

# Big Data and Automated Content Analysis

## Week 4 – Monday

### »Sentiment Analysis«

Damian Trilling

d.c.trilling@uva.nl  
@damian0604  
www.damiantrilling.net

Afdeling Communicatiewetenschap  
Universiteit van Amsterdam

26 February 2017

# Today

## ① Different types of analysis

What can we do?

Systematizing analytical approaches

## ② Data analysis 1: Sentiment analysis

What is it?

Bag-of-words approaches

Advanced approaches

A sentiment analysis tailored to your needs!

Packages for sentiment analysis

A recipe

Machine Learning as alternative

## ③ Take-home message, next meetings, & exam

# What we already can do

with regard to data collection:

- query a (JSON-based) API (GoogleBooks, Twitter)
- handle CSV files
- handle JSON files

# What we already can do

## with regard to data collection:

- query a (JSON-based) API (GoogleBooks, Twitter)
- handle CSV files
- handle JSON files

## with regard to analysis:

Not much. We counted some frequencies and calculated some averages.

## Data analysis: Overview

## What can we do?

## What can we do?

**What do you think? What are interesting methods to analyze large data sets (like, e.g., social media data? What questions can they answer?)**

- sentiment analysis
- automated coding with regular expressions
- natural language processing
- supervised and unsupervised machine learning
- network analysis

# What else can we do?

Or ideally...

...a combination of these techniques.



## Systematizing analytical approaches

# Systematizing analytical approaches

Taking the example of Twitter:

## Analyzing the *structure*

- Number of Tweets over time
- singleton/retweet ratio
- Distribution of number of Tweets per user
- Interaction networks

Bruns, A., & Stieglitz, S. (2013). Toward more systematic Twitter analysis: metrics for tweeting activities. *International Journal of Social Research Methodology*. doi:10.1080/13645579.2012.756095

# Systematizing analytical approaches

Taking the example of Twitter:

## Analyzing the *structure*

- Number of Tweets over time
- singleton/retweet ratio
- Distribution of number of Tweets per user
- Interaction networks

⇒ **Focus on the amount of content and on the question who interacts with whom, not on what is said**

Bruns, A., & Stieglitz, S. (2013). Toward more systematic Twitter analysis: metrics for tweeting activities. *International Journal of Social Research Methodology*. doi:10.1080/13645579.2012.756095

# Systematizing analytical approaches

Taking the example of Twitter:

## Analyzing the *content*

- Sentiment analysis
- Word frequencies, searchstrings
- Co-word analysis ( $\Rightarrow$ frames)

# Systematizing analytical approaches

Taking the example of Twitter:

## Analyzing the *content*


- Sentiment analysis
- Word frequencies, searchstrings
- Co-word analysis (⇒frames)

⇒ **Focus on what is said**

# Systematizing analytical approaches

⇒ **It depends on your reserach question which approach is more interesting!**

	Methodological approach		
	<i>Counting and Dictionary</i>	<i>Supervised Machine Learning</i>	<i>Unsupervised Machine Learning</i>
<b>Typical research interests and content features</b>	visibility analysis sentiment analysis subjectivity analysis	frames topics gender bias	frames topics
<b>Common statistical procedures</b>	string comparisons counting	support vector machines naive Bayes	principal component analysis cluster analysis latent dirichlet allocation semantic network analysis



Boumans, J.W., & Trilling, D. (2016). Taking stock of the toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars. *Digital Journalism*, 4, 1. 8–23.

# Data analysis 1:

## **Sentiment analysis**



# What is sentiment analysis?

Extracting subjective information from texts

# What is sentiment analysis?

## Extracting subjective information from texts

- the author's attitude towards the topic of the text

# What is sentiment analysis?

## Extracting subjective information from texts

- the author's attitude towards the topic of the text
- *polarity*: negative—positive

# What is sentiment analysis?

## Extracting subjective information from texts

- the author's attitude towards the topic of the text
- *polarity*: negative—positive
- *subjectivity*: neutral—subjective \*

# What is sentiment analysis?

## Extracting subjective information from texts

- the author's attitude towards the topic of the text
- *polarity*: negative—positive
- *subjectivity*: neutral—subjective \*
- advanced approaches: different emotions

# What is sentiment analysis?

## Extracting subjective information from texts

- the author's attitude towards the topic of the text
- *polarity*: negative—positive
- *subjectivity*: neutral—subjective \*
- advanced approaches: different emotions

\* Less sophisticated approaches do not see this as a sperate dimension but simply calculate  $objectivity = 1 - (negativity + positivity)$

# Applications

## Who uses it?

- Companies
- especially for Web Analytics
- Social Scientists
- applications in data journalism, politics, ...

