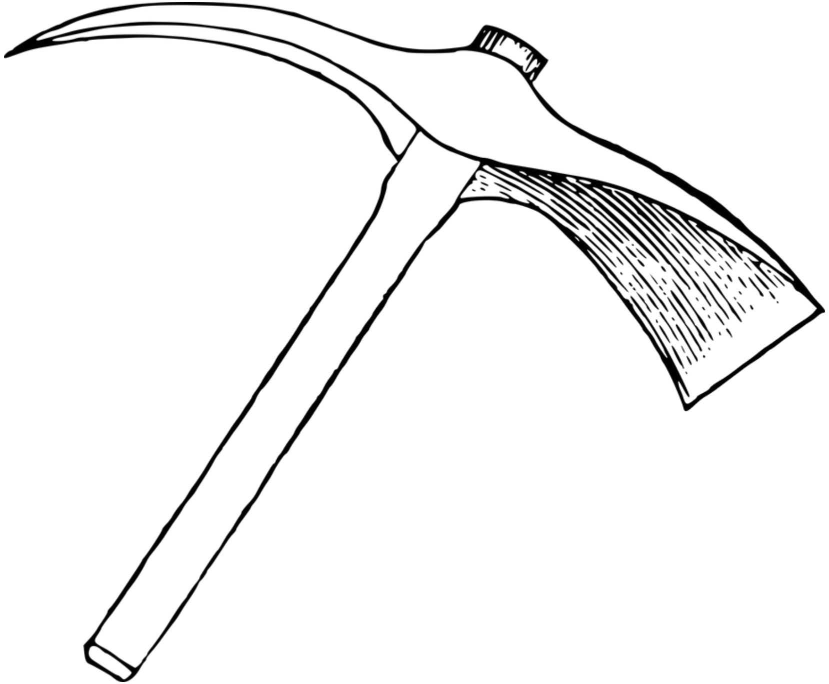
Introduction to the world of text mining

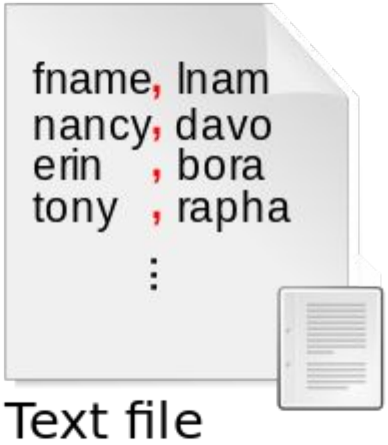
Using R and many helpful packages

Florian Gilberg
CorrelAidX Cologne

Text Mining - Get your pickaxe!



+



fname, lnam
nancy, davo
erin , bora
tony , rapha
⋮

Text file

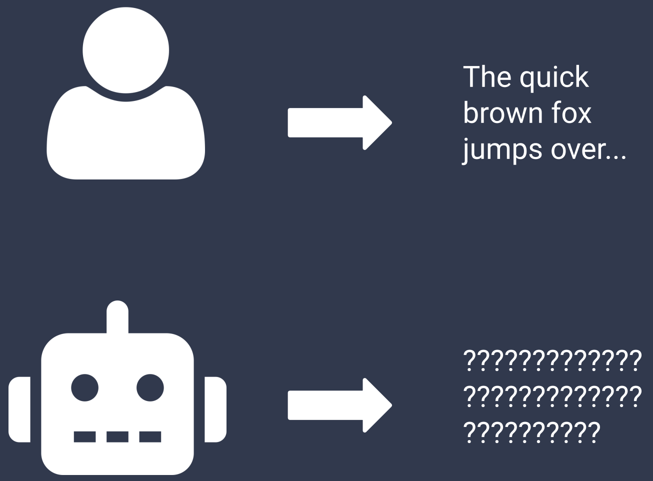
= ?

Text Mining - Key points

- Transform free (unstructured) text into normalized, structured data
- using various different Natural Language Processing (NLP) Tasks
- to discover and reveal hidden patterns and information

Transform Text into Data

Text as Data-approach



The quick brown fox jumps over...

