Sentimentanalyse für die deutsche Sprache

Melanie Siegel

Darmstädter Ontologenkreis, 15.5.2019

Sentimentanalyse: Worum es geht

Sentimentanalyse als Forschungsgebiet

- **Data Mining** sucht relevante Information in Daten.
- **Text Mining** sucht relevante Information in Sprachdaten.
- **Informationsextraktion** entwickelt Verfahren, um gefundene Information in Wissen zu verwandeln.
- **Opinion Mining/Sentimentanalyse** versucht, Meinungsäußerungen (Information) in Newsgroups und Foren (Sprachdaten) automatisch zu erkennen und zu klassifizieren und damit letztlich Wissen über Meinung zu extrahieren.
- *Zentrale Frage: Welche Verfahren führen zum Erfolg (und zu welchem eigentlich?)*

Was ist eigentlich eine Meinungsäußerung?

Beitrag in einem Forum über Kameras:

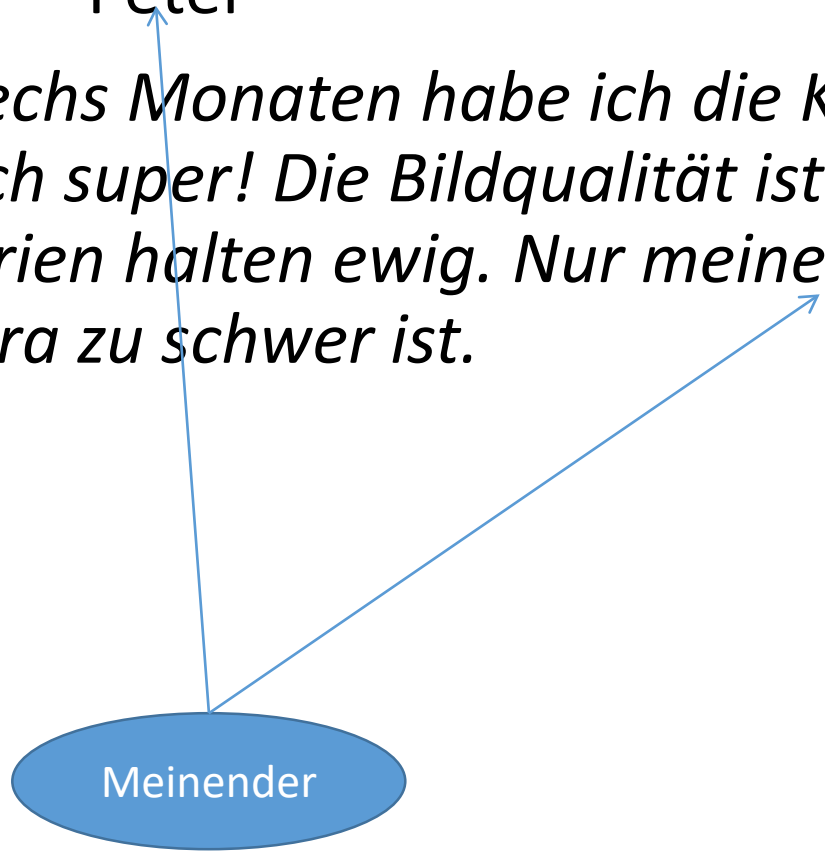
Peter 3.9.2019

Vor sechs Monaten habe ich die Kamera gekauft. Sie ist einfach super! Die Bildqualität ist überragend. Die Batterien halten ewig. Nur meine Frau denkt, dass die Kamera zu schwer ist.

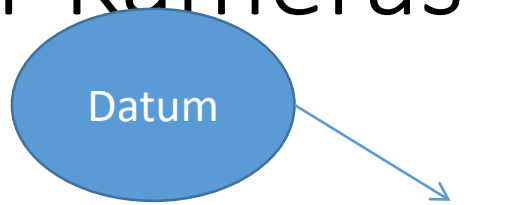
Beitrag in einem Forum über Kameras

Peter 3.9.2019

Vor sechs Monaten habe ich die Kamera gekauft. Sie ist einfach super! Die Bildqualität ist überragend. Die Batterien halten ewig. Nur meine Frau denkt, dass die Kamera zu schwer ist.



Beitrag in einem Forum über Kameras



Peter 3.9.2019

Vor sechs Monaten habe ich die Kamera gekauft. Sie ist einfach super! Die Bildqualität ist überragend. Die Batterien halten ewig. Nur meine Frau denkt, dass die Kamera zu schwer ist.

Beitrag in einem Forum über Kameras

Peter 3.9.2019

Vor sechs Monaten habe ich die Kamera gekauft. Sie ist einfach super! Die Bildqualität ist überragend. Die Batterien halten ewig. Nur meine Frau denkt, dass die Kamera zu schwer ist.

