Computational Communication Science 2

Week 1

Introduction & Lift Off

Anne Kroon
a.c.kroon@uva.nl

Saurabh Khanna
s.khanna@uva.nl

March 31, 2025

Digital Society Minor
University of Amsterdam

Today

Introducing. . . the people

Introducing. . . the course

Text as data

The Toolkit

- Bottom-up vs. Top-down
- Hands-on examples

Preprocessing

The people

Introducing...Saurabh

dr. Saurabh Khanna

Assistant Professor, Youth & Media Entertainment

- Studying the limits of human knowledge in an increasingly digitized world using
  - Computational methods

|s.khanna@uva.nl |https://www.uva.nl/en/profile/k/h/s.khanna/s.khanna.html

Introducing. . . Anne

dr. Anne Kroon

Associate Professor Communication, Organizations & Society

- Research focus on biased AI in recruitment, and media bias regarding minorities
- Text analysis using automated approaches, word embeddings

@annekroon |a.c.kroon@uva.nl |
http://www.uva.nl/profiel/k/r/a.c.kroon/a.c.kroon.html

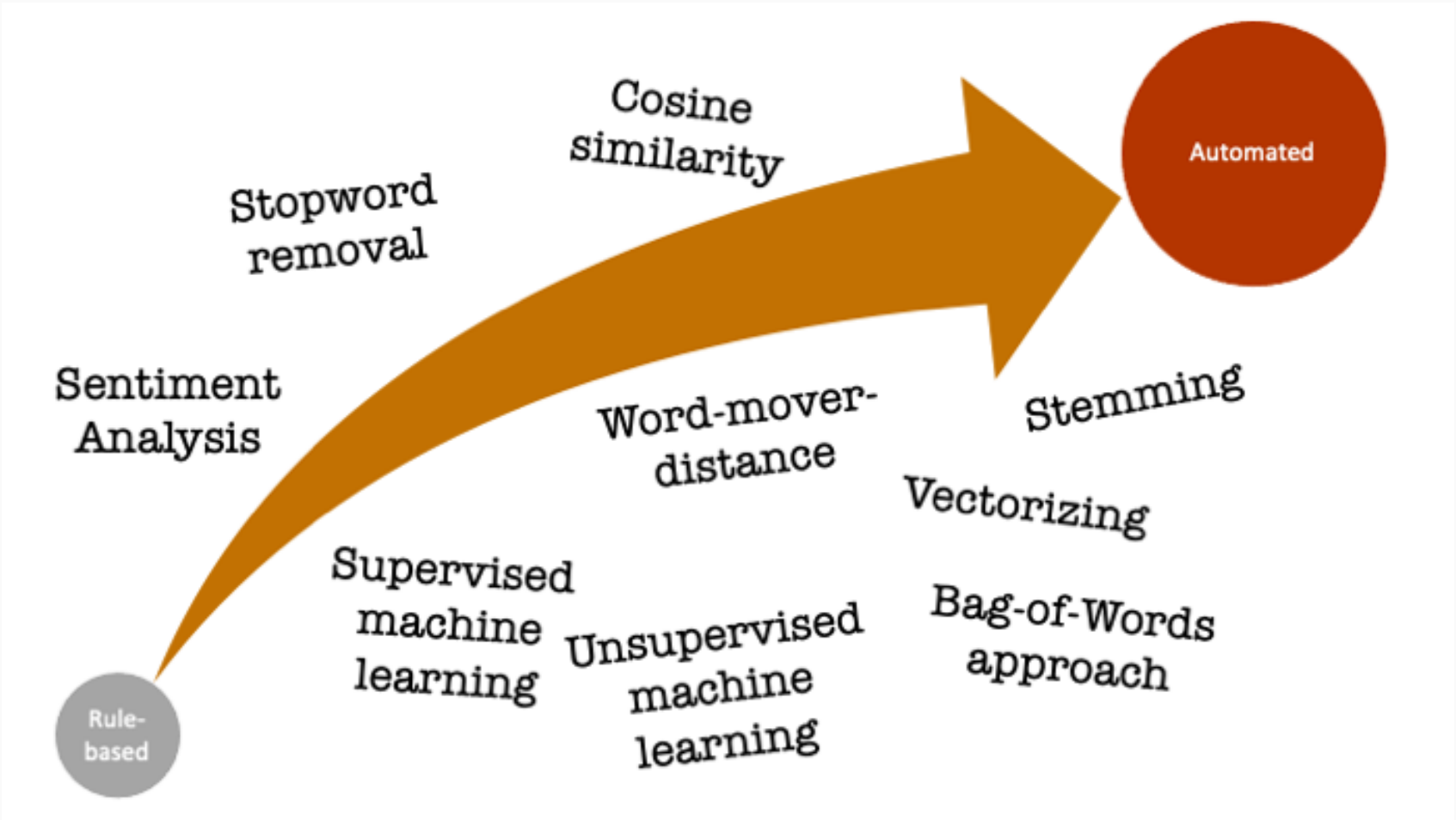
The course

About CCS-2

What is CCS-2?

- Next step after CCS-1
- Learn how to use what you learned in CCS-1 for research
- Expand on what you learned in CCS-1
  - Learn computational techniques (e.g. data vectorization, machine learning)
  - Learn how to use these techniques for research (e.g., automated content analysis)
- By the end of the course, you'll be prepared for the *Research Project*

About CCS-2


Automated
Cosine similarity
Stopword removal
Sentiment Analysis
Word-mover-distance
Stemming
Vectorizing
Supervised machine learning
Unsupervised machine learning
Bag-of-Words approach
Rule-based

About CCS-2

What will we do in this course?

- We discuss techniques in the lectures (Mondays)
- We practice with techniques in the tutorials (Tuesdays)
- Graded assignments to master the techniques:
  - Regular multiple choice questions (20%) about the readings and techniques that we discuss
    - 4 questions, in week 1, 2, 3, 7 and 8.
    - *total of 20 questions. 16 correct answers = full marks*
  - Group assignment: Get more experienced with the techniques and build a *basic* recommender system
