Präsentation des Papers „Thumbs up? Sentiment Classification using Machine Learning Techniques “ (Bo Pang, Lillian Lee, und Shivakumar Vaithyanathan)

Leander Girrbach

14. Oktober 2019

Inhalt

Informationen zum Paper

Titel: Thumbs up? Sentiment Classification using Machine Learning Techniques

Autoren: Bo Pang, Lillian Lee, und Shivakumar Vaithyanathan

Jahr: 2002 (Auf der Conference on Empirical Methods in Natural Language Processing)

Motivation

barrier between you and the screen.

Jul 26, 2017 | Rating: 4.5/5 | Full Review...

Sandra Hall
Sydney Morning Herald
★ Top Critic

characters, come as a relief from the chromatic tedium.

Jul 25, 2017 | Full Review...

Christian Lorentzen
The New Republic
★ Top Critic

With a dedication to excellent "show don't tell" presentation, stunning cinematography and a downright scary soundtrack and score, Dunkirk succeeds at putting viewers in the thick of an unsung but no less dark time of the Second World War.

Oct 10, 2019 | Rating: 9/10 | Full Review...

Nick Monahan
Cultured Vultures

A lofty divorce from Nolan's usual genre oriented flavor, this plays best as a multi-perspective experience of survival and rescue rather than a character driven ensemble as suggested by its cast.

Oct 10, 2019 | Rating: 3.5/5 | Full Review...

Nicholas Bell
IONCINEMA.com

Nolan has managed to make a masterpiece in a genre crowded with masterpieces, one that will likely be held up as one of the best ever about World War II.

Sep 29, 2019 | Rating: 4.5/5 | Full Review...

David Harris
Spectrum Culture

[Dunkirk] is massive and impressive.

Sep 18, 2019 | Full Review...

Linda and Al Lerner
Movies and Shakers

Nolan's knack for 'sculpting in time' reaches a new apex. Dunkirk is to Nolan as 'David' is to Michelangelo. It's his masterpiece.

From the artful cinematography to Hans Zimmer's disorienting score, and the non-linear narrative to the dialogue-less emotion, Dunkirk is a practice in the

Untersuchte Aufgabe

Experimente

- Sentimentklassifikation
- durch Verfahren überwachten maschinellen Lernens
- durch von Menschen erstellte Wortlisten

Ziel

- Schwierigkeiten des Problems analysieren

Was ist mit Sentimentklassifikation gemeint?

- hier: Einschränkung auf Filmkritiken
- entscheiden, ob eine gegebene Kritik positiv oder negativ ist („Orientierung“)
  $\Rightarrow$ binäre Klassifikationsaufgabe
- Ähnlich zu Themenklassifikation:
  $\Rightarrow$ Pang et al. (2002) vergleichen ihre Ergebnisse damit

Daten

- Filmkritiken
- Quelle: IMDb
- Datenset online verfügbar
- Datenset: 700 positive und 700 negative Kritiken
  $\Rightarrow$ balanciertes Datenset

Warum Filmkritiken?

- sind online verfügbar
- Sternebewertung: automatische Ermittlung möglich, ob die Kritik positiv oder negativ ist
- für Filmkritiken angewandte Methoden lassen sich auf andere Domänen anwenden
- Turney (2002) erzielte niedrigstes Ergebnis für Filmkritiken

Wie werden Kritiken fürs Datenset ausgewählt?

- automatische Ermittlung der Orientierung möglich
- max. 20 Kritiken pro Autor
- zufallsbasierte Auswahl von 700 positiven und 700 negativen Kritiken
- Annahme: alle Kritiken sind auf Englisch

Wie werden Labels für Kritiken ermittelt?

- Bewertungsschemata unterschiedlich
- in verwendeten Kritiken: Bewertung angegeben durch Sterne oder Zahl
- Wenn keine Höchstwertung angegeben:
  $\Rightarrow$ 4 Sterne als Höchsbewertung angenommen
- Bei Zahlen: Nur wenn Höchsbewertung angegeben

Höchsbewertung

4 Sterne:	• 3 Sterne und mehr: positiv • 1 Stern und weniger: negativ
5 Sterne:	• 4 Sterne und mehr: positiv • 2 Sterne und weniger: negativ

1. Überblick über die untersuchte Aufgabe
2. Daten
3. Klassifikation anhand von Wortlisten
4. Klassifikation durch Machine Learning
   - Features
   - Verfahren maschinellen Lernens
   - Ergebnisse
5. Zusammenfassung