Beitrag in einem Forum über Kameras

Peter 3.9.2019

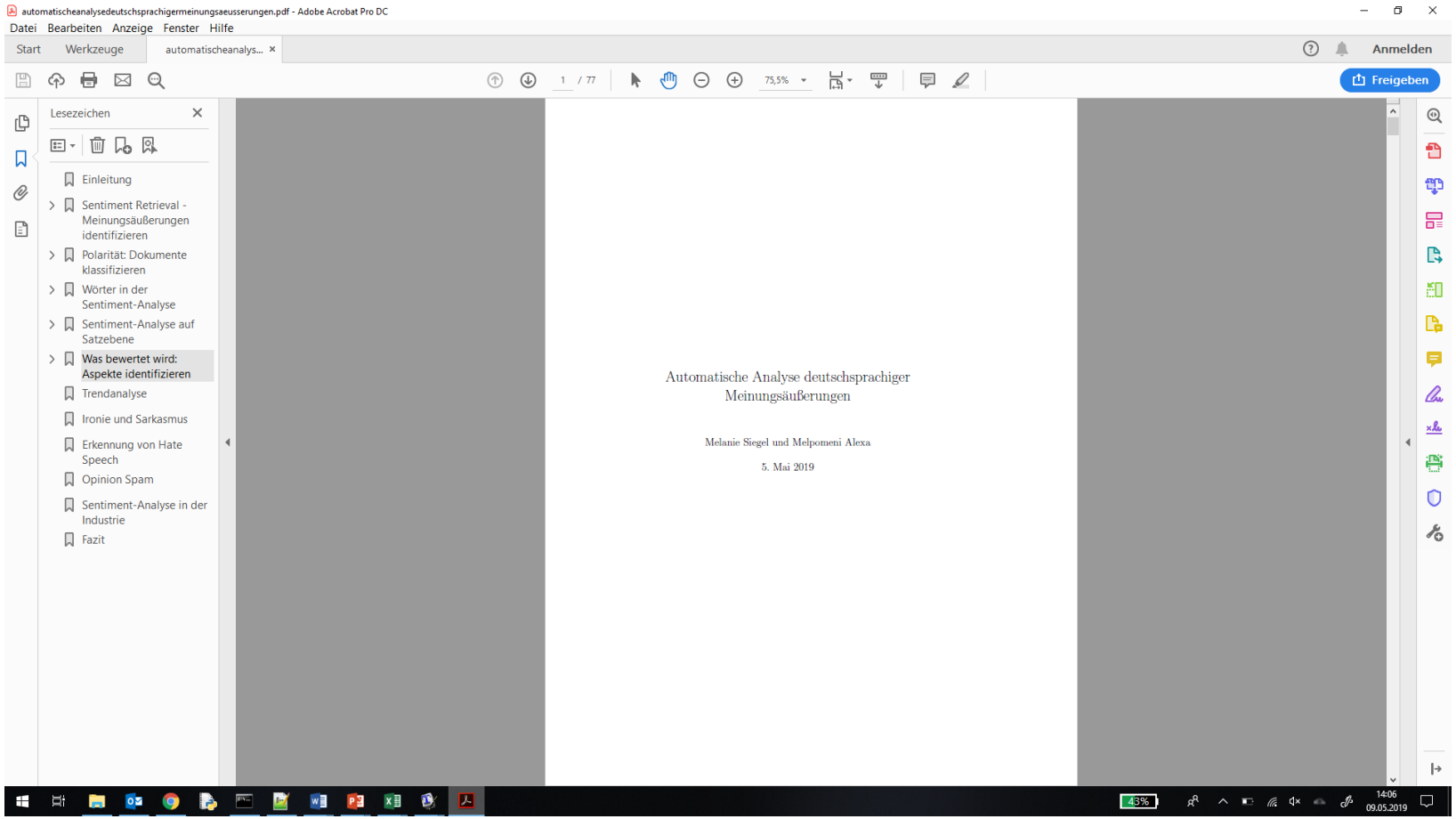
Vor sechs Monaten habe ich die Kamera gekauft. Sie ist einfach super! Die Bildqualität ist überragend. Die Batterien halten ewig. Nur meine Frau denkt, dass die Kamera zu schwer ist.

