

# ICWSM - A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews

Referenten: Dung Nguyen, Maryna Charniuk

Proseminar "Einführung in die Sentiment Analyse"

Dr. Michael Wiegand

WS 19/20

Institut für Computerlinguistik

Ruprecht-Karls Universität Heidelberg

11.11.2019

# Gliederung

## Einführung

- Motivation

- Was ist Sarkasmus?

## Modell

## Algorithmus

- Features

- Datenerweiterung

- Klassifikation

- Baseline

## Daten

## Evaluation

- 5-Fache Kreuzvalidierung

- Klassifikation der neuen ungesehenen Sätzen

## Ergebnisse

## Fazit

## Einführung

Motivation

Was ist Sarkasmus?

## Modell

## Algorithmus

Features

Datenerweiterung

Klassifikation

Baseline

## Daten

## Evaluation

5-Fache Kreuzvalidierung

Klassifikation der neuen ungesehenen Sätzen

## Ergebnisse

## Fazit

# Motivation

Die Liste "**Die 50 Beste Erfindungen von 2010**" ("The 50 Best Inventions of 2010") des amerikanischen Nachrichtenmagazins "Time"<sup>1</sup>:

- Kategorie ...
- Kategorie **Robotik/Software**
  - The Deceitful Robot
  - The English-Teaching Robot
  - Lifeguard Robot
  - **Sarcasm Detection**
- Kategorie ...

---

<sup>1</sup><http://content.time.com/time/specials/packages/completelist/0,29569,2029497,00.html>

# Motivation

Für welche CL-Aufgaben kann automatische Erkennung von Sarkasmus besonders nützlich sein?

- Studien zu Benutzerpräferenzen über hilfreiche Rezensionen
- Verbesserung der Personalisierung von Content-Ranking und Empfehlungssystemen
- Verbesserung der Zusammenfassung von Rezensionen und Opinion Mining-Systemen

## Was ist Sarkasmus?

**Macmillian English Dictionary, 2007:** *The activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry.*

**Duden Wörterbuch:** *beißender, verletzender Spott, Hohn, der jemanden, etwas lächerlich machen will*

## Beispiele

Sarkastische Äußerungen findet man oft in von Nutzern erstellten Inhalten wie Blogeinträgen und Produktbewertungen.

Beispiele für sarkastische Überschriften aus Bewertungen:

- *[!] Love The Cover* (Buch)
- *Great idea, now try again with a real product development team* (E-Reader)
- *Great for insomniacs* (Buch)
- *Just read the book!* (Buch vs. Kino)

## Einführung

Motivation

Was ist Sarkasmus?