    - Presentation – week 4 (10%)
    - Written report and code assignment – week 6 (20%)
  - In-class open book exam on May 27th (50%): at the end of the course so you can show off what you learned (materials: yes, internet: no)

About CCS-2

Course set-up:

- *Lectures*: Introduction to techniques used in computational communication science
- *Tutorial meetings*: Lab sessions to learn how to work with these techniques
    - Possibility to ask questions about your code

All course materials can be found at. . .

https://github.com/uva-cw-ccs2/2425s2

About CCS-2

How to stay informed and where to find all the materials? Regularly check:

- The course Canvas page
- Your email
- The course Github page

In addition, make sure that you read the course manual so that you know all the ins and outs of this course!

Ready? Set? Go!

Without further ado...

...let's get started!

Text as data

Text as data

CCS-1: You learned how to...

- Work with Python, for example, you:
  - Store text in json-files, csv-files etc.
  - Difference between a dict, a list, a string etc.
  - Work with data (e.g., creating a loop)

Text as data: Analyzing text as a goal

Studying text can teach us a lot about human behavior:

- . . . to study the content cancer-related online platforms (e.g., Sanders et al., 2020)
  - what *topics* are being discussed on expert and peer-generated platforms?
- . . . does content differ between online and print news? (e.g., Burggraaff and Trilling, 2020)
  - E.g., online, journalists are more likely to publish *follow-up* articles.

Text as data: Analyzing text as a means

Studying text can give us information we can use to answer broader questions:

- . . . analyze textual information about movies from IMDB to learn about the representation of women in movies (e.g.,Poma-Murialdo, 2019)
- . . . automatically distinguish between reliable and unreliable online information about vaccines by investigating what characterizes reliable and unreliable texts (e.g.,Meppelink et al., 2021)

Computational communication science

New opportunities

Gaining insights from large-scale digital data.