Elemente einer Meinungsäußerung

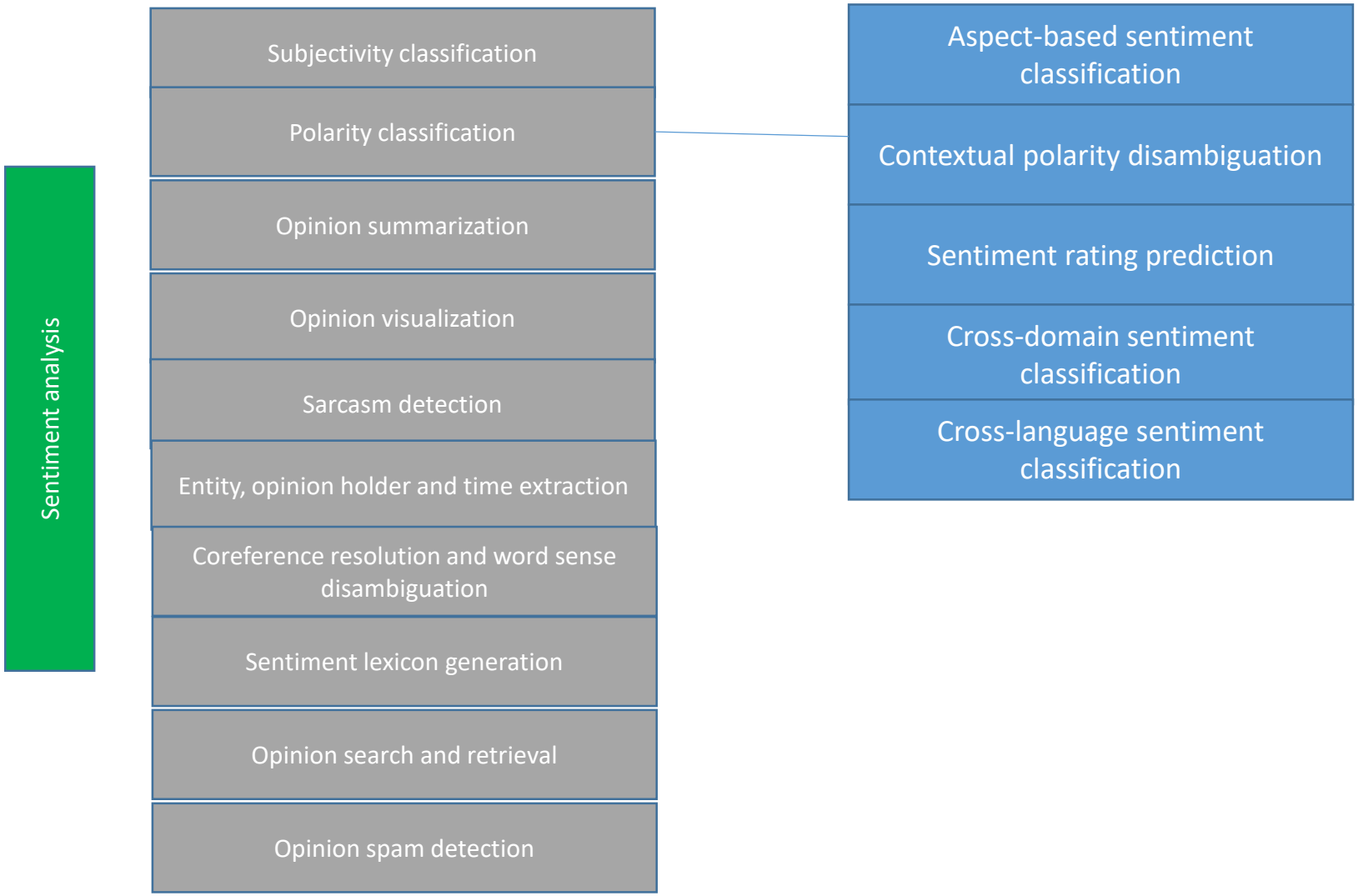
- Ziel (target, goal): g
- Meinung (sentiment) dazu: (g,s)
- Meinender (opinion holder): h
- Datum (date, time): t
- **Meinung (opinion): (g, s, h, t)**

Bing Liu (2012): Sentiment Analysis and Opinion Mining. Morgan & Claypool

Buchprojekt



Sentiment Analysis Tasks*



*Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2016): Challenges of Sentiment Analysis in Social Networks: An Overview. In: Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2016): *Sentiment Analysis in Social Networks*. Morgan Kaufmann, S. 4

Sentiment Retrieval: Meinungsäußerungen identifizieren

- Eine subjektive Aussage ist keine Meinungsäußerung
  - *Ich dachte, sie kommt heute Abend nicht*.
- Eine Äußerung von Emotionen ist nicht immer eine Meinungsäußerung
  - *Ich bin so traurig, dass ich den Film verpasst habe!*

Meinungsäußerungen

- Meinungsäußerungen können positiv oder negativ sein. Nicht immer ist das ohne Kontext sofort klar.
- Es gibt weitere Abstufungen, wie stark positiv oder negativ eine Meinungsäußerung ist.
- Die Meinung einer Person wird geäußert, wobei das nicht immer die Autorin/der Autor des Beitrags ist, sondern auch über Meinungen anderer Personen geschrieben werden kann.
- Meinungsäußerungen beziehen sich nicht immer auf die gesamte Entität (das Produkt, den Film etc.), sondern auch auf einzelne Aspekte (wie die deutsche Übersetzung).