## Modell

### Algorithmus

Features

Datenerweiterung

Klassifikation

Baseline

### Daten

### Evaluation

5-Fache Kreuzvalidierung

Klassifikation der neuen ungesehenen Sätzen

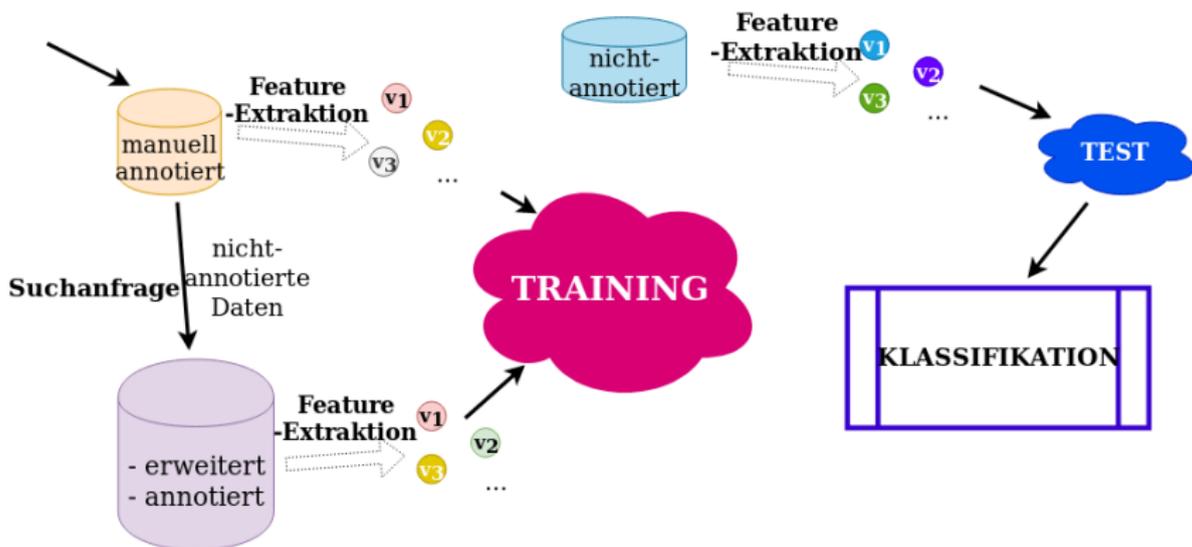
### Ergebnisse

### Fazit

# Überblick

- SASI-Algorithmus benutzt klassisches halb-überwachtes Lernen
- benötigte Daten:
  - eine kleine Menge von manuell annotierten Daten (*seeds*)
  - eine große Menge von nicht-annotierten Daten

# Modell



$v$ : Feature-Vektor

- Sätze im Test Set sind klassifiziert, unter Verwendung von Sätzen im annotierten Training Set.

## Output

Satz	Score
Satz 1	2
Satz 2	1
Satz 3	5
...	



- Je höher der Score ist, desto sicherer ist der Satz sarkastisch.

## Einführung

- Motivation

- Was ist Sarkasmus?

## Modell

## Algorithmus

- Features

- Datenerweiterung

- Klassifikation

- Baseline

## Daten

## Evaluation

- 5-Fache Kreuzvalidierung

- Klassifikation der neuen ungesehenen Sätzen

## Ergebnisse

## Fazit

# Datenvorverarbeitung

- relevante Features basieren auf **oberflächlichen Mustern**
- irrelevante Informationen : autoren-, produktspezifisch  
**Bsp.:** Namen der Autoren, Produkten, Firmen/Hersteller, Titel  
→ durch Tags *[author]*, *[product]*, *[company]*, *[title]* ersetzt

# Datenvorverarbeitung

## Beispiel

Die **Armbanduhr** von **Welly Merck** wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die **Uhr** leider nur entweder vergrößern oder verkleinern.

# Datenvorverarbeitung

## Beispiel

Die **Armbanduhr** von **Welly Merck** wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die **Uhr** leider nur entweder vergrößern oder verkleinern.

## Nach der Vorverarbeitung

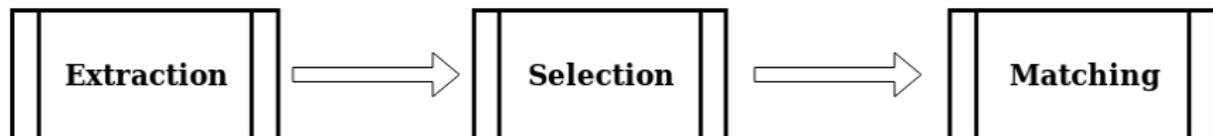
Die **[product]** von **[company]** wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die **[product]** leider nur entweder vergrößern oder verkleinern.

## Satzzeichen-basierte Features

- Satzlänge
- Anzahl von '!'
- Anzahl von '?'
- Anzahl von '"'
- Anzahl von großgeschriebenen Worten

