


Session 1

Key concepts for automated classification
and introduction to bag-of-words approaches

Michele Scotto di Vettimo

King's College London

 <https://mscottodivettimo.github.io/>

 michele.scotto_di_vettimo@kcl.ac.uk

LISS2117 · *Quantitative methods for text classification and topic detection*

Programme

- ▶ Introduction
 - Who?
 - What?
 - How?
- ▶ Text-as-data
- ▶ Bag-of-words representations
- ▶ Basic approaches to classification
- ▶ Validation of results

Who

Who: Dr. Michele Scotto di Vettimo

► About me

Political scientist, PhD at King's College London (2022)

Postdoc at the Centre for Computational Social Sciences of the University of Exeter

Currently Research Associate in the Department of Political Economy at King's College London

► Get in touch

michele.scotto_di_vettimo@kcl.ac.uk

www.mscottodivettimo.github.io/ | [@michelesdv.bsky.social](https://www.bsky.social/michelesdv)

- my research: **European Union** | **public opinion** | **policy responsiveness** | **non-majoritarian institutions**

What

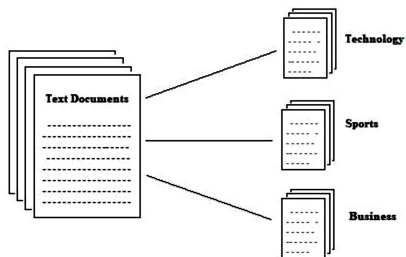
What: automated classification of texts

We will:

focus on a specific task among those that can be performed with text-as-data approaches: **text classification**;

highlight its conceptual flexibility and proximity with other tasks;

cover various ways in which it can be implemented.



How

How: course format

The course is designed to be useful for students with different levels of familiarity with text-as-data approaches to classification (e.g., should be useful both to those who are considering using these methods but still know little about them, and to those that are already more experienced but want to learn more about a specific technique).

How: course format

The course is designed to be useful for students with different levels of familiarity with text-as-data approaches to classification (e.g., should be useful both to those who are considering using these methods but still know little about them, and to those that are already more experienced but want to learn more about a specific technique).

Our sessions will be covering both conceptual and theoretical fundamentals of automated text classification, and practical coding demonstrations and exercises. Feel free to stop me any time to ask questions.

How: course format

The course is designed to be useful for students with different levels of familiarity with text-as-data approaches to classification (e.g., should be useful both to those who are considering using these methods but still know little about them, and to those that are already more experienced but want to learn more about a specific technique).

Our sessions will be covering both conceptual and theoretical fundamentals of automated text classification, and practical coding demonstrations and exercises. Feel free to stop me any time to ask questions.

Required readings cover essential aspects of what will be discussed in class. Hence, you should read them before our classes. Optional readings give you additional food-for-thought and examples of applications of specific methodologies.

How: course structure

► **Outline:**

Session 1: introduction, key concepts and basic approaches (May 7, 2025)

Session 2: topics models and machine learning algorithms (May 14, 2025)

Session 3: word-embeddings and large language models (May 21, 2025)

How: course structure

► Outline:

Session 1: introduction, key concepts and basic approaches (May 7, 2025)

Session 2: topics models and machine learning algorithms (May 14, 2025)

Session 3: word-embeddings and large language models (May 21, 2025)

► Practicalities:

- Sessions runs from 10am to 4pm
- There will be short breaks as appropriate, as well as a longer break around 1pm
- Slides will be shared the day before each session
- All materials can be accessed on the KEATS page of the course, or at [this page](#)
- A full reading list with additional readings is provided [here](#)
- We will use R and Python coding languages. Please refer to [this document](#) for guidance on setting up the software.

Text-as-data

Text-as-data: an overview

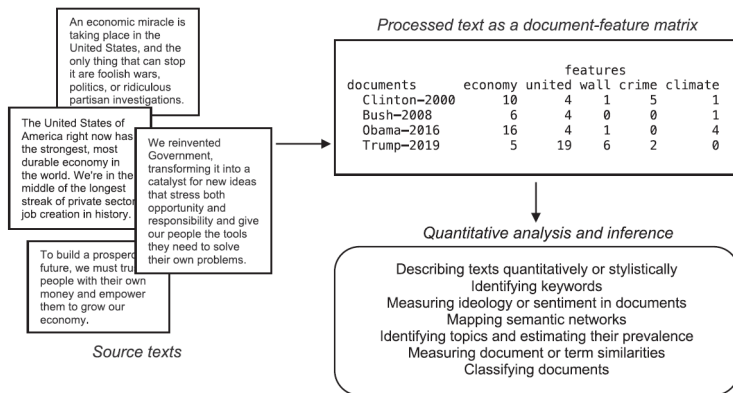
- ▶ Text as Text *versus* Text as Data

- “Ironically, generating insight from text as data is only possible once we have destroyed our ability to make sense of the texts directly. To make it useful as data, we had to obliterate the structure of the original text [...].” [Benoit et al., 2020]

