

Dokumentation: Irony Detection on Amazon Customer Reviews

Max Blunck, Steffen Knapp, Lennart Schönwandt, Daniel Wachter

8th February 2018

Contents

1	Motivation & Ziele	2
1.1	Motivation	2
1.2	Ziele	2
2	Grundlagen	2
2.1	Daten	2
3	Features	4
3.1	F1: Bag of n-Grams	4
3.2	F2: Bag of POS-Bigrams	4
3.3	F3: Surface-Pattern Feature	4
3.4	F4: Sentiment/Rating Feature	4
3.5	F5: Punctuation Features	4
3.6	F6: Contrast-Feature	4
3.7	F7: Star-Rating	5
4	Implementation	5
4.1	Tools	5
5	Evaluation	5
5.1	Baseline	5
5.2	Ergebnisse	6
5.2.1	Feature-Kombinationen	6
5.2.2	Features isoliert F_1 -Scores	7
6	Zusammenfassung & Ausblick	7
7	Aufgabenverteilung & Zeitplan	8
8	Literaturverzeichnis	9

1 Motivation & Ziele

1.1 Motivation

Ironie und Sarkasmus sind weit verbreitete sprachliche Phänomene, deren Gebrauch und Erkennung menschlichen Sprechern meist intuitiv zugänglich sind, die durch ihre inhärente Implizitheit von verschiedenen Sprechern allerdings oft auch unterschiedlich bewertet werden. Ist in der mündlichen Rede die Intonation in vielen Fällen noch ein guter Indikator, so sind Ironie und Sarkasmus in der Schriftsprache meist nur durch ihren Kontext oder Weltwissen verstehbar. Eine zuverlässige Erkennung dieses Phänomens kann dabei helfen, die Performanz vieler NLP-Systeme signifikant zu verbessern, insbesondere in den Bereichen Sentiment-Analyse und Opinion-Mining. Einem direkten kommerziellen Interesse entspricht der Aufgabe, Meinungsäußerungen und insbesondere Produktbewertungen auf Handelsplattformen im Internet besser klassifizieren und auswerten zu können.

In der Forschungsliteratur herrscht weitgehende Einigkeit über die Unterscheidung zwischen verbaler und situativer Ironie: Verbale Ironie wird zumeist so verstanden, dass ein Sprecher das Gegenteil (oder zumindest etwas davon sehr verschiedenes) des Gesagten meint (Colston and Gibbs, 2007), situative Ironie geht dagegen mit einer Erwartungsenttäuschung, z.B. der überraschenden Abweichung von einer erwartbaren Ereigniskette einher (Lucariello, 1994). Im Kontext von Produktbewertungen spielt allerdings fast ausschließlich verbale Ironie eine Rolle. Sarkasmus verstehen wir, der gängigen Meinung folgend (Dews and Winner, 1995), als besonders „scharfe“ Form von Ironie, die häufig einen persönlichen Angriff beinhaltet.

Zwar existieren neben dieser grundlegenden Unterscheidung einige pragmatische Theorien zu Ironie (Grice, 1975; Utsumi, 1996; Wilson and Sperber, 2007), die Forschung tendiert allerdings zu der Feststellung, dass eine hinreichend linguistisch-formale Beschreibung ein unerfülltes Desiderat bleiben muss.

1.2 Ziele

Unser Projektziel ist die Klassifizierung von ironischen und nicht ironischen Kundenrezensionen auf Amazon. Wir haben dazu einen machine learning Ansatz gewählt, für den wir sowohl Features nutzen, die relativ spezifisch für unsere Untersuchungsdomäne sind, wie z. B. das Amazon-Bewertungssystem mit der Vergabe von einem bis fünf Sternen, als auch Features, die domänenunabhängig funktionieren können, wie z.B. POS-tag-n-grams, Sentiment-Kontrastierung oder syntagmatische Strukturen.

Als Korpus für die Feature-Extraktion und das Training und Testing unseres Modells dient uns das Sarcasm Corpus von Elena Filatova (2012) und als Baseline die Ergebnisse von Buschmeier et al. (2014).