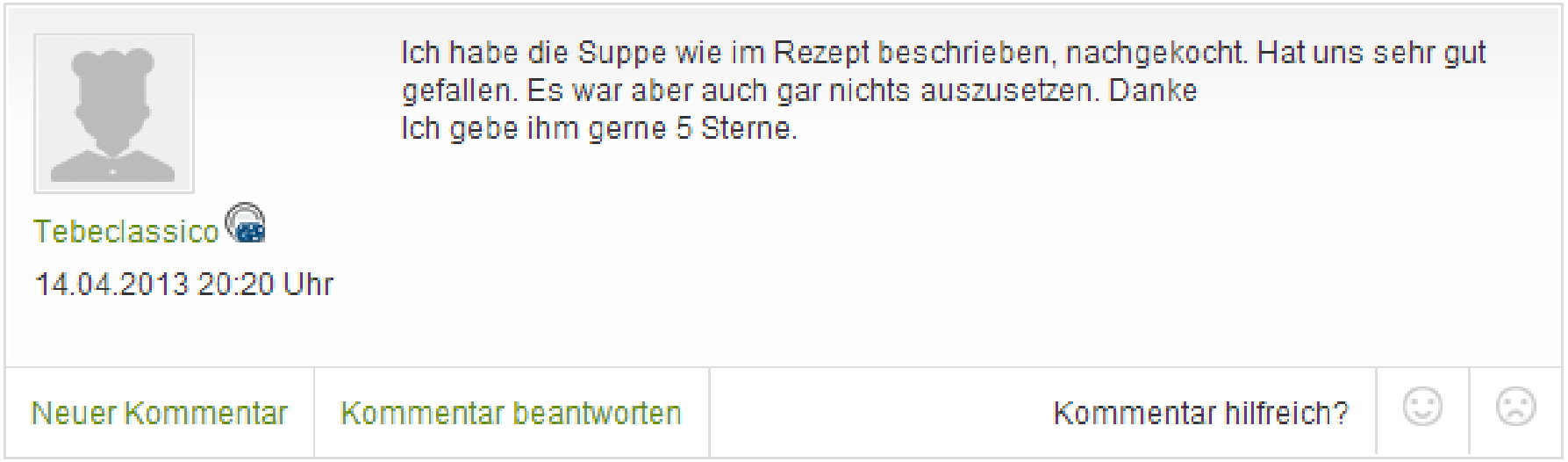
Meinungsäußerungen -2

- Ausdrücke der Meinungsäußerung können kontextunabhängig sein, wie „faszinierend“ oder kontextabhängig, wie „einschlafen“. In einigen Fällen wird zur Interpretation Weltwissen benötigt.
- Negation und Verstärker müssen gesondert behandelt werden.
- Auch Fragen benötigen eine besondere Behandlung.

Methoden

- Meist Abgleich mit Wortlisten (siehe auch NLTK, Textblob, Spacy)
- Übungen: Testen von Textblob auf einigen deutschen Sätzen

Dokumentklassifikation



Ich habe die Suppe wie im Rezept beschrieben, nachgekocht. Hat uns sehr gut gefallen. Es war aber auch gar nichts auszusetzen. Danke
Ich gebe ihm gerne 5 Sterne.

Tebeclassico

14.04.2013 20:20 Uhr

Neuer Kommentar | Kommentar beantworten | Kommentar hilfreich?

aus: chefkoch.de

Dokumentklassifikation

- Klassifikation des Dokuments:
  - Entität *e*
  - Sentiment (Meinung) *s*
- nicht relevant:
  - Aspekte
  - Opinion Holder (Meinender) *h*
  - Zeit?
- also: (e, GENERAL, s, h, t)
  - wobei alle Werte außer s festgelegt sind
  - und es nur ein Quintupel für das Dokument gibt

„Dokument“

Re: DB Bahn Pünktlich zu meiner Reise :) perfekt.*

- (e, GENERAL, s, h, t)
- (Bahn, GENERAL, *positiv*, ?, ?): pünktlich, perfekt, :)

Deutsche Bahn - Eine Horrorgeschichte Darf jetzt erstmal zum nächsten Bahnhof laufen, weil der Zug auf der Strecke stehengeblieben ist

- (Bahn, GENERAL, *negativ*, ?, ?): Horrorgeschichte, stehengeblieben

*aus dem GermEval 2017 Korpus

Relevant sind „Wörter“ – aber was ist das?

Morgens um halb sieben, dem vollen Pendler-RE einen ganzen Wagen

1. Klasse anhängen. Der ist natürlich leer. Fehlleistungen der @DB_Bahn.

→ Tokens!

Erste Idee: Wortlistenvergleich

Aufgabe:

- Stellen Sie eine Wortliste mit positiven Wörtern und eine mit negativen Wörtern auf.
- Schreiben Sie eine Funktion, die einen Eingabesatz tokenisiert. Nutzen Sie dafür entweder die Python-Funktion split() oder einen Tokenizer von Spacy oder Textblob
- Schreiben Sie eine Funktion, die die Tokens im Eingabesatz mit den Wörtern in der Wortliste vergleicht.
- Testen Sie Ihre Funktion mit verschiedenen Sätzen und dokumentieren Sie das Ergebnis

SemEval-2016 Task 4: Sentiment Analysis in Twitter

We replace the two- or three-point scale with a fivepoint scale {HIGHLYPOSITIVE, POSITIVE, NEUTRAL, NEGATIVE, HIGHLYNEGATIVE}, which is now ubiquitous in the corporate world where human ratings are involved: e.g., Amazon, TripAdvisor, and Yelp, all use a five-point scale for rating sentiment towards products, hotels, and restaurants.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). Semeval-2016 task 4: Sentiment analysis in Twitter. In: Proceedings of SemEval-2016, pages 1-18.