Text-as-data: an overview

► Text as Text *versus* Text as Data

- “Ironically, generating insight from text as data is only possible once we have destroyed our ability to make sense of the texts directly. To make it useful as data, we had to obliterate the structure of the original text [...]” [Benoit et al., 2020]



Text-as-data: an overview

- Text as Data approaches cover a wide variety of methods and tasks

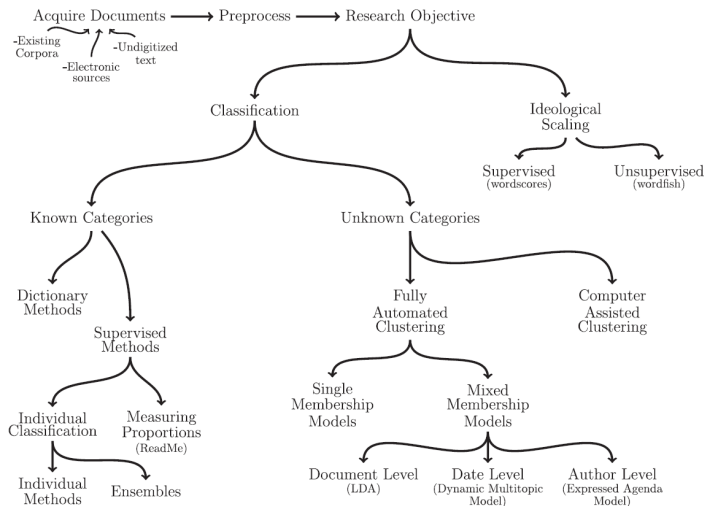


Fig. 1 An overview of text as data methods.

[Grimmer and Stewart, 2013]

Text-as-data: an overview

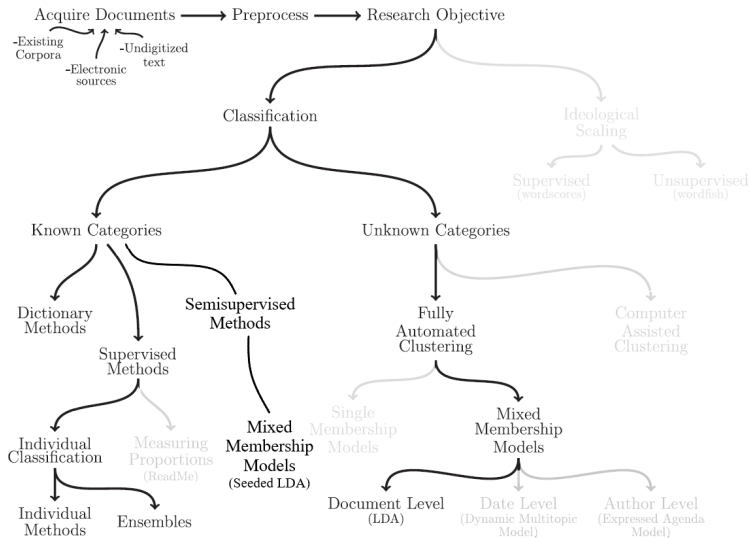


Fig. 1 An overview of text as data methods.

Text-as-data: an overview

How and why (should) we use text-as-data methods?

Text-as-data: an overview

How and why (should) we use text-as-data methods?

- ▶ The “six principles of text analysis” [[Grimmer et al., 2022](#)]:

Text-as-data: an overview

How and why (should) we use text-as-data methods?

- ▶ The “six principles of text analysis” [Grimmer et al., 2022]:
 1. Theories and substantive knowledge are essential for research design;

Text-as-data: an overview

How and why (should) we use text-as-data methods?

- ▶ The “six principles of text analysis” [Grimmer et al., 2022]:
 1. Theories and substantive knowledge are essential for research design;
 2. Text analysis augments – not replaces – humans;

Text-as-data: an overview

How and why (should) we use text-as-data methods?