2 Grundlagen

2.1 Daten

Neben dem von uns genutzten Datensatz von Filatova sind uns zwei weitere Datensätze mit als ironisch annotierten Amazon-Rezensionen bekannt. Tsur et al. (2010) benutzen neben Twitter-Daten auch ein Amazon-Korpus bestehend aus 66000 Rezensionen zur Sarkasmuserkennung, allerdings wurde aus dem Datensatz nur ein sehr kleines Sample manuell als ironisch annotiert und dieses Sample aus einzelnen Sätzen dann mit einer automatischen Suche nach ähnlichen Sätzen angereichert. Dieser Datensatz ist also sowohl hinsichtlich seiner Diversität, als auch der Menge seiner qualitativen Vorannotation sehr limitiert. Zudem ist er nicht öffentlich verfügbar. Reyes and Rosso (2011) klassifizieren ebenfalls Amazon-Rezensionen als ironisch, allerdings besteht ihr Datensatz lediglich aus Rezensionen zu fünf unterschiedlichen Produkten. Außerdem haben sie automatisch Rezensionen herausgefiltert, die weniger als vier Sterne als Bewertung vergeben haben, was wir angesichts unserer eigenen Ergebnisse auf dem Filatova-Korpus als ungeeignet einschätzen.

Als Korpus für das Training unseres Modells nutzen wir das „Sarcasm Corpus“ von Elena Filatova (2012). Dieses Korpus besteht aus 1254 Amazon Reviews, welche mithilfe von Amazons

Mechanical Turk¹ annotiert wurden und von denen 437 als ironisch/sarkastisch und 817 als nicht-ironisch/sarkastisch gelabelt wurden. Es enthält nicht nur einzelne Sätze, sondern jeweils die komplette Rezension zu einem Produkt, sodass sich dieses Korpus insbesondere für eine kontextsensitive Ironieerkennung eignet.

Die Korpusannotation wurde in einem zweistufigen Verfahren durchgeführt, welche die Schritte Data Collection und Data Quality Control umfasst: In einem ersten Schritt sollten von MTurk-ers 1000 Paare von ironischen oder sarkastischen und normalen Rezensionen zu jeweils demselben Produkt auf Amazon gefunden und gelabelt werden, sowie die jeweiligen Review-Texte, das Label und der dazugehörige Link gespeichert werden. Diese Daten wurden bereinigt und fünf weiteren MTurk-ers vorgelegt, die die Zuverlässigkeit der gelabelten Rezensionen bewerten sollten. Dabei wurde ein einfaches Mehrheitsverfahren, sowie Krippendorff's alpha coefficient zur Unterscheidung zwischen zuverlässigen und unzuverlässigen Annotatoren benutzt. Durch dieses Verfahren sollte sichergestellt werden, dass nur sehr eindeutige Daten in das Korpus gelangen, was zu den 437 ironisch/sarkastisch und 817 nicht-ironisch/sarkastischen Rezensionen führte. In einem letzten Schritt sollten die Annotatoren schließlich nur anhand des Rezensionstextes die Sternbewertung der Rezension schätzen. Dieser Korrelationskoeffizient fiel mit 0,889 sehr hoch aus und ist ein erstes Indiz dafür, dass die Sternbewertung bei Amazon-Reviews ein guter Indikator für Ironie ist.

Filatova selbst macht keine Angabe dazu, welchen Anteil ironische Reviews auf Amazon insgesamt ausmachen. Für andere Domänen lassen sich Werte zwischen 8% (Gibbs, 2007: Personen, die mit einem ironischen Umgang untereinander vertraut sind) und 0,25% (Khodak et al., 2017: Sarkasmus-Tags auf Reddit) finden.

		Number of reviews with				
		1 ★	2 ★	3 ★	4 ★	5 ★
sarcastic	437	262	27	20	14	114
regular	817	64	17	35	96	605

Auffällig in dem Korpus ist die Tendenz sarkastischer Reviews, „extrem“ (entweder 1 oder 5) und überwiegend negativ (mit nur einem Stern) zu bewerten. Diese Tendenz hat sich als ziemliches zuverlässiges Feature zur Ironieerkennung erwiesen (vgl. auch Buschmeier et al. 2014), allerdings ist sie recht korpuspezifisch, da sie natürlich nur bei Reviews mit vergleichbaren expliziten Bewertungsmetadaten funktioniert.