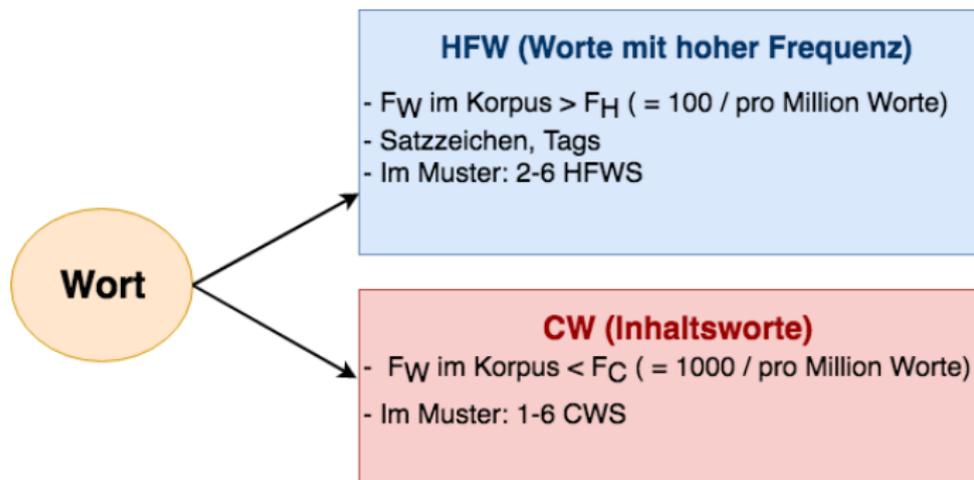
## Muster-basierte Features

**Pipeline:**



## Muster-Extraktion

Klassifikation von Wörtern in jedem Muster:



$F_W$  : Wortfrequenz

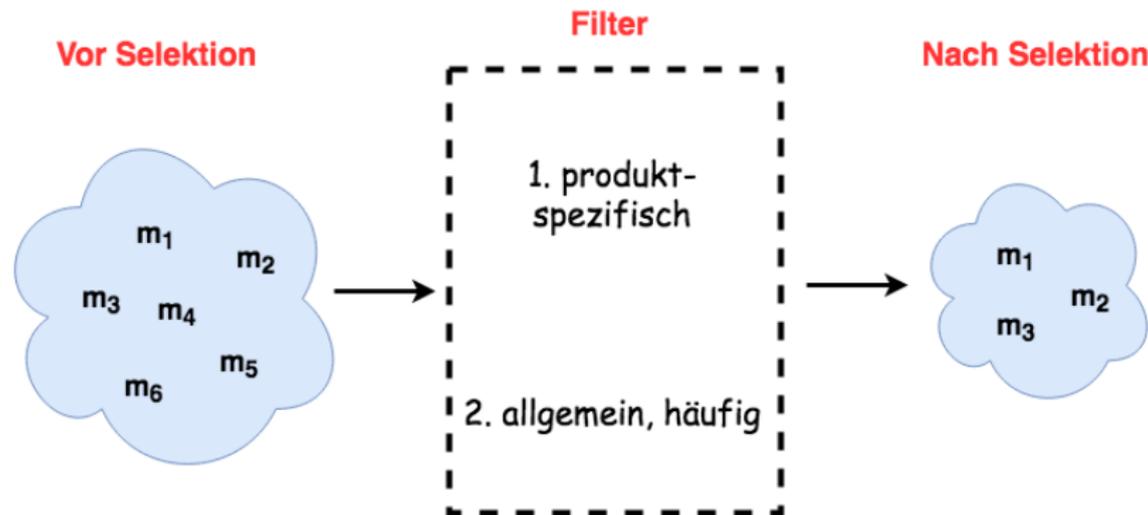
$F_H$  : Grenzwert für Frequenz von *HFW*

$F_C$  : Grenzwert für Frequenz von *CW*

## Muster-Extraktion

- Alle Muster fangen an und enden mit *HFW* (Vermeidung von Phraseologismen)
- Muster: **HFW {HFW|CW}\* HFW**