Many references to examples in Mostafa (2013).

⇒ Cases in which you have a huge amount of data or real-time data and you want to get an idea of the tone.

Mostafa, M. M. (2013). More than words: Social networks' text mining for consumer brand sentiments. *Expert Systems with Applications*, 40(10), 4241– 4251. doi:10.1016/j.eswa.2013.01.019

# Example

```
1 >>> sentiment("Great service by @NSHighspeed")
2 (0.8, 0.75)
3 >>> sentiment("Bad service by @NSHighspeed")
4 (-0.6166666666666667, 0.6666666666666666)
```

(polarity, subjectivity) with

$$-1 \leq \text{polarity} \leq +1$$

$$0 \leq \text{subjectivity} \leq +1$$

This is the module `pattern.nl`, available for Python 2 via `pip`. The development branch on github supports Python 3.

De Smedt, T., & Daelemans W. (2012). Pattern for Python. *Journal of Machine Learning Research*, 13, 2063-2067.



# Data analysis 1: Sentiment analysis

## Bag-of-words approaches

# Bag-of-words approaches

## How does it work?

- We take each word of a text and look if it's positive or negative.

# Bag-of-words approaches

## How does it work?

- We take each word of a text and look if it's positive or negative.
  - Most simple way: compare it with a list of negative words and with a list of positive words (That's what Mostafa (2013) did)

# Bag-of-words approaches

## How does it work?

- We take each word of a text and look if it's positive or negative.
  - Most simple way: compare it with a list of negative words and with a list of positive words (That's what Mostafa (2013) did)
  - More advanced: look up a subjectivity score from a table

# Bag-of-words approaches

## How does it work?

- We take each word of a text and look if it's positive or negative.
  - Most simple way: compare it with a list of negative words and with a list of positive words (That's what Mostafa (2013) did)
  - More advanced: look up a subjectivity score from a table
- e.g., add up the scores and average them.

# How to do this

If you were to run an analysis like the one by Mostafa (2013), how could you do this?

# How to do this

(given a *string* `tekst` that you want to analyze and two *lists* of strings with negative and positive words, `lijstpos=["great","fantastic",...,"perfect"]` and `lijstneg`)

```
1 sentiment=0
2 for woord in tekst.split():
3     if woord in lijstpos:
4         sentiment=sentiment+1 #same as sentiment+=1
5     elif woord in lijstneg:
6         sentiment=sentiment-1 #same as sentiment-=1
7 print (sentiment)
```

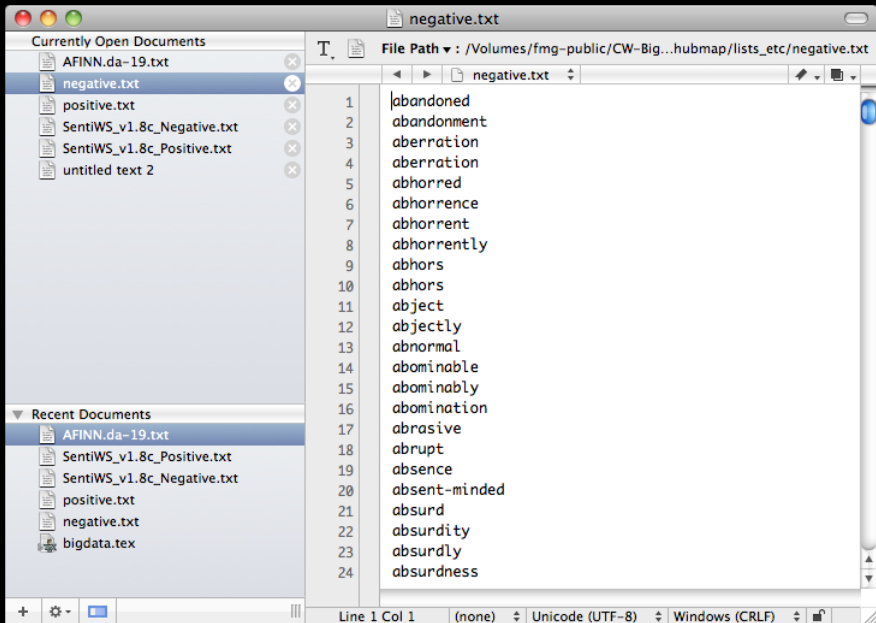
# Do we need to have the lists in our program itself?

No.

You could have them in a separate text file, one per row, and then read that file directly to a list.

```
1 poslijst=open("filewithonepositivewordperline.txt").read().splitlines()
2 neglijst=open("filewithonenegativewordperline.txt").read().splitlines()
```





# More advanced versions

- CSV files or similar tables with weights
- Or some kind of dict?