- ▶ The “six principles of text analysis” [Grimmer et al., 2022]:
 1. Theories and substantive knowledge are essential for research design;
 2. Text analysis augments – not replaces – humans;
 3. Refining and testing theories requires iteration and cumulation;

Text-as-data: an overview

How and why (should) we use text-as-data methods?

- ▶ The “six principles of text analysis” [Grimmer et al., 2022]:
 1. Theories and substantive knowledge are essential for research design;
 2. Text analysis augments – not replaces – humans;
 3. Refining and testing theories requires iteration and cumulation;
 4. Text analysis methods distill generalisations from language;

Text-as-data: an overview

How and why (should) we use text-as-data methods?

- ▶ The “six principles of text analysis” [Grimmer et al., 2022]:
 1. Theories and substantive knowledge are essential for research design;
 2. Text analysis augments – not replaces – humans;
 3. Refining and testing theories requires iteration and cumulation;
 4. Text analysis methods distill generalisations from language;
 5. The best methods depends on the task;

Text-as-data: an overview

How and why (should) we use text-as-data methods?

- ▶ The “six principles of text analysis” [Grimmer et al., 2022]:
 1. Theories and substantive knowledge are essential for research design;
 2. Text analysis augments – not replaces – humans;
 3. Refining and testing theories requires iteration and cumulation;
 4. Text analysis methods distill generalisations from language;
 5. The best methods depends on the task;
 6. Validations are essential and depend on th theory and the task.

Text-as-data: an overview

How and why (should) we use text-as-data methods?

The “six principles of text analysis” [Grimmer et al., 2022]:

Theories and substantive knowledge are essential for research design;

2. Text analysis augments – not replaces – humans;

Refining and testing theories required iteration and cumulation;

4. Text analysis methods distill generalisations from language;
5. The best methods depends on the task;
6. Validations are essential and depend on the theory and the task.

Text-as-data: a general workflow

There is a general workflow that is more or less common to all text-as-data tasks

Text-as-data: a general workflow

There is a general workflow that is more or less common to all text-as-data tasks

- 1 Data collection: identifying texts
 - o Obvious but important step. It can introduce bias in the analysis
[Grimmer and Stewart, 2013, Grimmer et al., 2022]

Text-as-data: a general workflow

There is a general workflow that is more or less common to all text-as-data tasks

- ➊ Data collection: identifying texts
 - Obvious but important step. It can introduce bias in the analysis
[Grimmer and Stewart, 2013, Grimmer et al., 2022]
- ➋ Text transformation: from words to numbers
 - Bag-of-words representation
 - Embeddings representation

Text-as-data: a general workflow

There is a general workflow that is more or less common to all text-as-data tasks

- ❶ Data collection: identifying texts
 - Obvious but important step. It can introduce bias in the analysis
[Grimmer and Stewart, 2013, Grimmer et al., 2022]
- ❷ Text transformation: from words to numbers
 - Bag-of-words representation
 - Embeddings representation
- ❸ Text pre-processing: reducing complexity
 - Not strictly required, but often useful
 - Should be appropriate for the method and goal of our research

Text-as-data: a general workflow

There is a general workflow that is more or less common to all text-as-data tasks

- ❶ Data collection: identifying texts
 - Obvious but important step. It can introduce bias in the analysis
[Grimmer and Stewart, 2013, Grimmer et al., 2022]
- ❷ Text transformation: from words to numbers
 - Bag-of-words representation
 - Embeddings representation
- ❸ Text pre-processing: reducing complexity
 - Not strictly required, but often useful
 - Should be appropriate for the method and goal of our research
- ❹ Implementation of the text-as-data method
 - Should be suitable for both the data and the research question we have

Text-as-data: a general workflow

There is a general workflow that is more or less common to all text-as-data tasks

- ➊ Data collection: identifying texts
 - Obvious but important step. It can introduce bias in the analysis
[Grimmer and Stewart, 2013, Grimmer et al., 2022]
- ➋ Text transformation: from words to numbers
 - Bag-of-words representation
 - Embeddings representation
- ➌ Text pre-processing: reducing complexity
 - Not strictly required, but often useful
 - Should be appropriate for the method and goal of our research
- ➍ Implementation of the text-as-data method
 - Should be suitable for both the data and the research question we have
- ➎ Validation of the results
 - How can we convince someone that our selected method works?

Bag-of-words representations

Bag-of-words representations: key concepts

Bag-of-words models are the most common text representation.

Each text is represented by counting how many times each word appears in it, as if each text is just an unordered collection (a “bag”) of items (words).