- **Digital traces:** Social media, news, and archives.
- **New tools:** Network and sentiment analysis, text mining.
- **Big data:** Analyzing complex datasets at scale.

Why it matters:

Real behavior • Natural contexts • Hidden patterns •

Challenges in computational research

Key challenges

- **Data access:** Ensuring open and reproducible datasets.
- **Data validity:** Addressing biases and representativeness.
- **Measurement:** Evaluating the accuracy of automated analyses.
- **Ethics:** Protecting privacy and ensuring responsible data use.

Takeaway

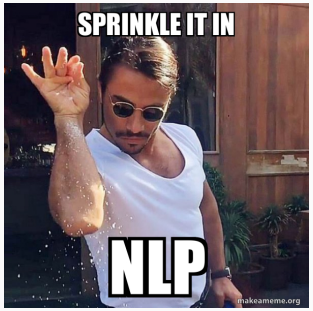
Blending traditional and computational approaches strengthens theory-driven research.

Reference: van Atteveldt and Peng, 2018

Text as data: natural language processing (NLP)

What is NLP?

Natural language processing (NLP) is a branch of computer science—specifically artificial intelligence (AI)—that enables computers to understand and interpret human language.



(IBM, 2020)

Want to know more?

Explore how NLP is reshaping data analysis in this TED Talk: https://www.ted.com/talks/jean_baptiste_michel_erez_lieberman_

The Toolkit

The Toolkit

Bottom-up vs. Top-down

Automated content analysis can follow two main approaches:

Bottom-up: Inductive, explorative, based on pattern recognition.

Top-down: Deductive, guided by pre-defined rules and categories.

Or, it can lie somewhere in between.

The CCS Toolbox in Practice

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

Source: Boumans and Trilling, 2016

Understanding the approaches

Bottom-up

- Identify frequent words or patterns.
- Explore word co-occurrence: *Which words appear together?*

Key idea: We *don't* specify what to look for in advance.

Top-down

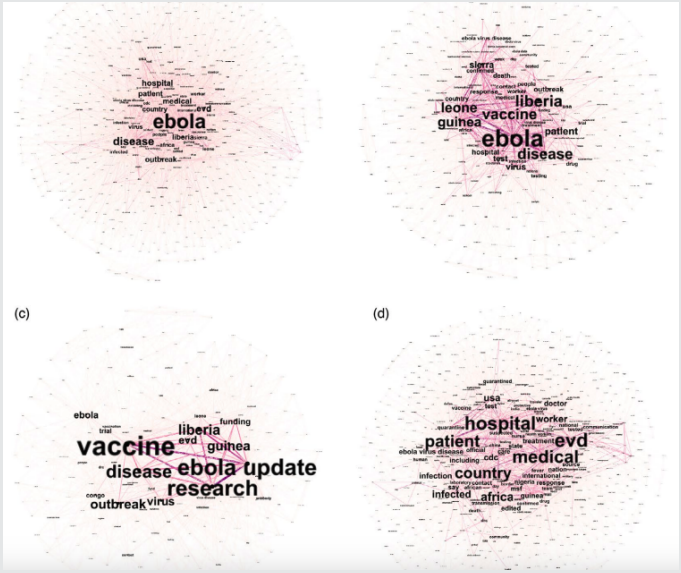
- Track occurrences of pre-defined words.
- Identify specific patterns of interest.

Key idea: We *do* specify what to look for in advance.

Research example of bottom-up approach (data-driven)

Word co-occurrence graphs in disease surveillance

- Analyzes co-occurrence of words to identify patterns in disease reporting.
- Useful for detecting emerging trends without predefined categories.