Polaritätswerte

- Sentiment-Lexikon mit Polaritätswerten, z.B.:
  - gut: 0.5
  - toll: 1.0
  - schlecht: -0.7
  - doof: -1.0
- Wie bekommt man diese Werte?
  - Von Hand und nach Gefühl eintragen
  - Von mehreren Annotator_innen eintragen lassen und die Ergebnisse vergleichen
  - Wörter in Textkorpora mit Sternchen-Annotationen (wie bei Amazon) automatisch klassifizieren und die Polarität nach der Wahrscheinlichkeit berechnen, mit der sie in positiven oder negativen Bewertungen vorkommen

Aufgaben: Berechnung eines Polarity-Werts

Aufgabe:

- Machen Sie aus den Listen der positiven und der negativen Wörter ein Python-Dictionary und geben Sie jedem Wort einen Polarity-Wert.
- Verändern Sie Ihren Wortlistenvergleich so, dass für jedes Wort der Polarity-Wert ermittelt und im Satz addiert wird, sodass das Ergebnis ein Polarity-Wert für den Satz ist.
- Normalisieren Sie diesen Polarity-Wert, sodass er zwischen -1 und +1 liegt.
- Verändern Sie das Programm zur Evaluation so, dass die Zahlenwerte interpretiert werden (positive Werte: Klassifikation positiv, negative Werte: Klassifikation negativ, 0: Klassifikation neutral).

Dokumentklassifikation mit maschinellem Lernen (Supervised Learning)

- Ziel: Dokument-Klassifikation als neutral, positiv oder negativ
- Daten:
  - Trainingsdaten – Testdaten
  - normalerweise Produkt-Reviews
  - Trainingsdaten begleitet von Sternen ➔ annotierte Daten
  - Liste von klassifizierten Sentiment-Wörtern, z.B. SentiWordNet

Sentiment-Klassifikation im Satz: Supervised Learning, Probabilistisches Sprachmodell

- Trainingskorpus: positive Sätze
  - Das Handy ist super.
  - Ich habe es gekauft und muss sagen, das Handy ist toll.
- Testsatz:
  - Das Handy ist toll.
- Ziel: Wahrscheinlichkeit, dass auch der Testsatz ein positiver Satz ist.

Sentiment-Klassifikation im Satz: Supervised Learning, Probabilistisches Sprachmodell

- Wahrscheinlichkeit des Auftretens von „toll“ im positiven Kontext:

  $p(w_i) = \frac{|wi|}{\sum |wk|}$

  mit

  $|w_i|$: Anzahl der Vorkommen von w_i

  $\sum |wk|$: Summe aller Wortformen im Korpus

Das Handy ist super.
Ich habe es gekauft und muss sagen, das Handy ist toll.

- Also:

  $p(\textit{toll}) = \frac{1}{15}$

Sentiment-Klassifikation im Satz: Supervised Learning, Probabilistisches Sprachmodell

- aber eigentlich: abhängig von umgebenden Wörtern, Klassifikation ganzer Sätze

➔ Bi- und Trigramme!

- Trick: Korpus um Satzanfangmarker ergänzen

  S1 S2 Das Handy ist super.

  S1 S2 Ich habe es gekauft und muss sagen, das Handy ist toll.

Sentiment-Klassifikation im Satz: Supervised Learning, Probabilistisches Sprachmodell

Satzgewicht: G (*Das Handy ist toll*)

= p(S1) · p(S1|S2) · p(das|S1,S2) · p(handy|S2,das) · p(ist|das,handy) · p(toll|handy,ist)

$$= \frac{|S1|}{|S1|+|S2|+\sum|wk|} \cdot \frac{|S1,S2|}{|S1|} \cdot \frac{|S1,S2,das|}{|S1,S2|} \cdot$$

$$\frac{|S2,das,handy|}{|S2,das|} \cdot \frac{|das,handy,ist|}{|das,handy|} \cdot \frac{|handy,ist,toll|}{|handy,ist|}$$

Sentiment-Klassifikation im Satz: Supervised Learning, Probabilistisches Sprachmodell

$$\frac{|S1|}{|S1|+|S2|+\sum|wk|} \cdot \frac{|S1,S2|}{|S1|} \cdot \frac{|S1,S2,das|}{|S1,S2|} \cdot$$

$$\frac{|S2,das,handy|}{|S2,das|} \cdot \frac{|das,handy,ist|}{|das,handy|} \cdot \frac{|handy,ist,toll|}{|handy,ist|}$$

$$= \frac{2}{19} \cdot \frac{2}{2} \cdot \frac{1}{2} \cdot \frac{1}{1} \cdot \frac{2}{2} \cdot \frac{1}{2}$$

$$= \frac{1}{38}$$

S1 S2 Das Handy ist super.
S1 S2 Ich habe es gekauft und muss sagen, das Handy ist toll.

Supervised Learning: Features

- Um auf annotierten Daten ML anwenden zu können, muss man zunächst entscheiden, was die Features sind, die dem ML gefüttert werden
- Z.B. im Germeval 2017-Korpus:

```
<Document id="http://twitter.com/DOMKEYTV/statuses/73330230607937945 6">
    <Opinions>
      <Opinion category="Zugfahrt#Pünktlichkeit"
          from="46" to="55" target="pünktlich" polarity="negative"/>
    </Opinions>
    <relevance>true</relevance>
    <sentiment>negative</sentiment>
    <text>Wäre ja mal ein Wunder wenn die deutsche Bahn pünktlich fährt</text>
</Document>
```

- Welche Information ist für die Dokumentklassifikation relevant?