Bag-of-words representations: key concepts

Bag-of-words models are the most common text representation.

Each text is represented by counting how many times each word appears in it, as if each text is just an unordered collection (a “bag”) of items (words).

Key to this representation is the “**document-feature matrix**” (abbreviated **dfm**; sometimes called also “document-term matrix”):

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

Bag-of-words representations: key concepts

We start by converting the selected texts (we use the collective noun “**corpus**” for the collection of our texts) into **tokens**, a step called “tokenisation”

- Tokens are the individual units we split our texts into, before counting them

Bag-of-words representations: key concepts

We start by converting the selected texts (we use the collective noun “**corpus**” for the collection of our texts) into **tokens**, a step called “tokenisation”

- Tokens are the individual units we split our texts into, before counting them
- We could think of tokens as the individual words making up the text, though this does not have to be the case. For instance, the sentence “*The American president lives in the White House*” can be tokenised as:
 - a. The; American; president; lives; in; the; White; House

Bag-of-words representations: key concepts

We start by converting the selected texts (we use the collective noun “**corpus**” for the collection of our texts) into **tokens**, a step called “tokenisation”

- Tokens are the individual units we split our texts into, before counting them
- We could think of tokens as the individual words making up the text, though this does not have to be the case. For instance, the sentence “*The American president lives in the White House*” can be tokenised as:
 - a. The; American; president; lives; in; the; White; House
 - b. The; American; president; lives; in; the; White_House
- “White_House” is a n -gram, an ordered set of n words

Bag-of-words representations: key concepts

We start by converting the selected texts (we use the collective noun “**corpus**” for the collection of our texts) into **tokens**, a step called “tokenisation”

- Tokens are the individual units we split our texts into, before counting them
- We could think of tokens as the individual words making up the text, though this does not have to be the case. For instance, the sentence “*The American president lives in the White House*” can be tokenised as:
 - a. The; American; president; lives; in; the; White; House
 - b. The; American; president; lives; in; the; White_House
- “White_House” is a n -gram, an ordered set of n words
- n -grams count as single tokens (so sentence *a.* has 8 tokens, whereas sentence *b.* has 7), and they can be introduced in the tokenisation every time we need to take into account multi-word expressions or order of words in the dfm.

Bag-of-words representations: key concepts

Before using the tokenised corpus to construct the document-feature matrix, we could reduce the complexity via **pre-processing** our tokens.

Bag-of-words representations: key concepts

Before using the tokenised corpus to construct the document-feature matrix, we could reduce the complexity via **pre-processing** our tokens.

This has many practical advantages:

- Removes non-informative tokens (e.g., articles, very common tokens)
- Speeds up analysis by reducing dimensions of our data
- Makes analysis more parsimonious and reduce risk of over-fitting

Bag-of-words representations: key concepts

Before using the tokenised corpus to construct the document-feature matrix, we could reduce the complexity via **pre-processing** our tokens.

This has many practical advantages:

- Removes non-informative tokens (e.g., articles, very common tokens)
- Speeds up analysis by reducing dimensions of our data
- Makes analysis more parsimonious and reduce risk of over-fitting

Note: Pre-processing is useful and generally advised when working with bag-of-words approaches. However, that is not the case when working with embeddings representation (in fact, it can harm performance)

Bag-of-words representations: key concepts

Common pre-processing steps to reduce complexity are:

Bag-of-words representations: key concepts

Common pre-processing steps to reduce complexity are:

- o Lowercasing: “the” and “The” → “the”

Bag-of-words representations: key concepts

Common pre-processing steps to reduce complexity are:

- Lowercasing: “the” and “The” → “the”
- Removing punctuation and numbers

Bag-of-words representations: key concepts

Common pre-processing steps to reduce complexity are:

- Lowercasing: “the” and “The” → “the”
- Removing punctuation and numbers
- Removing stop words (e.g., “and”, “the”, and so on)

Bag-of-words representations: key concepts

Common pre-processing steps to reduce complexity are:

- Lowercasing: “the” and “The” → “the”
- Removing punctuation and numbers
- Removing stop words (e.g., “and”, “the”, and so on)
- Grouping tokens in equivalent classes
 - ▶ Lemmatisation: “see”, “seeing”, and “saw” → “see”
 - ▶ Stemming: “familiar”, “family”, and “families” → “famili”

Bag-of-words representations: key concepts

Common pre-processing steps to reduce complexity are:

- Lowercasing: “the” and “The” → “the”
- Removing punctuation and numbers
- Removing stop words (e.g., “and”, “the”, and so on)
- Grouping tokens in equivalent classes
 - ▶ Lemmatisation: “see”, “seeing”, and “saw” → “see”
 - ▶ Stemming: “familiar”, “family”, and “families” → “famili”
- Removing very rare or very common words based on frequency

Bag-of-words representations: key concepts

Common pre-processing steps to reduce complexity are:

- Lowercasing: “the” and “The” → “the”
- Removing punctuation and numbers
- Removing stop words (e.g., “and”, “the”, and so on)
- Grouping tokens in equivalent classes
 - ▶ Lemmatisation: “see”, “seeing”, and “saw” → “see”
 - ▶ Stemming: “familiar”, “family”, and “families” → “famili”
- Removing very rare or very common words based on frequency

These steps help reducing the size of the final document-feature matrix, thus making subsequent analysis easier and faster.

However, you should always think carefully about the appropriateness (or even just the ordering!) of these pre-processing steps in the context of your research focus.

Rethinking this default procedure and tailoring it to your needs and to the specificities of the selected text-as-data method is key to avoid a poor use of the available data [Grimmer et al., 2022, 57-59]



We will be mostly relying on the **quanteda** package:

- o `corpus()`
- o `tokens()`
- o `dfm()`
- o `tokens_*()`, `dfm_*()`

When lost, cry for `help()`! Like this: `help(corpus)`

Basic approaches to classification

Basic approaches: keyword counting and dictionary methods

In general terms, automated classification methods for texts rely on models that map the text representation (the tokens) to the category (e.g., a topic) the text belongs to.

Basic approaches: keyword counting and dictionary methods

In general terms, automated classification methods for texts rely on models that map the text representation (the tokens) to the category (e.g., a topic) the text belongs to.

Statistical models reconstruct this mapping from sets of annotated texts, and then use what they have learned to predict the categories of other texts (we'll see these models in Session 2).

Basic approaches: keyword counting and dictionary methods

In general terms, automated classification methods for texts rely on models that map the text representation (the tokens) to the category (e.g., a topic) the text belongs to.

Statistical models reconstruct this mapping from sets of annotated texts, and then use what they have learned to predict the categories of other texts (we'll see these models in Session 2).

Yet, there are simpler strategies where the mapping of tokens into categories is explicitly provided by the analyst:

- Keyword counting
- Dictionary methods

Basic approaches: keyword counting and dictionary methods

In general terms, automated classification methods for texts rely on models that map the text representation (the tokens) to the category (e.g., a topic) the text belongs to.

Statistical models reconstruct this mapping from sets of annotated texts, and then use what they have learned to predict the categories of other texts (we'll see these models in Session 2).

Yet, there are simpler strategies where the mapping of tokens into categories is explicitly provided by the analyst:

- Keyword counting
- Dictionary methods

Both approaches stand on the assumption that human-defined rules at the token level can be used to determine labels (assignment into a category) at the text level

Basic approaches: keyword counting and dictionary methods

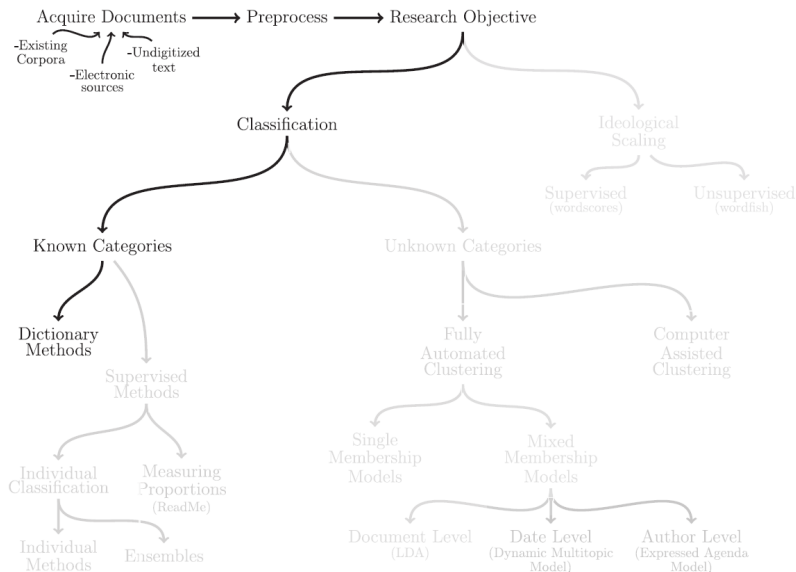


Fig. 1 An overview of text as data methods.