You et al., 2021

Research example of top-down approach (theory-driven)

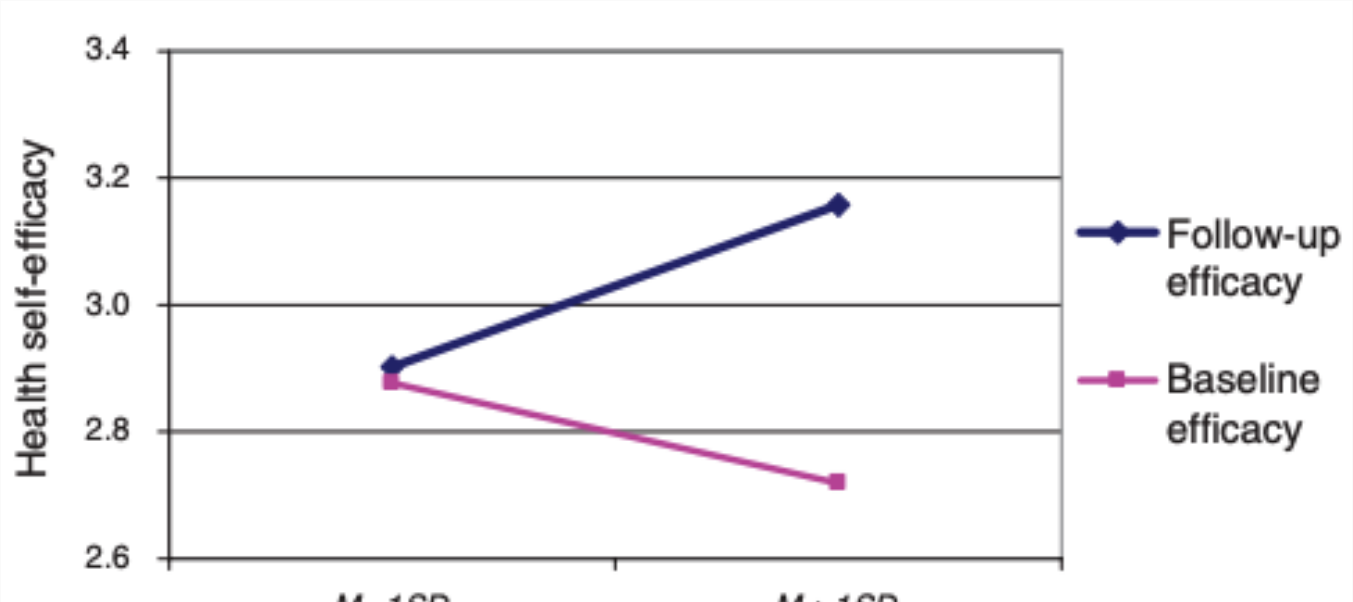
Sentiment analysis using LIWC

- LIWC is a widely applied (commercial and closed) software tool.
- Counts occurrences of predefined terms (e.g., psychological and linguistic categories).
- Guided by theoretical frameworks; useful for hypothesis testing and interpreting psychological states—but does it work?

Tausczik and Pennebaker, 2010

LIWC application in online support groups

- LIWC was used to analyze language in online support groups for breast cancer patients (Shim et al., 2011).
- It categorizes words into emotional and cognitive dimensions (e.g., insight, positive/negative emotion).
- The study found that insightful disclosure improved emotional and functional well-being by reducing breast cancer concerns.



Linking bottom-up and top-down approaches

- Both approaches complement each other in computational communication science (van Atteveldt and Peng, 2018).
- **Bottom-up:** Ideal for exploring data and discovering unknown patterns.
- **Top-down:** Useful for testing established theories and measuring specific constructs.
- **Why combine them?** It enables a more comprehensive analysis—uncovering new insights while validating theoretical frameworks.