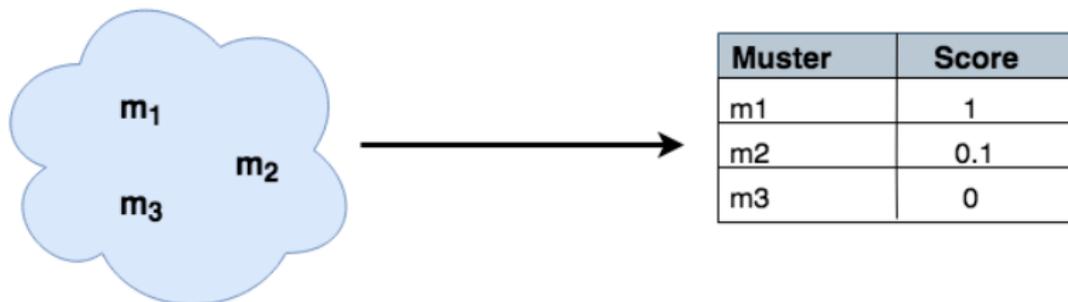
## Muster-Selektion



$m$ : Muster

- produktspezifische oder allgemeine Muster sind ausgefiltert.

## Muster-Matching



- Jedes Muster bekommt einen Score abhängig von seinem Match-Typ.

## Muster-Matching

Typ	Score	Erklärung
exact match	1	völlige Übereinstimmung zwischen Muster und Substring im Satz
sparse match	$\alpha = 0.1$	wie <i>exact match</i> , aber ein paar Wörter aus Substring kommen nicht im Muster vor
incomplete match	$x, 0 < x < 1$	wie <i>sparse match</i> , aber ein paar Teile im Muster haben keine Entsprechungen im Substring
no match	0	Muster ist gar nicht im Satz gefunden oder nur 1 Teil davon kommt vor

## Beispiel

*Die [product] von [company] wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die [product] leider nur entweder vergrößern oder verkleinern.*

### 1. Extraktion

## Beispiel

*Die [product] von [company] wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die [product] leider nur entweder vergrößern oder verkleinern.*

### 1. Extraktion

- [company] wurde CW und
- entweder CW oder CW .
- [company] CW und
- [company] CW wurde CW und
- mit CW wird
- ...

## Beispiel

*Die [product] von [company] wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die [product] leider nur entweder vergrößern oder verkleinern.*

### 1. Extraktion

- [company] wurde CW und
- entweder CW oder CW .
- [company] CW und
- [company] CW wurde CW und
- mit CW wird

...

### 2. Selektion

## Beispiel

*Die [product] von [company] wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die [product] leider nur entweder vergrößern oder verkleinern.*

### 1. Extraktion

- [company] wurde CW und
- entweder CW oder CW .
- [company] CW und
- [company] CW wurde CW und
- mit CW wird

...

### 2. Selektion

entweder CW oder CW .  $\implies$  ausgeschlossen (häufig vorkommt)

## Beispiel

*Die [product] von [company] wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die [product] leider nur entweder vergrößern oder verkleinern.*

### 1. Extraktion

- [company] wurde CW und
- entweder CW oder CW .
- [company] CW und
- [company] CW wurde CW und
- mit CW wird

...

### 2. Selektion

entweder CW oder CW .  $\implies$  ausgeschlossen (häufig vorkommt)

### 3 Matching

## Beispiel

*Die [product] von [company] wurde schnell und in einer schönen schwarzen Schachtel geliefert, aber durch ihren Mechanismus kann man die [product] leider nur entweder vergrößern oder verkleinern.*

### 1. Extraktion

- [company] wurde CW und
- entweder CW oder CW .
- [company] CW und
- [company] CW wurde CW und
- mit CW wird

...

### 2. Selektion

entweder CW oder CW .  $\implies$  ausgeschlossen (häufig vorkommt)

### 3 Matching

- [company] wurde CW und  $\implies 1$  (exact match)
- [company] CW und  $\implies 0.1$  (sparse match)
- [company] CW wurde CW und  $\implies 0.1 * 4/5 = 0.08$  (incomplete match)
- mit CW wird  $\implies 0$  (no match)

# Datenerweiterung

# Datenerweiterung

- Wie? : Suchanfragen in Suchmaschine

# Datenerweiterung

- Wie? : Suchanfragen in Suchmaschine
- Vorgehensweise:
  - Satz im Seed Training Set:  
**This book was really good until page 2!**

# Datenerweiterung

- Wie? : Suchanfragen in Suchmaschine
- Vorgehensweise:
  - Satz im Seed Training Set:  
**This book was really good until page 2!**
  - Suchanfrage:  
**this book was really good until**