Basic approaches: keyword counting and dictionary methods

► Keyword counting

- Counting instances of a word or phrase that should be indicative of the category of interest:

“European Union” relates to the category “EU politics” (human defined mapping)

“European Union” is in text → text is about EU politics

“European Union” is not in text → text is not about EU politics

Basic approaches: keyword counting and dictionary methods

► Keyword counting

- Counting instances of a word or phrase that should be indicative of the category of interest:

“European Union” relates to the category “EU politics” (human defined mapping)

“European Union” is in text → text is about EU politics

“European Union” is not in text → text is not about EU politics

► Dictionary methods

- Generalisation of keyword counting: list of keywords are assigned to two or more categories
- They can be used to classify texts into categories or to measure “tone” of documents (“polarised dictionaries”)
- Many off-the-shelf dictionaries are available for different purposes, or researchers can create (and validate) their own

Basic approaches: keyword counting and dictionary methods

► Keyword counting

- Counting instances of a word or phrase that should be indicative of the category of interest:

“European Union” relates to the category “EU politics” (human defined mapping)

“European Union” is in text → text is about EU politics

“European Union” is not in text → text is not about EU politics

► Dictionary methods

- Generalisation of keyword counting: list of keywords are assigned to two or more categories
- They can be used to classify texts into categories or to measure “tone” of documents (“polarised dictionaries”)
- Many off-the-shelf dictionaries are available for different purposes, or researchers can create (and validate) their own

You can find examples of applications of keyword counting and dictionary methods among the readings under “Dictionary methods and semantic scaling” [on this page](#).

Basic approaches: keyword counting and dictionary methods

► Pros

- Simple, intuitive, and “self-evident”
- Flexible, as keywords can be adapted to context with low effort
- Easily exportable and replicable

Basic approaches: keyword counting and dictionary methods

► Pros

- Simple, intuitive, and “self-evident”
- Flexible, as keywords can be adapted to context with low effort
- Easily exportable and replicable

► Cons

- Dictionary keywords might be very context dependent
- Sometimes the concept we are interested is nuanced, and keywords would not capture it satisfactorily
- Keywords can be ambiguous
- Exhaustive dictionaries can become very long
- Need to anticipate evolution of language (or be updated regularly)

Basic approaches: keyword counting and dictionary methods

Off-the-shelf dictionaries *versus* do-it-yourself (DIY) dictionaries

Basic approaches: keyword counting and dictionary methods

Off-the-shelf dictionaries *versus* do-it-yourself (DIY) dictionaries

- ▶ Off-the-shelf dictionaries
 - Generally, they have been already tested and validated
 - Will make your findings comparable with other studies using the same dictionary
 - However, one needs to determine how context dependent they are, and if using them in the case at hand is appropriate

Basic approaches: keyword counting and dictionary methods

Off-the-shelf dictionaries *versus* do-it-yourself (DIY) dictionaries

► Off-the-shelf dictionaries

- Generally, they have been already tested and validated
- Will make your findings comparable with other studies using the same dictionary
- However, one needs to determine how context dependent they are, and if using them in the case at hand is appropriate

► DIY dictionaries

- Tailored to your needs, and useful if alternatives are unsatisfactory/unavailable
- Good dictionaries can be valuable contributions in their own right, and will be used by other as well
- Construction can be time consuming [King et al., 2017, Watanabe and Zhou, 2020]
- You cannot escape thorough testing and validation

Basic approaches: keyword counting and dictionary methods

Off-the-shelf dictionaries *versus* do-it-yourself (DIY) dictionaries

- ▶ Off-the-shelf dictionaries
 - Generally, they have been already tested and validated
 - Will make your findings comparable with other studies using the same dictionary
 - However, one needs to determine how context dependent they are, and if using them in the case at hand is appropriate
- ▶ DIY dictionaries
 - Tailored to your needs, and useful if alternatives are unsatisfactory/unavailable
 - Good dictionaries can be valuable contributions in their own right, and will be used by other as well
 - Construction can be time consuming [King et al., 2017, Watanabe and Zhou, 2020]
 - You cannot escape thorough testing and validation
- ▶ As a compromise, one can start from a validated dictionary and make additions or deletions so as to adapt it to the research needs