Verfahren

Hypothese

- Orientierung einer Kritik lässt sich anhand bestimmter Wörter erkennen

Voraussetzung

- Wortlisten
- eine Liste für positive Wörter
- eine Liste für negative Wörter

Idee

- Menschen erstellen Wortlisten
- ergibt Baseline

Klassifikation anhand von Wortlisten

```
function classifyDocument(positiveWords, negativeWords, document)
    int negativeCounter ← 0;
    int positiveCounter ← 0;
    for word ∈ document do
        if word ∈ positiveWords then
            positiveCounter ← positiveCounter + 1;
        else if word ∈ negativeWords then
            negativeCounter ← negativeCounter + 1;
    if positiveCounter > negativeCounter then return positive;
    else if positiveCounter < negativeCounter then return negative;
    else return Wert, der die Accuracy der Baseline maximiert
```

Wie sehen die Wortlisten aus?

Durch Studenten erstellte Wortlisten:

	Wörter
Student 1	positiv: *dazzling*, *brilliant*, *phenomenal*, *excellent*, *fantastic* negativ: *suck*, *terrible*, *awful*, *unwatchable*, *hideous*
Student 2	positiv: *gripping*, *mesmerizing*, *riveting*, *spectacular*, *cool*, *awesome*,*thrilling*, *badass*, *excellent*, *moving*, *exciting* negativ: *bad*, *cliched*, *sucks*, *boring*, *stupid*, *slow*

Wie sehen die Wortlisten aus?

- zusätzlich: von Autoren erstelltes Listenpaar
- mithilfe von Statistiken ausgewählt
- 7 positive und 7 negative Wörter
- Statistiken anhand des kompletten Datensets

positiv:	*love*, *wonderful*, *best*, *great*, *superb*, *still*, *beautiful*
negativ:	*bad*, *worst*, *stupid*, *waste*, *boring*, *?*, *!*

Ergebnisse

Auf komplettem Datenset:

	Accuracy	Gleichstand
Student 1	58%	75%
Student 2	64%	39%
Autoren + Statistiken	69%	16%

Bei Gleichstand

- Accuracy der Baselines soll maximiert werden

Folgerung der Autoren

- Nutzen von Statistik: höhere Accuracy + weniger Gleichstand
  $\Rightarrow$ Einsatz statistischer Verfahren sinnvoll

1. Überblick über die untersuchte Aufgabe
2. Daten
3. Klassifikation anhand von Wortlisten
4. Klassifikation durch Machine Learning
   - Features
   - Verfahren maschinellen Lernens
   - Ergebnisse
5. Zusammenfassung

Features

- Dokumente müssen als Vektoren dargestellt werden
- Jedes Feature misst eine bestimmte Eigenschaft
- Features, die Häufigkeiten angeben
- Features, die Anwesenheit angeben (binär)
- Dokument d und Features $f_1, \ldots f_m$:

$$\vec{d} = (n_1(d), n_2(d), \ldots, n_m(d))$$

- n_i: Funktion, die Feature f_i misst

Bag-of-Words: Unigramme

- Features, die Häufigkeit oder Anwesenheit von einzelnen Wörtern angeben
- keine Kontextinformationen (Reihenfolge, . . .)
- keine Stoppwörterfilterung oder Stemming
- Tokens müssen mindestens 4-mal vorkommen

Beispiel: Unigramme

Häufigkeit

Beschreibung des Features:	Anzahl des Tokens „rose“
Beispieldokument:	„A rose is a rose is a rose“
Wert des Features:	3

Anwesenheit

Beschreibung des Features:	Vorhandensein des Tokens „rose“
Beispieldokument:	„A rose is a rose is a rose“
Wert des Features:	1

Unigramme: Negationsbehandlung

- Negation ändert die Bedeutung von Wörtern
- Potentiell wichtig für Sentimentklassifikation:

Beispiel 1. „The movie is very good.“

Beispiel 2. „The movie is not very good.“

- markiere Wörter zwischen Negationswort und Punktuationssymbol:

Beispiel 3. „. . . not unique but effective .“
⇒ „not `not_`unique `not_`but `not_`effective .“

Bigramme

- Beispiel: „The movie is good“
- Bigramme: „The movie“, „movie is“, und „is good“
- keine Negationsbehandlung
- gleich viele Bigramm-Features wie Unigramm-Features

POS-Tags

- markiert jedes Wort mit Wortart

Beispiel

- DT NN VBZ JJ .
  The movie is good .

wird zu:

- wird zu: „`DT_`The `NN_`movie `VBZ_`is `JJ_`good `._`.“

- hilft, folgende Sätze zu unterscheiden („love“):
  1. „I love (VBP) this movie “
  2. „This is a love (NN) story “