The Toolkit

Hands-on examples

A simple bottom-up approach

Goal: Identify frequent words in a text corpus.

```
1  from collections import Counter
2  texts = ["Communication in the Digital Society is very very complex", "I like to study it"]
3  bottom_up = []
4  for t in texts:
5      bottom_up.append(Counter(t.lower().split()).most_common(3))
6  print(bottom_up)
```

This produces:

```
[('very', 2), ('communication', 1), ('in', 1)]
[('i', 1), ('like', 1), ('to', 1)]
```

Equivalent with list comprehension:

```
1  bottom_up = [Counter(t.lower().split()).most_common(3) for t in texts]
```

Tip: List comprehensions save space and improve speed!

A Simple Top-down Approach

Goal: Count specific word occurrences.

```
1  texts = ["Communication in the Digital Society is complex", "I like to study it"]
2  features = ["communication", "digital", "study"]
3
4  top_down = []
5  for t in texts:
6      counts = [t.lower().count(f) for f in features]
7      top_down.append(counts)
8  print(top_down)
```

This produces:

```
[[1, 1, 0], [0, 0, 1]]
```

Equivalent with list comprehension:

```
1  top_down = [[t.lower().count(f) for f in features] for t in texts]
```

Tip: Store results as lists for further analysis!

When would you use which approach?

Try it out yourself! https://github.com/uva-cw-ccs2/2425s2/blob/main/week01/exercise-lecture/sentiment-guesser.ipynb

Some considerations

- Both can have a place in your workflow (e.g., bottom-up as first exploratory step)
- You have a clear theoretical expectation? Bottom-up makes little sense.
- But in any case: you need to transform your text into something "countable".

Preprocessing

Preprocessing in NLP

- Text preprocessing in *Natural Language Processing*.
- Typical step to get textual data into a more structured format for subsequent analyses
- These steps will come back in the upcoming weeks when we discuss bottom-up and top-down techniques

Typical preprocessing steps

Preprocessing steps

tokenization How do we (best) split a sentence into tokens (terms, words)?

pruning How can we remove unneccessary words/ punctuation?

lemmatization and stemming How can we make sure that slight variations of the same word are not counted differently?

ngrams Neighbouring terms

Simple string methods

Slicing

`mystring[2:5]` to get the characters with indices 2,3,4

String methods

- `.lower()` returns lowercased string
- `.strip()` returns string without whitespace at beginning and end
- `.find("bla")` returns index of position of substring "bla" or -1 if not found
- `.replace("a","b")` returns string where "a" is replaced by "b"
- `.count("bla")` counts how often substring "bla" occurs

Use tab completion for more!

OK, good enough, perfect?

.split()

- space → new word
- no further processing whatsoever
- thus, only works well if we do some form of preprocessing ourselves (e.g., remove punctuation)

```
1 docs = ["This is a text",  "I haven't seen John's derring-do. Second
  ↪  sentence!"]
2 tokens = [d.split() for d in docs]
```

```
1 [['This', 'is', 'a', 'text'], ['I', "haven't", 'seen', "John's",
2 'derring-do.', 'Second', 'sentence!']]
```

OK, good enough, perfect?

Tokenizers from the NLTK package

- multiple improved tokenizers that can be used instead of .split()
- e.g., Treebank tokenizer:
  - split standard contractions ("don't")
  - deals with punctuation

```
1 from nltk.tokenize import TreebankWordTokenizer
2 tokens = [TreebankWordTokenizer().tokenize(d) for d in docs]
```

```
1 [['This', 'is', 'a', 'text'], ['I', 'have', "n't", 'seen', 'John',
↪  "'s", 'derring-do.', 'Second', 'sentence', '!']]
```

Notice the failure to split the . at the end of the first sentence in the second doc. That's because `TreebankWordTokenizer` expects *sentences* as input. See book for a solution.

Stopword removal

- *The logic of the algorithm is very much related to the one of a simple sentiment analysis!*

Stopword removal

What are stopwords?