Eine erste Korpusanalyse zur Verteilung der Tokens zeigt, dass von 21.744 voneinander verschiedenen Tokens (Wortform) 5.336 Tokens nur in ironischen Reviews und 9.468 Tokens nur in normalen Reviews vorkommen. Eine mögliche Erklärung für diese ungleichmäßige lexikalische Verteilung könnte der Inhalt des Korpus sein: Unterschiedliche Produkte spannen unterschiedliche lexikalische Felder auf. Auch dieses Merkmal lässt sich daher gut für unser Modell nutzen, ist aber wiederum ziemlich korpuspezifisch und umso stärker ausgeprägt, je kleiner das Korpus ist.

Beispielrezension aus dem Korpus:

```
<STARS>1.0 </STARS>
<TITLE>The title is the best part of the book</TITLE >
<DATE>July 25, 1998 </DATE>
<AUTHOR>A Customer </AUTHOR>
<PRODUCT>The God of Small Things (Paperback)</PRODUCT >
<REVIEW>
If you enjoy repeated phrases (e.g. „fountain in a Love-In-Tokyo“), overused metaphors,
and capitalized words in the Middle Of Sentences For Effect, then this is the book for
you. Otherwise, rest assured that the title really is the best part of the book.
</REVIEW>
```

¹<https://www.mturk.com/mturk/welcome>

3 Features

3.1 F1: Bag of n-Grams

Wort-n-Grams werden nach dem Bag-of-Word Prinzip mithilfe des CountVectorizer von scikit-learn extrahiert und vektorisiert. Jedes n-Gram entspricht einem Feature, dessen Wert der Vorkommenshäufigkeit des n-Grams entspricht. Hierbei ist das n variabel ($n = \{1,2,3,\dots\}$).

3.2 F2: Bag of POS-Bigrams

Mit dem NLTK POS-Tagger haben wir die einzelnen Korpusinstanzen automatisch getaggt und durch ein eigenes Skript POS-Bigramme generiert. Es wird dann ein Vokabular aller Bigrame der Trainingsdaten angelegt, auf dessen Grundlage dann Vorkommenshäufigkeiten der Bigrame einer Korpusinstanz (ein Feature pro Vokabulareintrag) extrahiert werden können.

3.3 F3: Surface-Pattern Feature

Beim Surface-Pattern Feature wird eine Liste erstellt, in welcher alle Wörter, die frequenter als ein bestimmter threshold (1.000/1.000.000) sind, als high-frequency-words (HFWs) gespeichert werden. Zusätzlich werden auch die Punctuationen als HFWs gespeichert. Alle anderen Wörter im Korpus werden mit einem Platzhalter (CW - content word) ersetzt. Diese neu erstellten Strukturen werden wie in F1 nach dem Bag-of-Words-Ansatz vektorisiert (ebenfalls mit variablem n-Parameter).

Beispiel:

- This CW is about as CW as a CW.

3.4 F4: Sentiment/Rating Feature

Bei diesem Feature nutzen wir zum ersten mal die Star-Ratings von Amazon. Ziel des Features ist es, einen Kontrast zwischen dem Sentiment des Reviews und der Bewertung des Produkts zu finden. Sowohl ein negatives Rating (1 Stern) und Review mit positivem Sentiment, als auch ein positives Rating (5 Sterne) und Review mit negativem Sentiment können auf Ironie hinweisen.

Der Featurevektor ist hier binär: Wenn ein Kontrast zwischen Rating und Review Sentiment vorliegt, wird der Wert auf 1 gesetzt, ansonsten auf 0.