Lernverfahren für Dokumentklassifikation: Supervised Learning

- Basisinformation (Features):
  - Häufigkeit von Wörtern
  - syntaktische Kategorien, POS (meist Adjektive)
  - Lemmata
  - Sentiment-Wörter (gut, super, schlecht,…)
  - Negationen, “Sentiment Shifters” (nicht, don‘t,…)
  - syntaktische Dependenzen

Aufgabe: Datensatz aufstellen

- Schreiben Sie ein Programm, das aus dem GermEval-2017-Datensatz eine Tabelle (csv) der relevanten Dokumente generiert, mit folgenden Spalten:
  - Text
  - Addierte Polarität für Wörter aus dem Sentiment-Wörterbuch
  - Zahl der Negationen
  - Sentiment-Klassifikation aus dem Datensatz

Aufgabe: Daten normalisieren/standardisieren

- Normalisieren oder standardisieren Sie die Features im Datensatz:
  - Installieren Sie mit pip: scipy, sklearn ***oder***
  - definieren Sie die Normalisierung/Standardisierung selbst
  - Im ersten Schritt berechnen Sie die Werte für das Dokument:
    - Minimum
    - Maximum
    - Mittelwert
    - Standardabweichung
  - Im zweiten Schritt berechnen Sie mit dieser Basis jeden Wert neu.

- Das ist jetzt die Basis für das maschinelle Lernen!

Weiter…

- Datensatz aufteilen in Trainings- und Testdaten
- Klassifikator auf den Trainingsdaten trainieren
- Modell abspeichern

Aufgabe: Modell auf einen Satz anwenden

- Implementieren Sie eine Funktion, die für einen Satz das Sentiment berechnet
  - Eingabe für die Funktion ist ein Satz
  - Dann werden die Polarität, Negationen und Verstärker berechnet
  - Diese Information wird als Array gespeichert
  - Dann das Modell laden
  - Und mit model.predict für das Array eine Voraussage machen

```
>>> pred_senti_sentence("Diese Bahn ist saudoof")
'negative'
```

Wörter in der Sentimentanalyse

- Zentrale Rolle der Wörter
- Ziel: umfangreiche Wortlisten, domänenspezifische Wortlisten
- Methoden:
  - Einbindung eines existierenden Sentiment-Lexikons
  - Gewinnung von Sentimentwörtern aus WordNet
  - Gewinnung domänenabhängiger Sentimentwörter aus annotierten Korpora
  - Gewinnung domänenabhängiger Sentimentwörter aus nicht annotierten Korpora

Einbindung eines existierenden Sentiment-Wörterbuchs

- SePL (Sentiment Phrase List)
  - Universität Hof
  - http://www.opinion-mining.org/SePL-Sentiment-Phrase-List
  - über 14.000 Einträge im csv-Format, die aus Produktbewertungen mit ihrer Sternewertung automatisch erzeugt und zum Teil von Hand korrigiert wurden
- Polarity Lexicon
  - Universität Zürich
  - Basis: literarische Texte
  - ca. 8400 Einträge
- Multi-Domain Sentiment Lexicon for German
  - Hochschule Darmstadt, studentisches Projekt von Kerstin Diwisch
  - Kombination der lexikalischen Daten aus drei verschiedenen Wortlisten
  - ca. 2900 Einträge im XML-Format

Gewinnung von Sentimentwörtern aus WordNet

- Synonyme von wenigen Adjektiven im WordNet finden und aufnehmen
- Synonyme zu „gut“ in OdeNet:
  - *['1a', 'O. K.', 'Seele von Mensch', 'abgemacht', 'akzeptiert', 'alles klar', 'alles paletti', 'angenehm', 'charmant', "d'accord", 'da sage ich nicht nein', 'das ist ein Wort', 'dein Wille geschehe', 'dienlich', 'eins a', 'einverstanden', 'erbaulich', 'erfreulich', 'ergötzlich', 'erhebend', 'erquicklich', 'ersprießlich', 'es geschehe nach deinen Worten', 'es sei', 'fein', 'fruchtbar', 'förderlich', 'gebongt', 'gedeihlich', 'gefreut', 'geht in Ordnung', 'geht klar', 'gemacht', 'genehmigt', 'gewinnbringend', 'glücklich', 'gutmütig', 'günstig', 'gütig', 'herzensgut', 'herzerfrischend', 'herzerquicklich', 'hilfreich', 'ich nehme dich beim Wort', 'ist recht', 'lohnend', 'machen wir', 'manierlich', 'menschlich', 'nutzbringend', 'nutzwertig', 'nützlich', 'o. k.', 'okay', 'okey-dokey', 'opportun', 'pläsierlich', 'positiv', 'roger', 'sachdienlich', 'schon überredet', 'schön', "so machen wir's", 'so sei es', 'sympathisch', 'tadellos', 'trefflich', 'von Nutzen', 'von Vorteil', 'von guter Qualität', 'vorteilhaft', 'warum nicht', 'wertvoll', 'wohl', 'wohltuend', 'zuträglich']*

Extraktion aus annotierten Daten: Vorgehensweise

- Extraktion aller negativen, positiven und neutralen Sätze in getrennte Dateien
  - Nutzung des csv-Readers für tsv-Datei oder ElementTree XML API für XML-Datei
- Tokenisierung, Lemmatisierung, POS-Tagging (Textblob oder Spacy)
- Extraktion aller Adjektive aus den Sätzen in getrennte Listen (negativ, positiv, neutral)
- Subtraktion der positiven und neutralen Adjektive von den negativen ergibt die Adjektive, die nur in negativen Sätzen vorkommen.