?????????????
?????????????
??????????

(Gentzkow et al. 2019)

How can we make text machine readable?

Machine Leaning

Complicated AI

Model training

Complex computing operations

Dictionaries.

1. Cleaning

Delete all frequently occurring “filling words” that (in most cases) don’t add any value for our analysis.

These are so called “stop words”. Stop words are collected in dictionaries.

In R, you can use pre-defined dictionaries, for example tm::stopwords()

~~The~~ quick brown fox jumps over ~~the~~ lazy dog

2. Stemming / Lemmatizing

Reduce all remaining words to their “core” by chopping of the different different grammatical forms.

“Lemmata” or “Stems” can be used to improve uniformity and provide comparable data.

In R, you can lemmatize by using the TreeTagger in the udpipie-package or us the stemmer with tm::stemDocument()

stemming: chops off the ends of words in the hope of achieving this goal correctly most of the time

~~The~~ quick brow~~n~~ fox jump~~s~~ over ~~the~~ laz~~y~~ dog.

lemmatizing: compares the word to a dictionary and returns the most likely candidate in its base form

~~The~~ quick brown fox jump~~s~~ over ~~the~~ lazy dog.

3. Counting with the Bag-of-words approach

The	0
quick	1
brown	1
fox	1
jump	1
over	1
the	0
lazy	1
dog.	

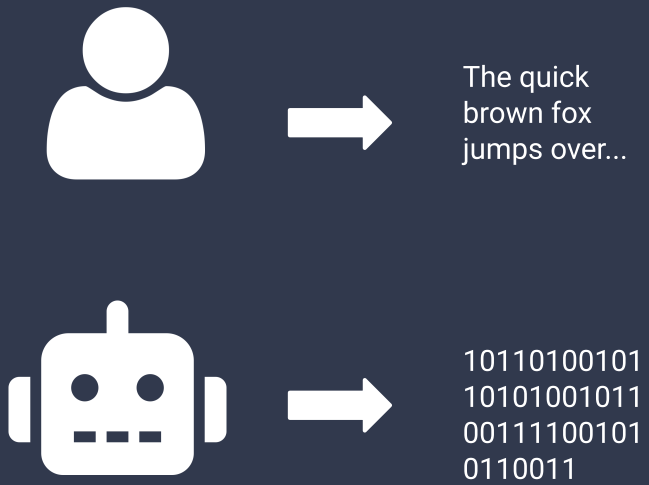
Source

3. Counting with the Bag-of-words approach

More info

	The	quick	brown	fox	jumps	over	the	lazy
Maria	0	0	0	0	0	0	0	0
jumps	0	0	0	0	**1**	0	0	0
up	0	0	0	0	0	0	0	0
quick	0	**1**	0	0	0	0	0	0
as	0	0	0	0	0	0	0	0
she	0	0	0	0	0	0	0	0
discovers	0	0	0	0	0	0	0	0

Text as Data-approach



The quick
brown fox
jumps over...

10110100101
10101001011
00111100101
0110011

(Gentzkow et al. 2019)

Discover hidden patterns

Linguistic annotation

To find patterns in speech we can use machine learning models to tokenize and annotate our documents.

Popular packages and programs for this task are StanfordNLP, SpaCy, openNLP and udpipe.

- ADJ: adjective
- ADP: adposition
- ADV: adverb
- AUX: auxiliary
- CCONJ: coordinating conjunction
- DET: determiner
- INTJ: interjection
- NOUN: noun
- NUM: numeral
- PART: particle
- PRON: pronoun
- PROPN: proper noun
- PUNCT: punctuation
- SCONJ: subordinating conjunction
- SYM: symbol
- VERB: verb
- X: other

Sentiment analysis

Discover positive and negative words or even emotions by using weighted dictionaries.