AFINN.da-19.txt

Currently Open Documents

- AFINN.da-19.txt
- negative.txt
- positive.txt
- SentiWS\_v1.8c\_Negative.txt
- SentiWS\_v1.8c\_Positive.txt
- untitled text 2

Recent Documents

- AFINN.da-19.txt
- SentiWS\_v1.8c\_Positive.txt
- SentiWS\_v1.8c\_Negative.txt
- positive.txt
- negative.txt
- bigdata.tex

File Path: /Volumes/fmg-public/CW-Big...ap/lists\_etc/AFINN.da-19.txt

AFINN.da-19.txt

1	absorberet	1
2	acceptere	1
3	accepterede	1
4	accepterer	1
5	accepteres	1
6	accepteret	1
7	advare	-2
8	advarede	-2
9	advarer	-2
10	advaret	-2
11	advarsel	-3
12	advarsler	-3
13	advarslerne	-3
14	afbrudt	-2
15	afbryde	-2
16	afbrydelse	-2
17	afbrydelser	-2
18	afbrydelserne	-2
19	afbryder	-2
20	affald	-1
21	afgift	-1
22	afgifter	-1
23	afhængig	-1
24	afhængige	-1

Line 1 Col 1 (none) Unicode (UTF-8, with BOM) Wind...CRLF

Currently Open Documents

- AFINN.da-19.txt
- negative.txt
- positive.txt
- SentiWS\_v1.8c\_Negative.txt
- SentiWS\_v1.8c\_Positive.txt
- untitled text 2

Recent Documents

- AFINN.da-19.txt
- SentiWS\_v1.8c\_Positive.txt
- SentiWS\_v1.8c\_Negative.txt
- positive.txt
- negative.txt
- bigdata.tex

File Path: /Volumes/fmg-public/CW-Big...tc/SentiWS\_v1.8c\_Negative.txt

SentiWS\_v1.8c\_Negative.txt

1	Abbau INN	-0.058	Abbaus,Abbaues,Abbauen,Abbaue
2	Abbruch INN	-0.0048	
...	Abbruches,Abbrüche,Abbruchs,Abbrüchen		
3	Abdankung INN	-0.0048	Abdankungen
4	Abdämpfung INN	-0.0048	Abdämpfungen
5	Abfall INN	-0.0048	
...	Abfalles,Abfälle,Abfalls,Abfällen		
6	Abfuhr INN	-0.3367	Abfahren
7	Abgrund INN	-0.3465	
8	Abhängigkeit INN	-0.3653	Abhängigkeiten
9	Ablehnung INN	-0.5118	Ablehnungen
10	Ablenkung INN	-0.0435	Ablenkungen
11	Abnahme INN	-0.0048	Abnahmen
12	Abneigung INN	-0.0048	Abneigungen
13	Abnutzung INN	-0.0048	
14	Abriss INN	-0.0048	
...	Abrisse,Abrissen,Abrisses,Abriss		
15	Abrutsch INN	-0.0048	
...	Abrutschen,Abrutsche,Abrutsches,Abrutschs		
16	Abschaffung INN	-0.058	Abschaffungen
17	Abschreckung INN	-0.0048	Abschreckungen
18	Abschreibung INN	-0.3345	Abschreibungen
19	Abschuß INN	-0.0048	
20	Abschwächung INN	-0.1935	Abschwächungen

Line 1 Col 1 (none) Unicode (UTF-8) Unix (LF) 216...

# Mustafa 2013: Interpreting the output

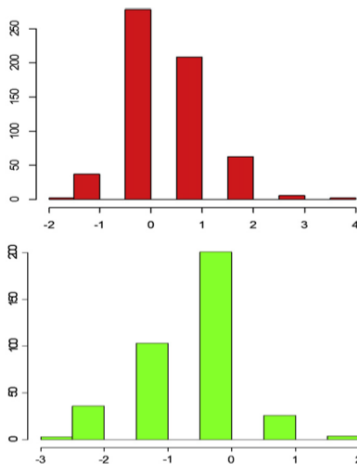


Fig. 5. Sentiment scores for Nokia (top) and Pfizer (bottom). X-axis represents score distributions, Y-axis represents count/frequencies.

# Mustafa 2013: Interpreting the output

Your thoughts?