# Datenerweiterung

- Wie? : Suchanfragen in Suchmaschine
- Vorgehensweise:
  - Satz im Seed Training Set:  
**This book was really good until page 2!**
  - Suchanfrage:  
**this book was really good until**
  - Treffer:  
**Gee, I thought this book was really good until I found out the author didn't get into Bread Loaf!**

# Datenerweiterung

- Wie? : Suchanfragen in Suchmaschine
- Vorgehensweise:
  - Satz im Seed Training Set:  
**This book was really good until page 2!**
  - Suchanfrage:  
**this book was really good until**
  - Treffer:  
**Gee, I thought this book was really good until I found out the author didn't get into Bread Loaf!**
  - Annahme: Sarkastische Sätze kommen oft in einem Text mit anderen sarkastischen Sätzen vor.  
→ **It just didn't make much sense.** (Kontext)

# Datenerweiterung

- Wie? : Suchanfragen in Suchmaschine
- Vorgehensweise:
  - Satz im Seed Training Set:  
**This book was really good until page 2!**
  - Suchanfrage:  
**this book was really good until**
  - Treffer:  
**Gee, I thought this book was really good until I found out the author didn't get into Bread Loaf!**
  - Annahme: Sarkastische Sätze kommen oft in einem Text mit anderen sarkastischen Sätzen vor.  
→ **It just didn't make much sense.** (Kontext)
  - Treffer haben dasselbe Label wie der Satz

# Klassifikation

**Strategie:** ähnlich wie *k-Nächste-Nachbar-Algorithmus* (kNN)

# Klassifikation

**Strategie:** ähnlich wie *k-Nächste-Nachbar-Algorithmus* (kNN)

Für jeden Vektor  $v$  im Testset:

- anpassende Vektoren aus Training Set : haben mind. ein gemeinsames Muster-Feature mit  $v$

# Klassifikation

**Strategie:** ähnlich wie *k-Nächste-Nachbar-Algorithmus* (kNN)

Für jeden Vektor  $v$  im Testset:

- anpassende Vektoren aus Training Set : haben mind. ein gemeinsames Muster-Feature mit  $v$
- $k$  anpassende Vektoren mit kleinsten Euklidischer Distanz zu  $v$

# Klassifikation

**Strategie:** ähnlich wie *k-Nächste-Nachbar-Algorithmus* (kNN)

Für jeden Vektor  $v$  im Testset:

- anpassende Vektoren aus Training Set : haben mind. ein gemeinsames Muster-Feature mit  $v$
- $k$  anpassende Vektoren mit kleinsten Euklidischer Distanz zu  $v$
- *Label/Score*:

# Klassifikation

**Strategie:** ähnlich wie *k-Nächste-Nachbar-Algorithmus* (kNN)

Für jeden Vektor  $v$  im Testset:

- anpassende Vektoren aus Training Set : haben mind. ein gemeinsames Muster-Feature mit  $v$
- $k$  anpassende Vektoren mit kleinsten Euklidischer Distanz zu  $v$
- *Label/Score*: gewichteter Durchschnitt von  $k$  nächsten Vektoren aus Trainingset

## Sentiment Baseline

Sarkasmus - *the activity of saying the opposite of what you mean in a way intended to make someone else feel stupid or show you are angry* (Macmilan 2007)

—→ **Annahme:** Sarkastische Sätze kommen oft in negativen Bewertungen vor, die positive Ausdrücken enthalten.

### Beispiel:

★★★★★ SMH

October 7, 2012

Format: Paperback | **Verified Purchase**

I am in love with this story and will forever be. It is truly an amazing love story that will never grow old.

- Dieses Beispiel bestätigt die Annahme, diese sind möglicherweise *sarkastische Sätze*.

## Einführung

- Motivation

- Was ist Sarkasmus?