Basic approaches: keyword counting and dictionary methods

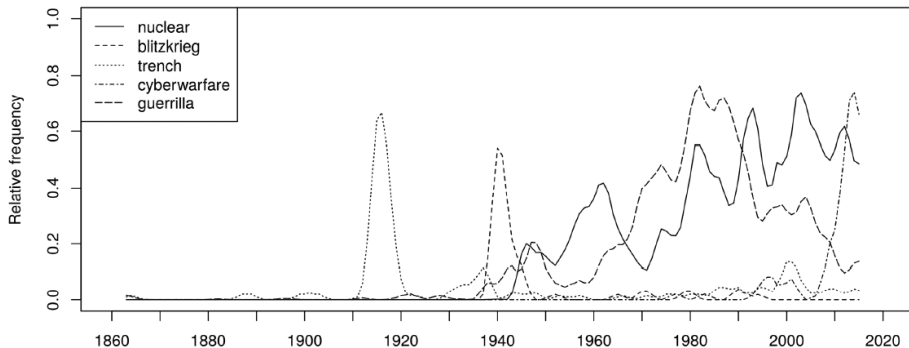


Figure 6. Frequency of keywords for international ideologies and military strategies.

[Trubowitz and Watanabe, 2021]

Basic approaches: keyword counting and dictionary methods

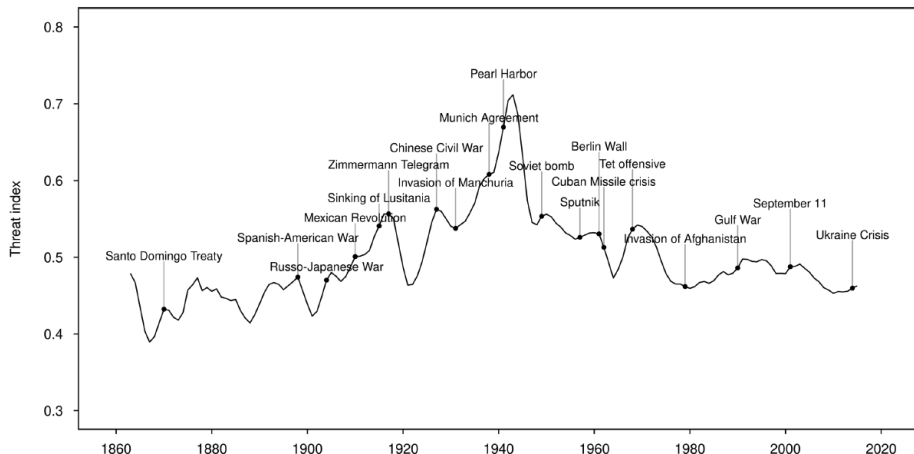


Figure 7. Baseline GTI for United States, 1861–2017. (Based on full sample of 225 countries and kernel smoothed by ± 1 year.)

[Trubowitz and Watanabe, 2021]

Basic approaches: keyword counting and dictionary methods

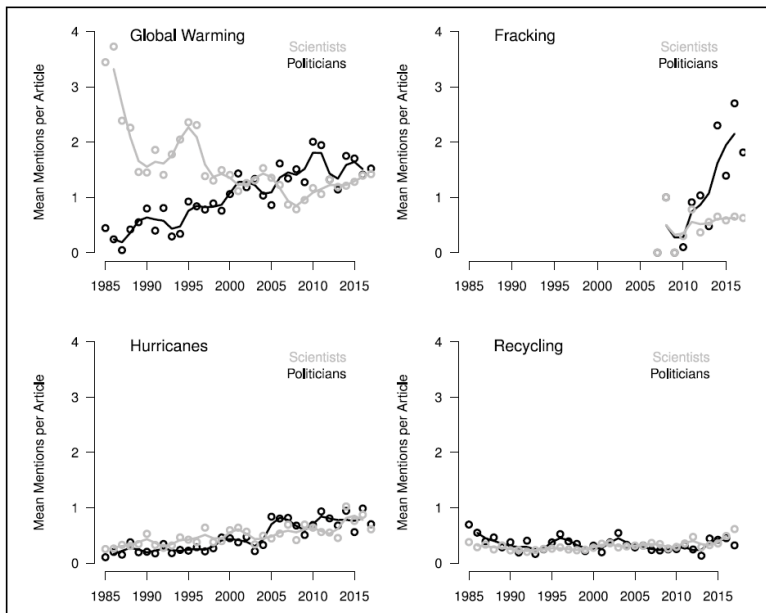


Figure 1. Politicization in news coverage of environmental issues.