Unsupervised Learning für die Erweiterung von Wortlisten: PMI

- Sentiment Orientation (SO) einer extrahierten Phrase:
- Pointwise Mutual Information Measure (PMI):

$$PMI(term_1, term_2) = \log_2\left(\frac{\Pr(term_1 \wedge term_2)}{\Pr(term_1)\Pr(term_2)}\right).$$

- $Pr(term_1 \Lambda term_2)$: Wahrscheinlichkeit, dass $term_1$und $term_2$ zusammen auftreten
- $Pr(term_1)Pr(term_2)$: Wahrscheinlichkeit, dass $term_1$und $term_2$ unabhängig voneinander auftreten
- PMI (gut, super): 8,61 – PMI (gut, kalt): 1,22

Andere Vorgehensweise: Word Embeddings

- Idee: Es wird geschaut, welche Wörter gemeinsam auftreten und dann werden ähnliche Kontexte identifiziert
- Jedes Inhaltswort im gesamten Korpus wird lemmatisiert und bekommt eine Nummer.
- Diese Nummer ist dann eine Position im Vektor.
- Dann wird für jedes Wort eine Umgebung definiert, z.B. 5 Wörter rechts und links davon.
- Diese Wörter haben dann einen Vektor, bei dem an der Position der Wörter rechts und links eine 1 steht.

Andere Vorgehensweise: Word Embeddings

- Ganz kleines Beispiel:
  - Das Handy ist super.
  - Ich habe es gekauft und muss sagen, das Handy ist toll.
- Lemmatisierung, Stoppwörter raus und Nummerierung:
  - Handy: 1
  - super: 2
  - kaufen: 3
  - sagen: 4
  - toll: 5
- Vektor für „super“:
  - [0,1,0,0,0]
- Vektor für „toll“:
  - [0,0,0,0,1]

Word Embeddings

- Jetzt schaut man sich die Umgebung an (2 rechts, 2 links)
- Vektor für „super“ im ersten Satz:
  - [1,1,0,0,0]
- Vektor für „toll“ im zweiten Satz:
  - [1,0,0,0,1]
- Wenn wir ganz viele Sätze mit „super“ haben, dann entstehen Wahrscheinlichkeiten an den Positionen im Vektor für das Wort.
- Die Vektoren werden dann mit ML-Verfahren verglichen

Aufgaben

- Reduzieren Sie Ihr Sentiment-Wörterbuch so, dass nur noch Lemmata drinstehen
- Verändern Sie den Wortlistenvergleich, indem Sie die kleingeschriebenen Lemmata der Wörter vergleichen
  - Lemmatisierung z.B. mit Spacy oder Textblob
- Binden Sie das Sentiment-Wörterbuch sentiment_words.py ein
- Fügen Sie die Synonyme zu „gut“ und „schlecht“ aus OdeNet in das Wörterbuch ein.
- Berechnen Sie die PMI-Werte für einige Ihrer Adjektive mit „gut“ oder „schlecht“

Weitere Aufgabe

- Nehmen Sie den in Moodle zur Verfügung gestellten Korpus der GermEval 2017 zur Hand und extrahieren Sie daraus negative und positive Wörter durch einen TF-IDF-Vergleich. Erweitern Sie damit Ihr Lexikon.
- Word Embeddings: Aufgrund der großen Datenmengen, die benötigt werden und der riesigen Vektoren, die dabei entstehen, können realistische Experimente nur mit Servern durchgeführt werden, die eine sehr große Rechenleistung haben. Für ein kleines Experiment nehmen Sie die Installation von Nathan Rooy und passen Sie sie für unser Spielzeugbeispiel von oben an: https://nathanrooy.github.io/posts/2018-03-22/word2vec-from-scratch-with-python-and-numpy/

Sentimentanalyse auf Satzebene

- Gradpartikeln
  - Ich finde das Produkt sehr schlecht.
  - Das ist ein bisschen ungünstig.
- Negationen
  - Sentiment-Wörter, die negiert die gegenteilige Polarität haben:
    - gut – nicht gut
    - schlecht – nicht schlecht
  - Sentiment-Wörter, die negiert eine abgeschwächte Polarität haben:
    - super – nicht super
    - toll – nicht so toll
    - schrecklich – nicht schrecklich

Negation ohne Grammatikanalyse: Potts

- Ein _neg an jedes Wort zwischen der Negation und dem nächsten Satzzeichen hängen:
  - *Niemandem macht das Spaß.*
    - *Niemandem*
    - *macht_neg*
    - *das_neg*
    - *Spaß_neg*

http://sentiment.christopherpotts.net/lingstruc.html

Negation ohne Grammatikanalyse: Aufgabe

- Verändern Sie Ihr Programm zur Sentimentanalyse so, dass beim Auftreten einer Negation die Polarität des nächsten Sentimentworts umgedreht wird.
- Finden Sie eine Liste deutscher Negationen im Internet, die Sie verwenden können?
- Fügen Sie dann noch Verstärker wie „sehr“ oder „total“ hinzu, die die Polarität erhöhen.
- Testen Sie Ihr Programm mit dem GermEval-Korpus. Was funktioniert und was funktioniert nicht?