## Modell

## Algorithmus

- Features

- Datenerweiterung

- Klassifikation

- Baseline

## Daten

## Evaluation

- 5-Fache Kreuzvalidierung

- Klassifikation der neuen ungesehenen Sätzen

## Ergebnisse

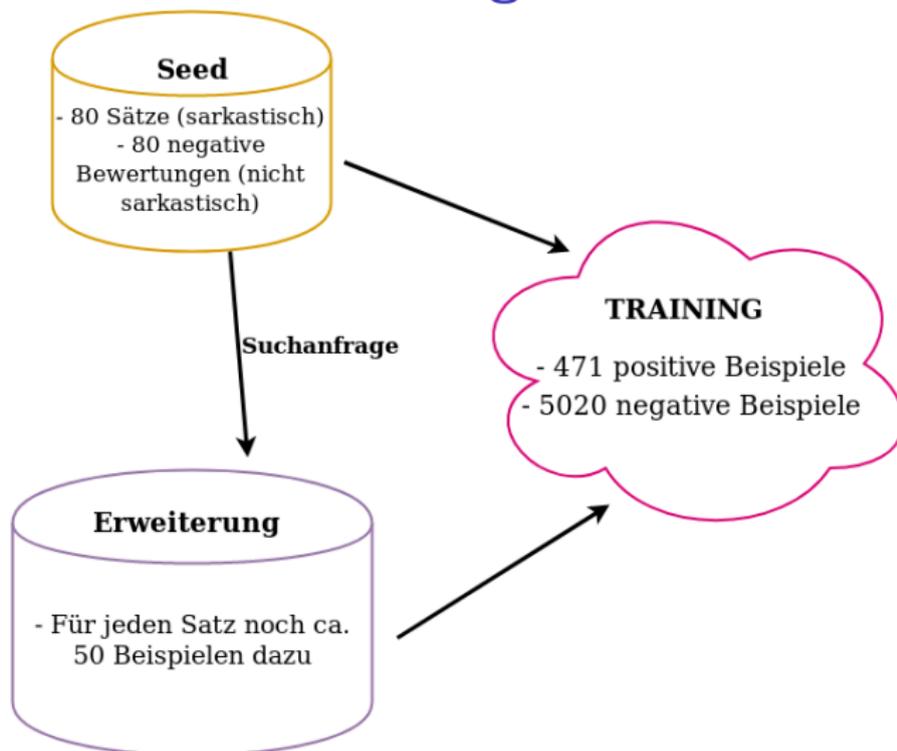
## Fazit

# Datensatz

- Online-Kundenrezensionen aus Amazon
- Aus verschiedenen Domänen: Bücher, Digitalkameras, GPS-Geräte, E-Reader, Spielekonsolen, Handys usw.
- Bewertung besteht aus 3 Teilen:
  - Sternebewertung
  - Zusammenfassung
  - Rumpf

#Produkte	#Bewertungen	durchschn. Sterne	durchschn. Länge (chars)
120	66 271	4.19	953

## Training Set



## Einführung

- Motivation

- Was ist Sarkasmus?

## Modell

## Algorithmus

- Features

- Datenerweiterung

- Klassifikation

- Baseline

## Daten

## Evaluation

- 5-Fache Kreuzvalidierung

- Klassifikation der neuen ungesehenen Sätzen

## Ergebnisse

## Fazit

## 5-fache Kreuzvalidierung

- Im ersten Experiment - Evaluation des Mustererfassungsprozess: wie gut jedes Feature ist und inwieweit es zur korrekten Klassifizierung beiträgt.
- **Ziel:** herausfinden welche Kombination von Features gibt das beste Ergebnis.

## Klassifikation der neuen ungesehenen Sätzen

- Im zweiten Experiment - Anwendung des SASI-Algorithmus auf ungesehenen Daten (Test-Set)
  - Test Korpus - 66 000 Bewertungen (Seed Training Korpus wurde ausgeschlossen)
- Output wurde mit Gold-Standard (manuell annotierte Sätze) verglichen.
  - Gold Standard - das kleine balancierte Test Set (180 Sätze)

## Manuelle Annotation

- Das Test Set wurde zufällig in 5 Teile aufgeteilt.
- Jeder Teil besteht aus 36 Sätzen + 4 gleichen Anker-Sätzen in 5 Teilen.
- Ziel der Anker-Sätzen: kontrollieren, dass Annotatoren die sarkastische Labels nicht zufällig vergeben.
- 15 Annotatoren  $\rightarrow$  jeder Satz wurde von drei verschiedenen Annotatoren annotiert.
- Inter-Annotatoren Agreement  $\kappa = 0.34$ .
- Agreement bei Anker-Sätzen  $\kappa = 0.53$ .