3.5 F5: Punctuation Features

Bei diesem Feature untersuchen wir Muster in der Punctuation der Reviews einschließlich Titel. Wir haben eine Liste mit unseres Erachtens relevanter Punctuation als feste Variable implementiert. Diese umfasst Ausrufezeichen, Fragezeichen und jegliche Kombinationen aus diesen (zum Beispiel '???', '!!!', '?!', '!?!?'), sowie Ellipsen und Anführungszeichen. Außerdem versuchen wir, den Gebrauch von Allcaps zu erkennen. Dies erweist sich jedoch als schwierig, wenn man es mit Worten wie DVD, TV oder USA zu tun hat. Ein dynamisches Generieren einer Liste mit zu ignorierenden Wörtern ist nicht leicht möglich, eine Hardcodierung wollten wir aber auch umgehen. Letztenendes wird bei diesem Feature jedes Wort als Allcaps erkannt, das länger als zwei Buchstaben ist und nicht ausschließlich aus Konsonanten besteht. Dies stellt eine Art Mittelweg dar und umgeht erfolgreich Wörter wie 'I', 'TV' und 'DVD'; 'NASA' und andere Akronyme bleiben aber erhalten.

3.6 F6: Contrast-Feature

Diese Featureidee stammt von Riloff et al. (2013). Hierbei geht es darum, Sentiment-Kontraste innerhalb eines Satzes zu finden. Genauer gesagt suchen wir ein positives (Adverb+)Verb auf welches eine negative Situation folgt.

Beispiel:

- Absolutely **adore** it **when** my **bus** is **late**.
[+ **positives** Verb] [- **negative** Situation]

Zum Aufspüren solcher Konstruktionen machten wir von POS-tag-Kombinationen gebrauch: Zunächst haben wir ein Verb gesucht, welches einen positiven Sentiment aufweist. Daraufhin haben wir nach vorab definierten POS-Uni/Bi/Tri-gram-Reihenfolgen gesucht, mit denen sich Situationssyntax realisierten lassen. Diese Phrasen haben wir anschließend auf einen negativen Sentiment untersucht. Sollte es zwischen ihnen und dem (Adverb+)Verb einen Kontrast im Sentiment geben, so wird der Feature-Wert um eines erhöht.

3.7 F7: Star-Rating

Für dieses Feature werden einfach nur die einzelnen Star-Ratings der Reviews extrahiert. Jedes einzelne Rating repräsentiert ein Feature im Featurevektor.

4 Implementation

Alle Programmteile sind in Python 3.6 geschrieben. Die grundlegende Struktur basiert auf einer modularen Aufteilung nach obigen Features, sowie einem Hauptprogramm, das alle Schritte des Lernverfahrens zusammenführt. Zu diesen Schritten gehört u.a. das Einlesen, Anreichern und Aufteilen des Datensets, das Extrahieren der Features, das Trainieren und Testen verschiedener Klassifizierer sowie das Ausgeben von Ergebnissen.

Zur Klassifikation nutzen wir die Algorithmen SVM, Logistic Regression, Multinomial Naive Bayes und Decision Trees. Zudem haben wir Optionen implementiert, um u.a. Kreuzvalidierung anzuwenden, Klassifizierer zu tunen sowie Featurekombinationen auszuwählen.

Zur einfacheren Handhabung des Datensets wurde außerdem das gesamte Korpus in eine einzelne Datei gespeichert, mit der allein weitergearbeitet wurde.

4.1 Tools

scikit-learn

Das Toolkit scikit-learn war in vielerlei Hinsicht nützlich für uns. Wir konnten auf alle nötigen Klassifizierer (SVM, Logistic Regression, Multinomial Naive Bayes, Decision Trees) zugreifen, deren GridSearch nutzen, um die besten Parameter für die Klassifizierer zu finden, sowie Kreuzvalidierung ausführen.

Zudem stellt scikit-learn das Modul CountVectorizer zur Verfügung, welches zur Extraktion der Features F1 und F3 diente.

TextBlob

TextBlob² ist eine Python Library zum Verarbeiten von Textdateien, welche man unter anderem auch für Sentiment Analysis nutzen kann. Wir benutzen TextBlob um den Sentiment von ganzen Reviews oder einzelnen Phrasen zu bestimmen.