IGGSA

negative examples

Abbau|NN	-0.058	Abbaus,Abbaues,Abbauen,Abbaue,Abbauten
Abbruch|NN	-0.0048	Abbruches,Abbrüche,Abbruchs,Abbrüchen,Abbruche
Abdankung|NN	-0.0048	Abdankungen
Abdämpfung|NN	-0.0048	Abdämpfungen
Abfall|NN	-0.0048	Abfalles,Abfälle,Abfalls,Abfällen,Abfalle
Abfuhr|NN	-0.3367	Abfuhren
Abgrund|NN	-0.3465	Abgründe,Abgrunde,Abgrundes,Abgrunds,Abgründen
Abhängigkeit|NN	-0.3653	Abhängigkeiten

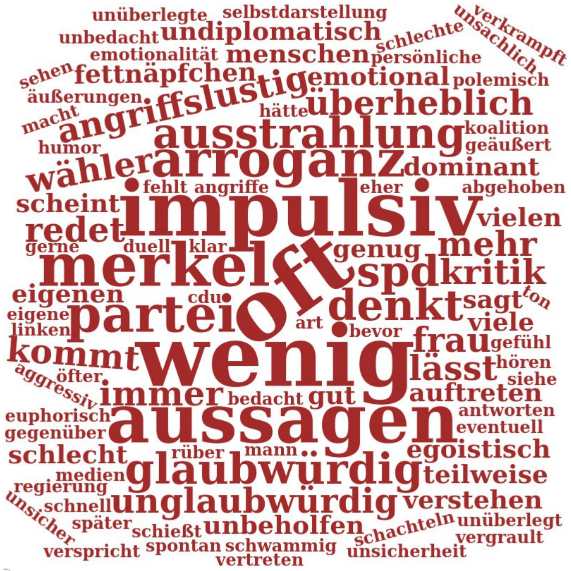
positive examples

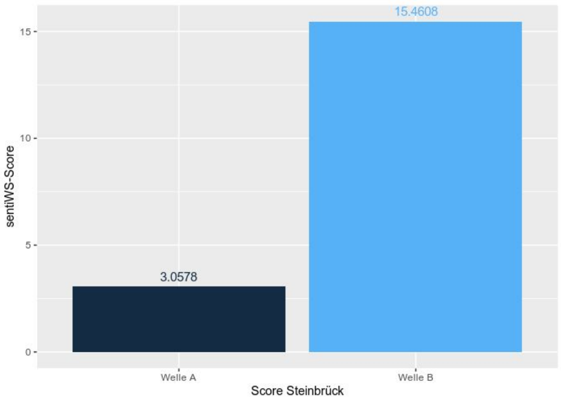
Freude|NN	0.6502	Freuden
Freund|NN	0.0116	Freunden,Freundes,Freunde,Freunds
Freundlichkeit|NN	0.0913	Freundlichkeiten
Freundschaft|NN	0.2059	Freundschaften
Frieden|NN	0.0040	Friedens
Fruchtbarkeit|NN	0.0040	
Funktionsfähigkeit|NN	0.0040	Funktionsfähigkeiten
Furchtlosigkeit|NN	0.0040	

Sentiment analysis

Discover positive and negative words or even emotions by using weighted dictionaries.