## Einführung

Motivation

Was ist Sarkasmus?

## Modell

## Algorithmus

Features

Datenerweiterung

Klassifikation

Baseline

## Daten

## Evaluation

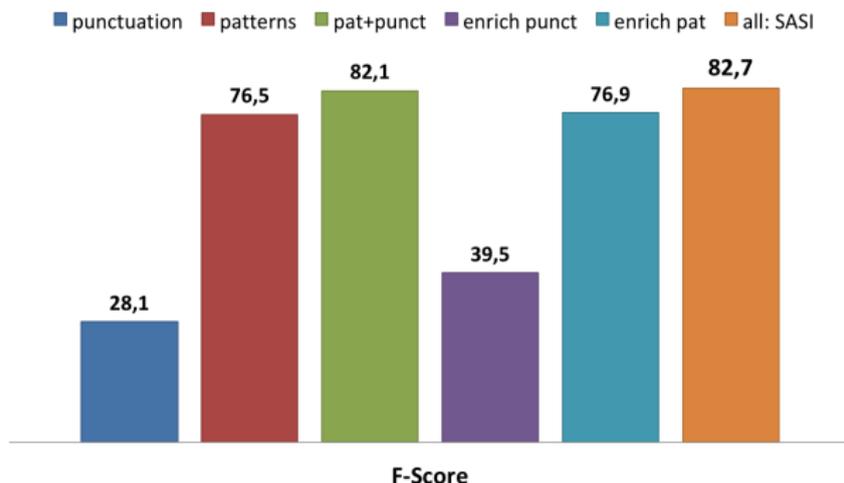
5-Fache Kreuzvalidierung

Klassifikation der neuen ungesehenen Sätzen

## Ergebnisse

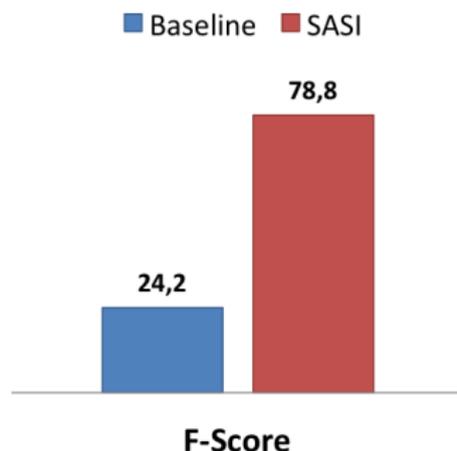
## Fazit

## Ergebnisse von 5-Facher Kreuzvalidierung



- Kombination von Features Satzzeichen und Mustern liefert bessere Ergebnisse, als einzelne Features
- Satzzeichen allein dienen als schwache Feature für Klassifikation

## Ergebnisse auf ungesehenen Daten



- Der SASI-Algorithmus liefert viel bessere Ergebnisse als Sentiment Baseline
- $\implies$  Die Meinung, dass Sarkasmus "saying or writing the opposite of what you mean" ist, wird widerlegt.

## Einführung

Motivation

Was ist Sarkasmus?

## Modell

## Algorithmus

Features

Datenerweiterung

Klassifikation

Baseline

## Daten

## Evaluation

5-Fache Kreuzvalidierung

Klassifikation der neuen ungesehenen Sätzen

## Ergebnisse

## Fazit

# Fazit

- SASI - Algorithmus zur Erkennung sarkastischer Sätze
- Halb-überwachtes Lernen
- Musterbasiertes Verfahren
- Ergebnisse des Experiments deutlich besser als Baseline

*Vielen Dank für Eure Aufmerksamkeit! :)*

## Literatur I



Dmitry Davidov und Ari Rappoport. „Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words“. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, Juli 2006, S. 297–304. DOI: 10.3115/1220175.1220213. URL: <https://www.aclweb.org/anthology/P06-1038>.