Adjektive + Wortposition

Adjektive

- nur Adjektiv-Unigramme als Features verwenden
- Sentimentklassifikation in Turney (2002) und Hatzivassiloglou and Wiebe (2000) verwendet Adjektive: gute Ergebnisse
- alternativ: Häufigste Unigramme (gleich viele wie Adjektive)

Wortposition

- markiere Wort mit Position im Dokument
- erstes Viertel, mittlere Hälfte, letztes Viertel
- begründet durch typische Struktur von Kritiken

Verfahren

Naïve Bayes (NB)

Besonderheit: Add-One Smoothing (bei Häufigkeit)

Maximum Entropy classification (ME)

Besonderheit: Nur binäre Features

Support Vector Machine (SVM)

Verwendete Software: $\text{SVM}^{\text{light}}$

Ergebnisse: Unigramme

Features	NB	ME	SVM
Unigramme (Häufigkeit)	79.0	n/a	72.8
Unigramme (Anwesenheit)	81.5	80.4	82.9

- Metrik: Accuracy
- 3-fache Kreuzvalidierung

Schlussfolgerung aus den Ergebnissen

- Features, die Anwesenheit angeben, sind besser
- folgende Features (Bigramme, . . .) auch binär

Ergebnisse: alle Features + Kombinationen

Features	SVM
Unigramme (Häufigkeit)	72.8
Unigramme	82.9
Unigramme und Bigramme	82.7
Bigramme	77.1
Unigramme und POS-Tags	81.9
Adjektive	75.1
2633 häufigste Unigramme	81.4
Unigramme und Position	81.6

Schlussfolgerungen

- andere Features (als Unigramme) nicht besser

Ergebnisse: Unigramme + Baselines

	Accuracy	Gleichstand
Student 1	58%	75%
Student 2	64%	39%
Autoren + Statistiken	69%	16%
	SVM	
Unigramme (Häufigkeit)	72.8%	
Unigramme (Anwesenheit)	82.9%	

Schlussfolgerungen

- Machine Learning besser als Baselines
- andere Textklassifikationsaufgaben erzielen bessere Ergebnisse z.B. Nigam et al. (1999)

Interpretationen der Autoren

- Sentimentklassifikation schwerer als Themenklassifikation (schlechtere Ergebnisse)

Interpretationen der Autoren

- Sentimentklassifikation schwerer als Themenklassifikation (schlechtere Ergebnisse)
- Häufigkeit vs. Anwesenheit: Unterschied zwischen Sentiment- und Themenklassifikation
- Unigramme mit Anwesenheitsinformation sind am besten
- Spekulation: Sentiment wird durch bestimmte Worte übermittelt

Interpretationen der Autoren

- Sentimentklassifikation schwerer als Themenklassifikation (schlechtere Ergebnisse)
- Häufigkeit vs. Anwesenheit: Unterschied zwischen Sentiment- und Themenklassifikation
- Unigramme mit Anwesenheitsinformation sind am besten
- Spekulation: Sentiment wird durch bestimmte Worte übermittelt
- Adjektive nicht nützlich
  $\Rightarrow$ Feature-Selection auf Unigrammen
- Bigramme nicht für Kontextinformationen geeignet
- andere Positionsschemata könnten besser sein

Fehleranalyse

Was ist das Problem?

„This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up“

Fehleranalyse

Was ist das Problem?

„This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up“

⇒ enttäuschte Erwartung

- kann von Bag-of-Words nicht erkannt werden
- bessere pragmatische Analyse notwendig

1. Überblick über die untersuchte Aufgabe
2. Daten
3. Klassifikation anhand von Wortlisten
4. Klassifikation durch Machine Learning
   - Features
   - Verfahren maschinellen Lernens
   - Ergebnisse
5. Zusammenfassung

Zusammenfassung

- Sentimentklassifikation bei Filmkritiken
- recht gutes Ergebnis durch Machine Learning
- Unigram-Bag-of-Words-Features am nützlichsten
- Anwesenheitsinformation nützlicher als Häufigkeit
- 'enttäuschte Erwartung' als Problem bei Sentimentklassifikation erkannt
- aber: keine klare Erkenntnis, wodurch Sentiment vermittelt ist

Vielen Dank!

Literatur I

Hatzivassiloglou, V. and Wiebe, J. M. (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 299–305, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nigam, K., Lafferty, J., and McCallum, A. (1999). Using maximum entropy for text classification. In *Proc. of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Literatur II

Turney, P. D. (2002). Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.