingsdaten wurden Regeln definiert, die dazu dienen, die Polaritäten für die Kategorien zu bestimmen. Im Folgenden werden die einzelnen Regeln kurz dargestellt.

**Regel 1.** Falls die Polaritäten der Aspekte innerhalb einer Kategorie gleich sind, dann ist dies auch die Polarität der Kategorie.

**Beispiel 6** *„Prices are higher to dine in and their chicken tikka marsala is quite good.“* Bei diesem Beispiel wurden die folgenden Aspekte „Prices“ als negative und „chicken tikka marsala“ als positive gefunden. Die Aspekte wurden zwei verschiedenen Kategorien zugeordnet und deswegen sind die Bewertungen unabhängig voneinander, d.h. die Kategorie food wurde als positive gekennzeichnet und die Kategorie price wurde als negative markiert.

**Regel 2.** Ist einer der Aspekte einer Kategorie neutral, dann hat dies keinen Einfluss auf die Polarität der Kategorie, solange mindestens eine andere Polarität vorhanden ist. Die Polaritäten der anderen Aspekte bestimmen die Polarität einer Kategorie.

**Beispiel 7** *„Our agreed favorite is the orrechiete with sausage and chicken (usually the waiters are kind enough to split the dish in half so you get to sample both meats).“* Dieser Satz enthält die Aspekte „orrechiete with sausage and chicken“ und „waiters“ mit der Polarität positive sowie „meats“ und „dish“ mit der Polarität neutral. Nur der Aspekt „waiters“ gehört zur Kategorie service, dadurch wird die Polarität von service positive. Die anderen drei Aspekte haben die Polaritäten neutral und positive, dadurch wird die Kategorie food als positive deklariert.

**Regel 3.** Falls die Polaritäten der jeweiligen Aspekte einer Kategorie sowohl *positive* als auch *negative* sind, dann wird die Polarität der Kategorie als *conflict* markiert.

**Beispiel 8** Zur Illustration nehmen wir den Satz: „The sweet lassi was excellent as was the lamb chettinad and the garlic naan but the rasamalai was forgettable.“ Hier gibt es vier Aspekte: „sweet lassi“ mit der Polarität positive, „lamb chettinad“ mit der Polarität positive, „garlic naan“ mit der Polarität positive und „rasamalai“ mit der Polarität negative. Dies ergibt die Kategorie food mit der Polarität conflict.

**Regel 4.** Falls die Kategorie als *anecdotes/miscellaneous* identifiziert wurde und es keinen Aspekt gibt, dann wird der Stanford Sentiment Tree genutzt, um die Polarität des ganzen Satzes zu bestimmen. Dies ergibt dann die Polarität für die Kategorie *anecdotes/miscellaneous*.

**Beispiel 9** Der Satz „A guaranteed delight!“ hat keine Aspekte. Da dieser Satz keine Aspekte hat, wird der Stanford Sentiment Tree verwendet, um eine Polariät zu bestimmen. In diesem Fall ist diese positive.

Wir haben unseren Ansatz auf den Trainingsdaten angewendet. Das Ergebnis ist in Tabelle 7 dargestellt. Unser Algorithmus erzielte eine F-measure von 0,85 für die Polarität *positive*, was angesichts der Neuartigkeit unseres Ansatzes ein vielversprechendes Ergebnis ist. Bei der Polarität *conflict* haben wir eine *f-Measure* von 0,16.

Tabelle 7: Ergebnisse des Aspect Category Polarity Algorithmus

Datensatz (Domain)	Precision	Recall	F-measure
Restaurant			
- positive	0,89	0,80	0,85
- negative	0,55	0,76	0,64
- neutral	0,43	0,41	0,42
- conflict	0,19	0,14	0,16

## 5 Evaluation

Mit der Einteilung in vier Teilbereiche durch die *SemEval-2014* wird etwas von der konventionellen Sentiment Analysis abgewichen. Für das Auffinden der Polaritäten konnten wir zum Beispiel davon ausgehen, dass die annotierten Aspekt-Terme und Aspekt-Kategorien richtig sind, um die einzelnen Algorithmen unabhängig testen zu können und nicht Fehler aus den Extraktions-Algorithmen weiter zu propagieren. Bei einer vollständigen Analyse, also einer, die Fehler weiter propagiert, wäre dies aber nicht zwingend der Fall, sodass zum Beispiel zu falschen Aspekten im folgenden Schritt eine Polarität ermittelt wird.