NLTK

Von NLTK nutzen wir den POS-Tagger um die benötigten part-of-speech-tags für F2 zu erhalten. Auch zur Tokenisierung griffen wir auf dieses Tool zurück.

5 Evaluation

5.1 Baseline

Als Baseline dienen uns die Ergebnisse von Buschmeier et al.³, die mit korpuspezifischen Features für die automatische Klassifikation ironischer Kundenbewertungen auf Amazon einen F_1 -Score von

²<http://textblob.readthedocs.io/en/dev/>

³Buschmeier et al. 2014. An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews.

guten 74 % erzielen konnten, bei problemspezifischeren Features jedoch deutlich niedrigere Werte erzielten.

5.2 Ergebnisse

Wir haben beim Testen zum einen alle Feature-Kombinationen, aber auch die einzelnen Features an sich ausprobiert und getestet. Die besten Feature-Kombinationen haben wir herausgesucht und in der untenstehenden Tabelle (5.2.1) zusammengefasst.

Die Ergebnisse der einzelnen Features sind in der Tabelle unter 5.2.2 zu sehen.

Wie bei unserem Vergleichspaper von Buschmeier et al. testen wir mit einer 10 fold cross-validation (CV scores), aber auch mit einer Korpusaufteilung von 80/20 (Test scores), welche wir demnach aber nicht mit unserer Baseline vergleichen können.

5.2.1 Feature-Kombinationen

Features	Algorithm	Test scores			CV scores		
		Precision	Recall	F_1 -Score	Precision	Recall	F_1 -Score
all	SVM	74.0	77.1	75.5	73.3	70.3	71.4 71.3
	DT	87.0	69.8	77.5	80.2	65.7	72.1 72.2
	NB	82.4	14.6	24.8	66.5	71.8	68.9 65.0
	LR	79.3	71.9	75.4	77.9	68.4	72.4 74.4
Best	SVM	81.3	77.1	79.1	75.4	73.0	74.0
Test score:	DT	80.7	69.8	74.9	77.8	68.0	72.2
f1+f3+f4+f7	NB	85.7	6.2	11.7	63.5	72.8	67.5
	LR	84.1	60.4	70.3	81.4	69.8	74.9
Best	SVM	78.6	68.8	73.3	78.8	70.7	74.2
CV score:	DT	69.7	71.9	70.8	78.4	65.0	70.7
f1+f4+f5+f7	NB	88.6	32.3	47.3	61.2	77.1	68.0
	LR	85.5	49.0	62.3	80.4	70.7	76.4
f1+f2+f3	SVM	69.2	65.6	67.4	68.9	62.7	65.4
	DT	55.3	27.1	36.4	54.5	32.2	39.7
no stars	NB	63.1	55.2	58.9	63.7	70.2	66.5
	LR	78.2	63.5	70.1	73.4	62.7	67.0

Im ersten Block „all“ stehen die Ergebnisse, die wir mit der Kombination von allen sieben Features erreicht haben. Die orange markierten Zahlen sind die Ergebnisse der Baseline.

Unseren besten Wert haben wir mit Logistic Regression erzielt. Hier haben wir einen F_1 -Score von 72.4%, der aber dennoch zwei Prozentpunkte unter der Baseline liegt. Dafür war unsere Klassifikation mit dem Naive Bayes erfolgreicher. Hier haben wir mit 68.9% einen Wert der knapp 4% über der Baseline liegt. Ähnliche Werte haben wir jeweils mit SVM und Decision Trees erzielt.

Im zweiten Block haben wir die Feature-Kombination, welche die besten Werte auf unseren Testdaten (Test scores) erzielt hat. In diesem Fall nutzen wir bei unserem N-gram Feature F1 Bigramme. Hier liegt unser bester Wert bei einem F_1 -Score von 79.1 % mit SVM.

Unser bestes Ergebnis bei der cross-validation haben wir wie auch schon bei der Kombination aller Features mit LR erzielt. Hier haben wir bei F1 sowohl Uni- als auch Bigramme genutzt. Mit einem F_1 -Score von 76.4% haben wir hier auch den besten Wert der Baseline übertroffen.