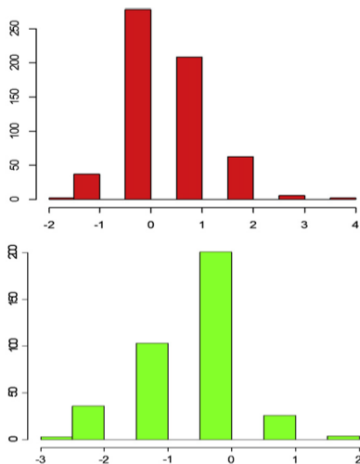


Fig. 5. Sentiment scores for Nokia (top) and Pfizer (bottom). X-axis represents score distributions, Y-axis represents count/frequencies.

# Mustafa 2013: Interpreting the output

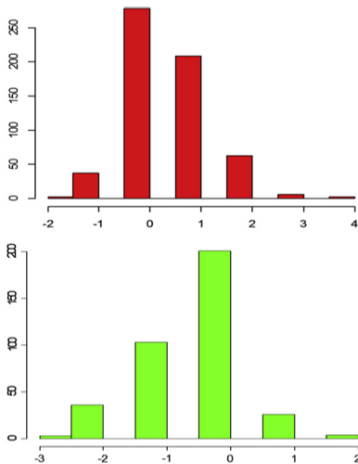


Fig. 5. Sentiment scores for Nokia (top) and Pfizer (bottom). X-axis represents score distributions, Y-axis represents count/frequencies.

Your thoughts?

- each word counts equally (1)
- many tweets contain no words from the list. What does this mean?
- Ways to improve BOW approaches?

# Bag-of-words approaches

e.g., Schut, L. (2013). Verenigde Staten vs. Verenigd Koninkrijk: Een automatische inhoudsanalyse naar verklarende factoren voor het gebruik van positive campaigning en negative campaigning door vooraanstaande politici en politieke partijen op Twitter. *Bachelor Thesis*, Universiteit van Amsterdam.



# Bag-of-words approaches

## pro

- easy to implement
- easy to modify:
  - add or remove words
  - make new lists for other languages, other categories (than positive/negative), ...
- easy to understand (transparency, reproducibility)

e.g., Schut, L. (2013). Verenigde Staten vs. Verenigd Koninkrijk: Een automatische inhoudsanalyse naar verklarende factoren voor het gebruik van positive campaigning en negative campaigning door vooraanstaande politici en politieke partijen op Twitter. *Bachelor Thesis*, Universiteit van Amsterdam.

# Bag-of-words approaches

## con

- simplistic assumptions
- e.g., intensifiers cannot be interpreted ("really" in "really good" or "really bad")
- or, even more important, negations.

# Data analysis 1: Sentiment analysis

## Advanced approaches

# Improving the BOW approach

## Example: The Sentistrength algorithm

- $-5 \dots -1$  and  $+1 \dots +5$
- spelling correction
- "booster word list" for strengthening/weakening the effect of the following word
- interpreting repeated letters ("baaaaaad"), CAPITALS and !!!
- idioms
- negation
- Idots

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social Web. *Journal of the American Society for Information Science and Technology*, 63(1), 163-173.

# Advanced approaches

## Take the structure of a text into account

- Try to apply linguistics concepts to identify sentence structure
- can identify negations
- can interpret intensifiers

# Example

```
1 from pattern.nl import sentiment
2 >>> sentiment("Great service by @NSHighspeed")
3 (0.8, 0.75)
4 >>> sentiment("Really")
5 (0.0, 1.0)
6 >>> sentiment("Really Great service by @NSHighspeed")
7 (1.0, 1.0)
```

(polarity, subjectivity) with  
 $-1 \leq \text{polarity} \leq +1$   
 $0 \leq \text{subjectivity} \leq +1$  )

Unlike in pure bag-of-words approaches, here, the overall sentiment is not just the sum or the average of its parts!

De Smedt, T., & Daelemans W. (2012). Pattern for Python. *Journal of Machine Learning Research*, 13, 2063-2067.

# Advanced approaches

# Advanced approaches

## pro

- understand intensifiers or negation
- thus: higher accuracy



# Advanced approaches

## pro

- understand intensifiers or negation
- thus: higher accuracy

## con

- Black box? Or do we understand the algorithm?
- Difficult to adapt to own needs
- *really* much better results?

## A sentiment analysis tailored to your needs!

Data analysis 1: Sentiment analysis

A sentiment analysis tailored to your needs!

# A sentiment analysis tailored to your needs!

## Identifying suicidal texts

- Bag-of-words-approach with very specific dictionary
- added negation
- added regular expression search for key phrases
- Very specific design requirements: False positives are OK, false negatives not!