Dmitry Davidov und Ari Rappoport. „Unsupervised Discovery of Generic Relationships Using Pattern Clusters and its Evaluation by Automatically Generated SAT Analogy Questions“. In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, Juni 2008, S. 692–700. URL: <https://www.aclweb.org/anthology/P08-1079>.

## Literatur II



Joseph Tepperman, David R. Traum und Shrikanth Narayanan. „"Yeah right": Sarcasm Recognition for Spoken Dialogue Systems.“. In: *INTERSPEECH*. ISCA, 2006. URL: <http://dblp.uni-trier.de/db/conf/interspeech/interspeech2006.html#TeppermanTN06>.



Oren Tsur, Dmitry Davidov und Ari Rappoport. „ICWSM - A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews.“. In: *ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*. The AAAI Press, 2010. URL: <https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1495/1851>.

## Appendix - Klassifikation der neuen ungesesehenen Sätzen

Um die Evaluation fairer und relevanter zu gestalten, wurde folgendes gemacht:

- Nicht sarkastische Sätze (1-3 Sternen), die zu den negativen Bewertungen gehören, wurden reduziert.

**Grund:** sarkastische Äußerungen kommen nicht unbedingt in negativen Sätzen vor

- Nur Sätze mit Named-Entity (NE) oder mit der Referenz auf NE wurden ausgewählt (Produkt/Hersteller/Titel/Autor).

**Grund:** es ist wahrscheinlicher, dass die Sätze mit NE relevante Sentimente beinhalten.

## Appendix - Manuelle Annotation

- Das Evaluation Set wurde zufällig in 5 Teile aufgeteilt.
- Jeder Teil besteht aus 36 Sätzen aus dem Evaluation Set und 4 gleichen Anker-Sätzen in 5 Teilen:
  1. "*I love it, although i should have waited 2 more weeks for the touch or the classic.*" - nicht sarkastisch
  2. "*Horrible tripe of a novel, i Lost IQ points reading it.*" - sarkastisch
  3. "*All the features you want - too bad they don't work!*" - sarkastisch
  4. "*Enjoyable light holiday reading.*" - nicht sarkastisch
- Ziele der Anker-Sätzen: kontrollieren, dass Annotatoren die sarkastische Labels nicht zufällig vergeben.
- 15 Annotatoren → jeder Satz wurde von drei verschiedenen Annotatoren annotiert.

## Appendix - Ergebnisse auf ungesehenen Daten

	<b>Precision</b>	<b>Recall</b>	<b>False Pos.</b>	<b>False Neg.</b>	<b>F Score</b>
<b>Baseline</b>	0.5	0.16	0.05	0.44	0.242
<b>SASI</b>	0.766	0.813	0.11	0.12	0.788

# Appendix - Diskussion

## 1. Features drücken deutlich Sarkasmus aus

- *“Silly me, the Kindle and the Sony eBook can't read these protected formats. Great!”*  
→ typische Ausdrücke für Sarkasmus im Muster : *great!, yay!*
- *“i guess i don't think very brilliantly... well... it was ok... but not good to read just for fun... cuz it's not fun...”*  
→ Dreipunkt mit anderen Features: erzeugt besseres Ergebnis
- *“Well you know what happened.  
ALMOST NOTHING HAPPENED!!!”*  
→ übermäßiger Gebrauch von großgeschriebenen Worten/Charakteren drückt Ärger/Emotion aus

# Appendix - Diskussion

## 2 Kontext und Mustergrenzen

- *This book was good until page 2 !*
- *This book was good until page 43 !*

→ Die beiden haben ähnliche Features-Vektoren

→ schwer zu erkennen

- *This book was good until page 2 ! **What** an achievement!*

→ Dieser Satz kann mithilfe vom Kontextwort **What** als **sarkastisch** annotiert werden