Wenn man die beste Feature-Kombination betrachtet, ohne die für Amazon spezifischen Star-Ratings zu berücksichtigen, also eine mögliche Feature-Kombination, die auch auf andere Daten anwendbar ist, erhält man im Vergleich eher schlechtere Werte, als mit den vorherigen Ergebnissen. Hier erhalten wir einen F_1 -Score von gerade mal 67%. Jedoch zeigen auch die Ergebnisse der Baseline (F_1 -Score: 67,8%), dass die Ergebnisse ohne Berücksichtigung der Star-Ratings eher schlechtere Werte erzielen.

5.2.2 Features isoliert F_1 -Scores

Feature	SVM	DT	N. Bayes	LR
F1: Bag-of-ngrams	66.7	33.3	56.6	59.5
F2: Bag-of-POS	57.1	39.0	54.5	53.1
F3: Surface-Patterns	55.8	38.3	51.7	57.1
F4: Sent/Rating	65.1	65.1	0.0	0.0
F5: Punctuation	0.0	42.3	0.0	0.0
F6: Contrast-Feat.	0.0	0.7	0.0	0.0
F7: Stars	71.2	71.2	0.0	69.1

Wenn man die einzelnen Features isoliert betrachtet erkennt man sofort, dass F7 mit einem F_1 -Score von 71.2% den besten Wert erzielt. Auch der Kontrast zwischen Rating und Review-Sentiment (F4) sowie das ngram-Feature (F1) erzielen gute Werte.

Unser Contrast-Feature (F6) erzielt leider sehr schlechte Werte, da es in unserem kleinen Korpus kaum Treffer dieser Art gab.

Entgegen unserer Erwartungen erzielt das Surface-Pattern-Feature (F3) den besten Wert bei der Betrachtung von Uni-grammen, wobei die eigentliche Intension dieses Features eigentlich das Finden von längeren Strukturmustern war.

6 Zusammenfassung & Ausblick

Die Ergebnisse unserer performantesten Featurekombinationen zeigen, dass die stärksten Features auch die korpuspezifischsten sind: Die Klassifikation mithilfe von Wort-Bigrammen profitiert von der lexikalischen Dispersion des Korpus, die bereits eine erste Analyse gezeigt hatte, und den schon von Filatova gezeigten Effekt, dass sich die Sternebewertung recht zuverlässig aus der Rezension vorhersagen lässt, nutzen wir in umgekehrter Richtung - zwei unserer drei besten Features basieren auf der Anzahl der Sterne, die eine Rezension vergibt. Daher können unsere Ergebnisse nur bedingt auf die Ironieerkennung in anderen Domänen übertragen werden, bzw. es zeigt sich, dass Ironiemarker meist eher domänenspezifisch als linguistisch-formaler Natur sind und sie sich daher nur recht schwer domänenübergreifend erkennen lassen können.

Mögliche weitere Schritte wären etwa die Verfeinerung unserer bereits existierenden Features, bzw. die Entwicklung komplett neuer Features, zu vermuten ist allerdings, dass sich der Einfluss der Domäne und der Korpuspezifika auf diesem Weg nicht grundlegend ändern werden.

Eine Analyse fehlerhaft klassifizierter Beispielrezensionen zeigt darüber hinaus, dass häufig die Erkennung von Ironie dort an ihre Grenzen stößt, wo sehr nuanciertes Kontextwissen oder erweitertes Weltwissen erforderlich ist. Dementsprechend wurden unsere besten Ergebnisse in allen Fällen auch von einem zu niedrigen Recall gedrückt, während die Precision meist deutlich höher ist.

3 of 3 people found the following review helpful:

★★★★★ **Best face lotion out there... Period.**, July 24, 2006

By **L. F. Bretts**

If you are looking for a lotion that won't leave your face feeling greasy, then look no further. It is THE best out there.

Help other customers find the most helpful reviews

Was this review helpful to you?