Huang, Y.-P., Goh, T., & Liew, C.L. (2007). Hunting suicide notes in web 2.0 – preliminary findings. *Ninth IEEE International Symposium on Multimedia*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4476021>

## A sentiment analysis tailored to your needs!

Already this still relatively simple approach seems to work satisfactory, but if 106 scientists from 24 competing teams (!) work on it, they can

Pestian, J.P.; Matykiewicz, P., Linn-Gust, M., South, B., Uzuner, O., Wiebe, J., Cohen, K.B., Hurdle, J., & Brew, C. (2012). Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, 5(1), p. 3-16.  
Retrieved from <http://europepmc.org/articles/PMC3299408?pdf=render>

Already this still relatively simple approach seems to work satisfactory, but if 106 scientists from 24 competing teams (!) work on it, they can

group suicide notes by these characteristics:

- swear
- family
- friend
- positive emotion
- negative emotion
- anxiety
- anger
- sad
- cognitive process
- biology
- sexual
- ingestion
- religion
- death

Pestian, J.P.; Matykiewicz, P., Linn-Gust, M., South, B., Uzuner, O., Wiebe, J., Cohen, K.B., Hurdle, J., & Brew, C. (2012). Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, 5(1), p. 3-16.

Retrieved from <http://europepmc.org/articles/PMC3299408?pdf=render>

## Packages for sentiment analysis

# Which packages are easy to use?

vader pro: in NLTK module, con: English only

# Which packages are easy to use?

vader pro: in NLTK module, con: English only

pattern pro: mutiple languages (including Dutch), con:  
porting to Python 3 is still under development (but in  
principle, it works)



# Which packages are easy to use?

vader pro: in NLTK module, con: English only

pattern pro: mutiple languages (including Dutch), con:  
porting to Python 3 is still under development (but in  
principle, it works)

sentistrength pro: multiple languages, widely used, con: needs  
Python wrapper, license

vader: Chapter 6.3; pattern: Chapter 6.5; sentistrength: Chapter 6.4

# Which packages are easy to use?

vader pro: in NLTK module, con: English only

pattern pro: mutiple languages (including Dutch), con:  
porting to Python 3 is still under development (but in  
principle, it works)

sentistrength pro: multiple languages, widely used, con: needs  
Python wrapper, license

vader: Chapter 6.3; pattern: Chapter 6.5; sentistrength: Chapter 6.4

**BUT: Keep in mind that the results of *any*  
off-the-shelf-package might be biased and/or noisy in *your*  
domain!**

# A possible receipe for doing your sentiment analysis

- 1 Construct a list `data` of strings with your input data
- 2 Create an empty list `sent` for storing the results
- 3 For each text `t` in `data`, estimate the sentiment of `t` and append the result to `sent`<sup>1</sup>
- 4 Confirm that `len(data) == len(sent)`
- 5 use `zip()` and a `csv.writer` to write input and output next to each other to a csv file.

---

<sup>1</sup>use multiple lists instead if you estimate for instance subjectivity *and* polarity

# Supervised ML ( $\Rightarrow$ week7)

An alternative state-of-the-art approach:

## Use supervised machine learning

- Instead of defining rules, hand-code (“annotate”) the sentiment of some tweets manually and let the computer find out which words or characters (“features”) predict sentiment
- Then use this model to predict sentiment for other tweets
- Essentially the same like what you know since the second year of your Bachelor: regression analysis (but now with DV sentiment and IV’s word occurrences)

Gonzalez-Bailon, S., & Paltoglou, G. (2015). Signals of public opinion in online communication: A comparison of methods and data sources. *The ANNALS of the American Academy of Political and Social Science*, 659(1), 95–107.

Take-home message  
Mid-term take-home exam  
Next meetings

# Take-home messages

## What you should be familiar with:

- You should have *completely* understood last week's exercise. Re-read it if necessary.
- Approaches to the analysis (e.g., structure vs. content)
- Types of sentiment analysis, application areas, pros and cons

# Mid-term take home exam

## Week 5: Wednesday, 7 March, to Sunday, 11 March

- You get the exam on Wednesday at the end of the meeting
- Answers have to be handed in no later than Sunday, 23.59
- 30% of final grade
- 3 questions:
  - ① Literature question: E.g., different methods (“Explain how... is done”) and/or epistemological or theoretical implications (“What does this mean for social-scientific research?”)
  - ② Empirical question (conceptual)
  - ③ Empirical question (actual programming task)

If you *fully* understood all exercises until now, it shouldn't be difficult and won't take too long. But give yourself *a lot* of buffer time!!!

# Next meetings

## Wednesday, 28–2

Task for during the meeting: Conduct an own sentiment analysis!  
You can bring your own data (you will probably learn more!), but then already think about how to write some script to read the data (as we did last week and as described in Chapter 5).