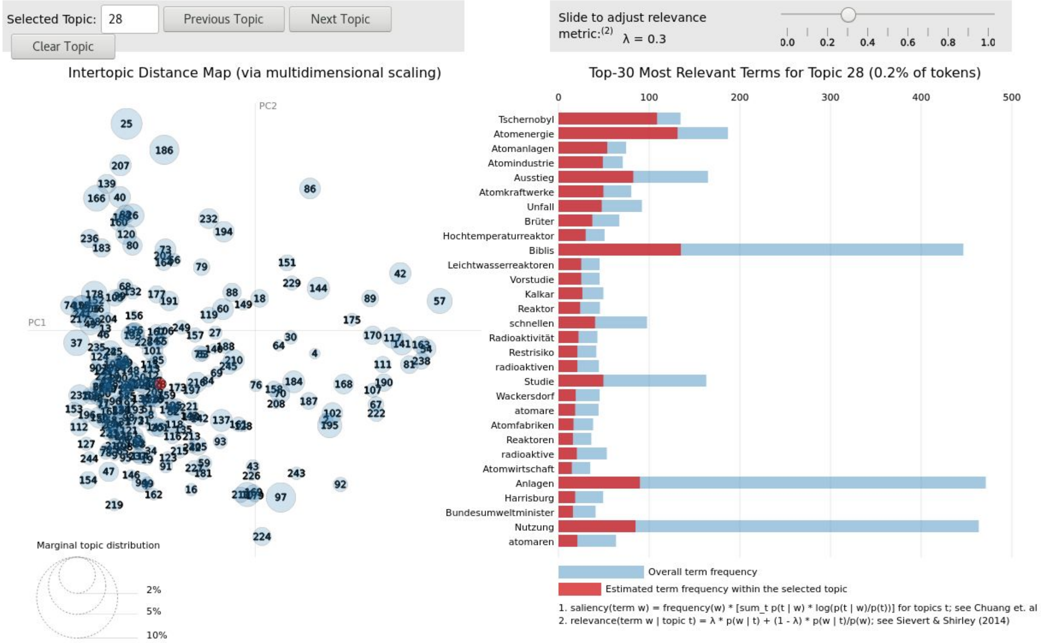
IGGSA





Topic modelling

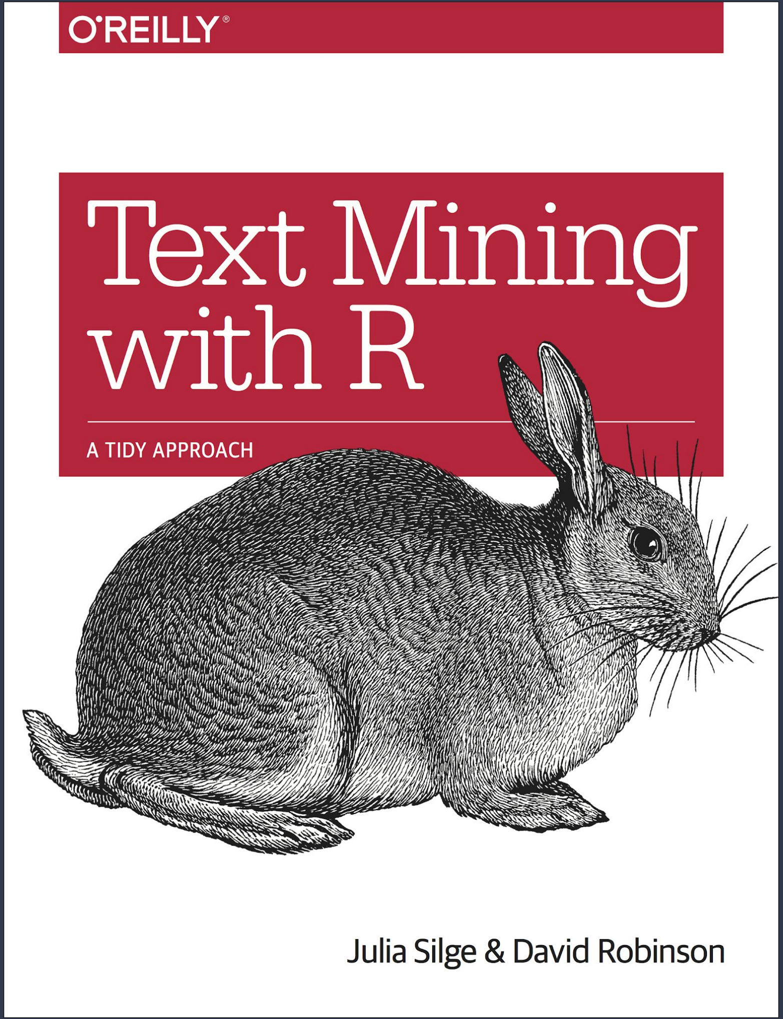
Trying to represent similarities in word occurrence by assigning them the same topic number.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Good introduction

And so much more…



O'REILLY®

Text Mining with R

A TIDY APPROACH

Julia Silge & David Robinson