[Report this](#) | [Permalink](#)

Review Details

Item not available

Reviewer

 **L. F. Bretts**
REAL NAME™

Location: San Diego, CA

New Reviewer Rank: 3,787,798
Classic Reviewer Rank: 1,236,697

Label: nicht ironisch; Vorhersage: ironisch

17 of 17 people found the following review helpful:

★★★★★ **OMG, so great**, October 3, 2009

By **Robert D. Walton "Wolf Heart"**

This review is from: **Handerpants (Misc.)**

I mean, I always wanted my crotch and my hands to have more in common, now they do!

Help other customers find the most helpful reviews | [Report this](#) | [Permalink](#)

Was this review helpful to you?

Review Details


Item

[Handerpants](#)

★★★★☆ (4 customer reviews)

5 star:  (2)

4 star:  (1)

3 star:  (0)

2 star:  (0)

1 star:  (1)

\$7.99

Label: ironisch; Vorhersage: nicht ironisch

7 Aufgabenverteilung & Zeitplan

	Aufgabe	Teilnehmer
Features	Implementation PP	Max
	Implementation F1	Lennart
	Implementation F2	Steffen
	Implementation F3	Daniel
	Implementation F4	Max
	Implementation F5	Steffen
	Implementation F6	Max
	Implementation F7	Max
Weitere Programmstrukturen	Implementation TT	Max
	Implementation CL	Daniel
	Implementation FE	Lennart
	Implementation CF	Steffen
Drumherum	Präsentationen	alle
	Dokumentation	alle
	Literaturrecherche	alle
	Statusberichte	Max

x: Plan; x: Umsetzung

Task	Dec 1	Dec 2	Dec 3	Dec 4	Jan 1	Jan 2	Jan 3	Jan 4	Feb 1
Spezifikation	x								
	x								
Feature-Implementation		x	x	x					
		x	x	x	x	x			
Evaluation					x	x			
						x	x	x	x
Tuning							x	x	
							x	x	x
Abschließende Dokumentation							x	x	
									x
Präsentation und Abgabe									x
								x	x

8 Literaturverzeichnis

- Buschmeier et al. 2014. An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews.
- D. Davidov. O. Tsur. and A. Rappoport. 2010. Semisupervised recognition of sarcastic sentences in Twitter and Amazon. In Proceeding of the 23rd international conference on Computational Linguistics. July.
- Shelly Dews and Ellen Winner. 1995. Muting the meaning: A social function of irony. *Metaphor and Symbolic Activity*, 10(1):3–19.
- Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Language Resources and Evaluation Conference. LREC2012*.
- Raymond W. Jr. Gibbs. 2007. Irony in talk among friends. In Raymond W. Jr. Gibbs and Herbert L. Colston, editors, *Irony in Language and Thought: A Cognitive Science Reader*, chapter 15, pages 339–360. Lawrence Erlbaum Associates, 1st edition, May.
- R. Gibbs and H. Colston. 2007. The future of irony studies. In R. Gibbs and H. Colston, editors, *Irony in Language and Thought*, pages 339–360. Taylor and Francis Group.
- Paul Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3: Speech Acts:41–58.
- Mingqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2004)*. pages 168–177. Seattle. WA. USA. August.
- Mikhail Khodak, Nikunj Saunshi and Kiran Vodrahalli. 2017. A Large Self-Annotated Corpus for Sarcasm.
- Joan Lucariello. 1994. Situational irony: A concept of events gone awry. *Journal of Experimental Psychology: General*, 123(2):129–145.
- Antonio Reyes. Paolo Rosso. 2011. Mining Subjective Knowledge from Customer Reviews: A Specific Case of Irony Detection.
- Ellen Riloff. Ashequl Qadir. Prafulla Surve. Lalindra De Silva. Nathan Gilbert. Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation.
- A. Utsumi. 1996. A unified theory of irony and its computational formalization. In *Proceedings of the 16th conference on Computational Linguistics*, pages 962–967, Morristown, NJ, USA. Association for Computational Linguistics.
- D. Wilson and D. Sperber. 2007. On verbal irony. In R. Gibbs and H. Colston, editors, *Irony in Language and Thought*, pages 35–56. Taylor and Francis Group.