Deshalb haben wir auch einmal eine solche Analyse auf den Test-Daten der *SemEval-2014* durchgeführt und diese dann mit den Ergebnissen verglichen, die jeweils mit den sogenannten *Gold-Daten*<sup>6</sup> ermittelt wurden. Das Ergebnis für diese Analyse, dargestellt in

<sup>6</sup> „richtig“ annotierte Daten

Tabelle 8: Ergebnisse einer vollständigen Sentiment Analysis der Aspekte

Datensatz (Domain)	Precision	Recall	F-measure
Restaurant			
- positive	0,83	0,31	0,45
- negative	0,46	0,37	0,41
- neutral	0,31	0,03	0,05
- conflict	-	-	-

Tabelle 8 für die Aspekte und Tabelle 9 für die Kategorien, verschlechtert sich wie erwartet deutlich. Auffällig ist jedoch, dass vor allem bei den Kategorien eine sehr hohe *precision* vorliegt. Mit dem vergleichsweise hohen *recall* in den positiven Kategorien fällt hier die F-measure im Verhältnis deutlich geringer ab als erwartet.

Tabelle 9: Ergebnisse einer vollständigen Sentiment Analysis der Kategorien

Datensatz (Domain)	Precision	Recall	F-measure
Restaurant			
- positive	0,73	0,57	0,64
- negative	-	-	-
- neutral	0,08	0,01	0,02
- conflict	-	-	-

## 6 Zusammenfassung und zukünftige Arbeit

In dieser Arbeit wurde ein konkreter Ansatz zur Sentiment Analysis auf Basis von Aspekten entwickelt. Es wird die Polarität zu den einzelnen Aspekten und Kategorien unter Nutzung der Bibliotheken Stanford CoreNLP, RiTa, WordNet, SentiWordNet und einer eigens entwickelten Datenbank, welche die Namen zu verschiedenen Gerichten enthält, bezogen auf einzelne Sätze ermittelt. Diese Bibliotheken bringen bereits eigene Methoden mit, um Sätze oder Worte zu klassifizieren, oder um die entsprechenden Polaritäten zu ermitteln.

Durch die Verwendung bereits vorgegebener Methoden konnte allerdings wenig Einfluss auf das Ergebnis genommen werden. An dieser Stelle könnten Bibliotheken wie NLTK helfen, das Ergebnis nochmals zu verbessern. Sie bieten die Möglichkeit verschiedene Klassifizierer zu trainieren und damit mehr Einfluss zu nehmen.

Generell kann man in Zukunft davon ausgehen, dass die Bewertungsmöglichkeiten in Richtung verbesserte Erkennung von einzelnen Aspekttermen und von unklaren Sätzen fortschreitet.

## Literatur

- [BSdJ13] Ayoub Bagheri, Mohamad Saraee und Franciska de Jong. An Unsupervised Aspect Detection Model for Sentiment Analysis of Reviews. In *NLDB*, Seiten 140–151, 2013.
- [dMM08] Marie-Catherine de Marneffe und Christopher D. Manning. Stanford Typed Dependency Manual, 2008. [http://nlp.stanford.edu/software/dependencies\\_manual.pdf](http://nlp.stanford.edu/software/dependencies_manual.pdf).
- [GGT13] Marco Guerini, Lorenzo Gatti und Marco Turchi. Sentiment Analysis: How to Derive Prior Polarities from SentiWordNet. In *EMNLP*, Seiten 1259–1269, 2013.
- [HL04] Mingqing Hu und Bing Liu. Mining and summarizing customer reviews. In *KDD*, Seiten 168–177, 2004.
- [KH06] Soo-Min Kim und Eduard H. Hovy. Automatic Identification of Pro and Con Reasons in Online Reviews. In *ACL*, 2006.
- [KK07] Nobuhiro Kaji und Masaru Kitsuregawa. Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents. In *EMNLP-CoNLL*, Seiten 1075–1083, 2007.
- [KM03] Dan Klein und Christopher D. Manning. Accurate Unlexicalized Parsing. In *ACL*, Seiten 423–430, 2003.
- [Liu10] Bing Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*. Taylor and Francis Group, Boca, 2010.
- [Luc12] Laurent Luce. Twitter sentiment analysis using Python and NLTK, 2012. <http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk>.
- [ME10] Samaneh Moghaddam und Martin Ester. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *CIKM*, Seiten 1825–1828, 2010.
- [Mil95] George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, 1995.
- [PLV02] Bo Pang, Lillian Lee und Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. *CoRR*, cs.CL/0205070, 2002.
- [SBMN13] Richard Socher, John Bauer, Christopher D. Manning und Andrew Y. Ng. Parsing with Compositional Vector Grammars. In *ACL (1)*, Seiten 455–465, 2013.
- [SPW<sup>+</sup>13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng und Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seiten 1631–1642. Association for Computational Linguistics, 2013.
- [TP12] Mikalai Tsytarou und Themis Palpanas. Survey on mining subjective data on the web. *Data Min. Knowl. Discov.*, 24(3):478–514, 2012.
- [Vec10] Olga Vechtomova. Facet-based opinion retrieval from blogs. *Inf. Process. Manage.*, 46(1):71–88, 2010.
- [ZY08] Yu Zhang und Dit-Yan Yeung. Semi-Supervised Discriminant Analysis using robust path-based similarity. In *CVPR*, 2008.