- Very frequent words with little inherent meaning
- `the, a, he, she, ...`
- context-dependent: if you are interested in gender, `he` and `she` are no stopwords.
- Many existing lists as basis

Stopword removal: What and why?

Why remove stopwords?

- If we want to identify key terms (e.g., by means of a word count), we are not interested in them
- If we want to calculate document similarity, it might be inflated
- If we want to make a word co-occurrence graph, irrelevant information will dominate the picture

Stopword removal

```
1  from nltk.corpus import stopwords
2  mystopwords = stopwords.words("english")
3  mystopwords.extend(["test", "this"])
4
5  tokens_without_stopwords = [[word for word in doc if word.lower() not
   ↪  in mystopwords] for doc in tokens]
```

```
1  [['text'], ["n't", 'seen', 'John', 'derring-do.', 'Second',
   ↪  'sentence', '!']]
```

You can do more!

For instance, you could add an `or` statement to also exclude punctuation.

Removing punctuation

```
1 from nltk.tokenize import RegexpTokenizer
2 tokenizer = RegexpTokenizer(r'\w+')
3 tokenizer.tokenize("Hi students, what's up!")
```

```
1 ['Hi', 'students', 'what', 's', 'up']
```

Tuesday April 1

Tutorial meeting tomorrow

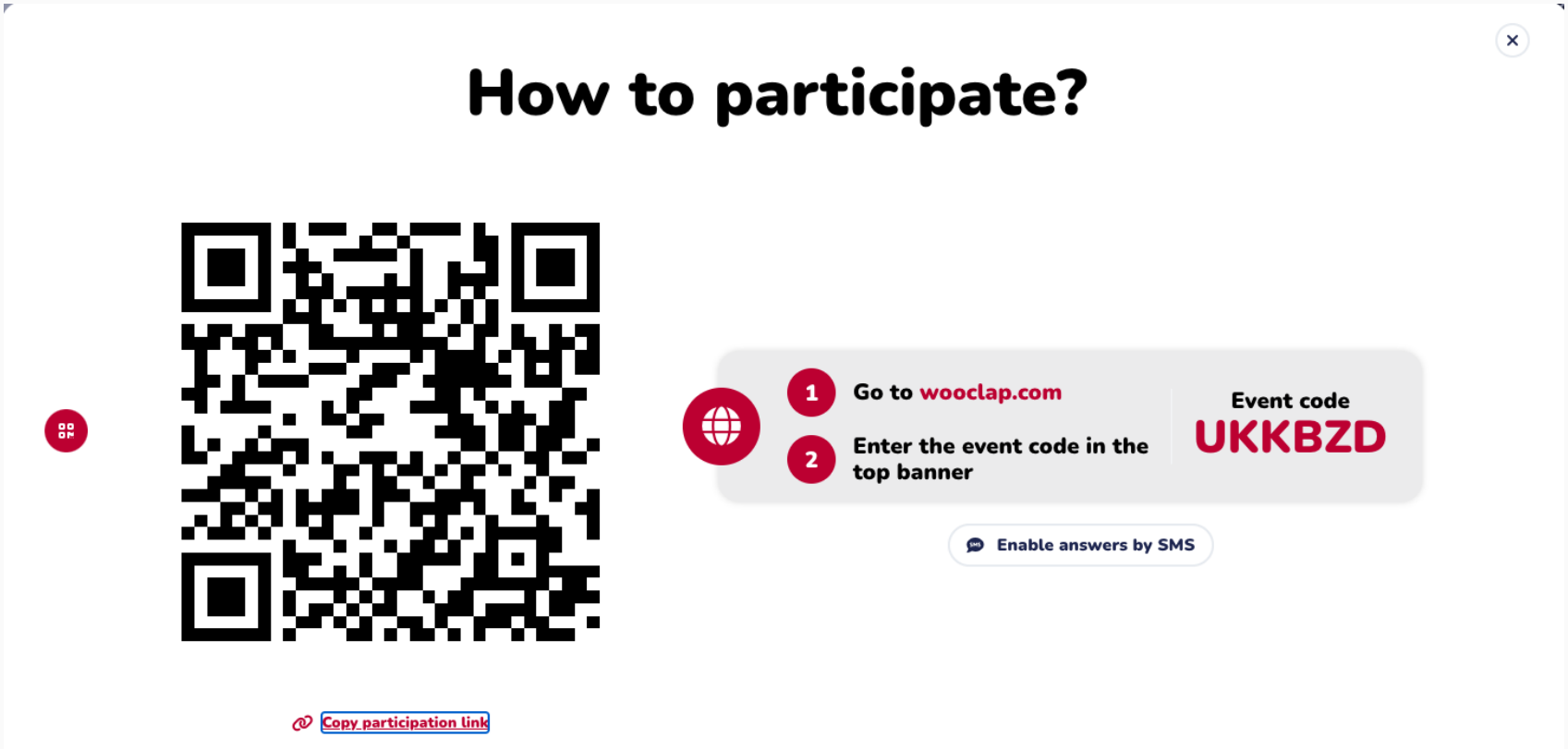
- *FIRST 4 MC QUESTIONS!*
- Start to practice with pre-processing techniques yourself
- Try the code in these slides at home. Make sure you can follow along.
- Use the tutorial meetings to discuss your questions.