Grammatikanalyse mit SpaCy

Text	POS	Dependency	Head	Lemma
Das	DET	nk	Handy	Das
Handy	NOUN	sb	ist	Handy
ist	AUX	ROOT	ist	sein
nicht	PART	ng	gut	nicht
gut	ADJ	pd	ist	gut

nk: noun kernel element
sb: subject
ng: negation
pd: predicate

https://spacy.io/api/annotation#section-dependency-parsing

Negation mit Dependenzgrammatik: Aufgabe

- Ändern Sie Ihr Programm so, dass eine Dependenzanalyse mit SpaCy gemacht wird und die Negation die Polarität seiner Head-Konstituente umdreht.
- Machen Sie dasselbe für Verstärker.
- Vergleichen Sie das Ergebnis mit der Negation ohne Grammatik auf dem Germeval-Korpus.

Was bewertet wird: Aspekte identifizieren

- Beispiel aus GermEval 2017:
  - alle so "Yeah, Streik beendet“ Bahn so "okay, dafür werden dann natürlich die Tickets teurer“ Alle so "Können wir wieder Streik haben?“
  - <tab>relevant<tab>neutral <tab>
  - Ticketkauf# Haupt:negativ
  - Allgemein# Haupt:positiv
- Aufgabe: Identifikation von
  - Ticketkauf# Haupt:negativ
  - Allgemein# Haupt:positiv

Taxonomie der Aspekte einer Domäne, z.B. GermEval 2017

- Allgemein
- Atmosphäre
  - Lautstärke
  - Beleuchtung
  - Fahrgefühl
  - Temperatur
  - Sauberkeit allgemein
  - Geruch
  - Sonstiges
- Connectivity
  - WLAN/Internet
  - Telefonie/Handyempfang
  - ICE Portal
  - Sonstiges
- Design
- Gastronomisches Angebot
  - Verfügbarkeit Bordbistro/- restaurant
  - Verfügbarkeit angebotener Produkte
  - Vielfalt/Auswahl
  - Preise
  - Gastronomiebetreuung
  - Sonstiges
- Informationen
- usw.

Zuordnung von Phrasen zu Aspekten

- In den GermEval-Daten als „Targets“ annotiert
- Wenn Targets nicht annotiert sind, dann Wörter aus den Texten zu den Aspekten extrahieren
- Erweiterung z.B. mit WordNet
- Problem Mehrwortlexeme:
  - RT @Tryli: Wie schön es ist wenn man sich nach nem nervigen Arbeitstag auch noch über die Bahn ärgern muss.
- Problem Anaphern:
  - Ich habe mir letzte Woche einen neuen Tisch gekauft. Er sieht sehr schön aus.
- Problem implizite Aspekte:
  - Diese Kamera passt nicht gut in eine Tasche (Größe)

Aufgaben

- Überarbeiten Sie die Target-Listen. Sehen Sie sich dabei besonders die Mehrwortlexeme an.
- Schreiben Sie eine Funktion, die - wenn sie im Text ein Target findet – den Text mit dem dazugehörigen Aspekt markiert.

Sentiment-Klassifikation des Aspekts

- Annahme der meisten Algorithmen:
  - Nur ein Aspekt und ein Sentiment pro Satz / Tweet / Review
- Aber:
  - Heute mal mit der @DB Bahn zur Arbeit. Deutlich entspannter, aber doppelt so lange unterwegs
  - Zugfahrt#Haupt: positiv, Zugfahrt#Fahrtzeit und Schnelligkeit: negativ
  - Teilung bei Wörtern wie „aber“, „jedoch“ usw.
- Implizite Aspekte:
  - stinkt: Atmosphäre#Geruch: negativ
- Negative Aspekte:
  - Z.B.: Sonstige_Unregelmäßigkeiten

Sentiment-Klassifikation des Aspekts

- Grundlegende Ansätze:
  - Supervised Learning
  - Topic Modelling
  - Lexikon-basiert
  - tiefe Analyse (Parsing, Dependenzanalyse)
- Problem:
  - Skopus des Sentiment-Ausdrucks finden:
  - *Jedoch sind die Grafiken und Tabellen sehr gut und dienen sicherlich dazu, das Thema besser zu verstehen.*
  - *sehr gut (Grafiken und Tabellen)*
  - *nicht das Thema!*

Skopus des Sentiment-Ausdrucks finden: Dependenzanalyse

- Spacy: Die Tabellen sind wirklich gut.

Die	DET	nk	Tabellen	Die
Tabelle	NOUN	sb	sind	Tabellen
sein	AUX	ROOT	sind	sind
wirklich	ADJ	mo	gut	wirklich
gut	ADJ	pd	sind	gut
.	PUNCT	punct	sind	.

Aufgaben

- Nutzen Sie den Dependenzparser von Spacy. Wenn Sie eine Meinungsäußerung gefunden haben, geben Sie auch den Aspekt an.
- Nutzen (und ggf. erweitern) Sie OdeNet, um Synonyme zu relevanten Aspekten zu finden

Weitere Themen

- Trendanalyse
- Ironie und Sarkasmus
- Erkennung von Hate Speech
- Opinion Spam
- Sentimentanalyse in der Industrie