Practice is key!

Thank you!!

Thank you for your attention!

- Questions? Comments?
- https://app.wooclap.com/UKKBZD?from=instruction-slide



References i

References


Boumans, J. W., & Trilling, D. (2016). **Taking stock of the toolkit: An overview of relevant autmated content analysis approaches and techniques for digital journalism scholars.** *Digital Journalism*, *4*(1), 8–23. https://doi.org/10.1080/21670811.2015.1096598


Burggraaff, C., & Trilling, D. (2020). **Through a different gate: An automated content analysis of how online news and print news differ.** *Journalism*, *21*(1), 112–129. https://doi.org/10.1177/1464884917716699

References ii

Meppelink, C. S., Hendriks, H., Trilling, D., van Weert, J. C., Shao, A., & Smit, E. S. (2021).**Reliable or not? an automated classification of webpages about early childhood vaccination using supervised machine learning.** *Patient Education and Counseling*, *104*(6), 1460–1466. https://doi.org/10.1016/j.pec.2020.11.013

Poma-Murialdo, S. C. (2019). ***Gender inequality in the movie industry:* [Master's Thesis]. University of Amsterdam.**

Sanders, R., Linn, A. J., Araujo, T. B., Vliegenthart, R., van Eenbergen, M. C., & van Weert, J. C. (2020).**Different platforms for different patients' needs: Automatic content analysis of different online health information platforms.** *International Journal of Human-Computer Studies*, *137*, 102386. https://doi.org/10.1016/j.ijhcs.2019.102386

References iii


Shim, M., Cappella, J. N., & Han, J. Y. (2011). **How does insightful and emotional disclosure bring potential health benefits? study based on online support groups for women with breast cancer.** *Journal of Communication*, *61*(3), 432–454.


Tausczik, Y. R., & Pennebaker, J. W. (2010). **The psychological meaning of words: Liwc and computerized text analysis methods.** *Journal of language and social psychology*, *29*(1), 24–54.


van Atteveldt, W., & Peng, T.-Q. (2018). **When communication meets computation: Opportunities, challenges, and pitfalls in computational communication science.** *Communication Methods and Measures*, *12*(2-3), 81–92. https://doi.org/10.1080/19312458.2018.1458084

References iv

You, J., Expert, P., & Costelloe, C. (2021). **Using text mining to track outbreak trends in global surveillance of emerging diseases: Promed-mail.** *Journal of the Royal Statistical Society Series A: Statistics in Society*, *184*(4), 1245–1259.