[Chinn et al., 2020]

Basic approaches: keyword counting and dictionary methods

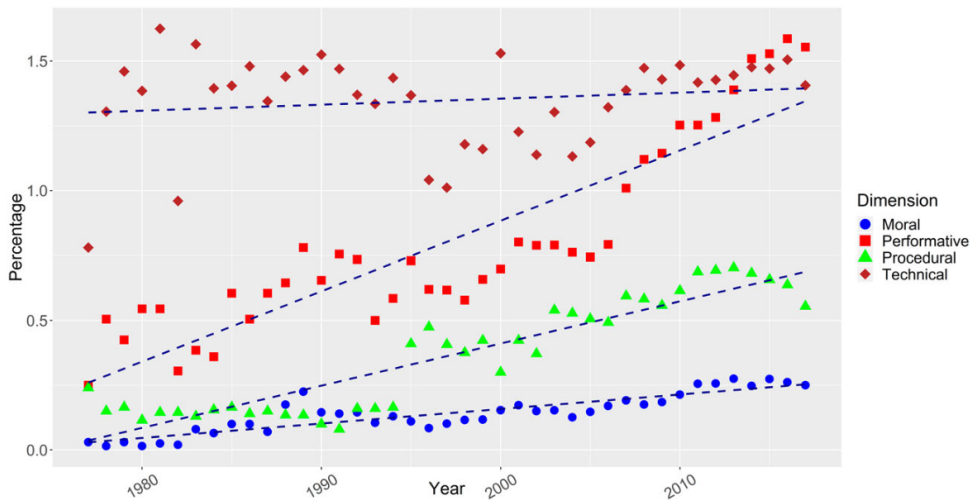


Figure 1. Dimension trends.

[Busuioc and Rimkutė, 2020]



We will practice with:

- o `kwic()`
- o `tokens_select()`
- o `tokens_lookup()`
- o `dictionary()`

We will create a customised R function with `function()`

When lost, cry for `help()`!

Validation of results

Validation of results

Text-as-data methods for classification will *always* enable you to assign a text to a class. But how do we know if it is the correct one?

Validation of results

Text-as-data methods for classification will *always* enable you to assign a text to a class. But how do we know if it is the correct one?

“Validate, validate, validate!” [Grimmer and Stewart, 2013]

Validation of results

Text-as-data methods for classification will *always* enable you to assign a text to a class. But how do we know if it is the correct one?

“Validate, validate, validate!” [Grimmer and Stewart, 2013]

For text-as-data methods, validation essentially means showing that the measures produced using our method actually work, and enable us to make valid social science inferences [Grimmer et al., 2022].

Validation of results

Text-as-data methods for classification will *always* enable you to assign a text to a class. But how do we know if it is the correct one?

“**Validate, validate, validate!**” [Grimmer and Stewart, 2013]

For text-as-data methods, validation essentially means showing that the measures produced using our method actually work, and enable us to make valid social science inferences [Grimmer et al., 2022].

Validity assessments take different forms (and this is true not just for text-as-data methods [Adcock and Collier, 2001]):

- ▶ Does our measure pass the inspection of a subject expert or conform with established general knowledge? (*face validity*)
- ▶ Is our measure appropriate for the way our concept of interest is theorised and understood? (*content validity or fidelity*)
- ▶ Is our measure behaving as we would expect with regard to established set of patterns or obvious hypotheses? (*hypothesis or convergent validity*)

Validation of results

More practical tips:

- ▶ Use the method to replicate a task on data for which we know the correct classification (kind of unavoidable in a supervised context)
- ▶ Read texts assigned to a particular category and assess their coherence (useful in semi or unsupervised settings)
- ▶ Examine other variables or features associated with a specific category

Validation of results

Dictionaries need explicit validation [Grimmer et al., 2022]

- ▶ If “gold-standard” annotation is available for part of the texts, then assess how good is the dictionary at replicating it
- ▶ Do your own coding: produce hand labels for a sample of texts and compare with dictionary classification (but do not cheat! use a separate validation data)
- ▶ Check if your categories correlate with other observable variables measured independently
- ▶ We will cover validation more extensively in the next session



No new commands or functions, just more practice with the ones already introduced.

When lost, cry for `help()`!

Recap

- ▶ Text-as-data methods requires transforming raw texts into numerical data for analysis and inference
- ▶ There are various ways of representing texts numerically, and bag-of-words representations are the most basic
- ▶ Automated text classification is a type of text-as-data tasks that maps tokens to classes or labels
- ▶ Keyword counting and dictionary methods are two simple approaches to detect topics or assign documents into classes, and both rely on bag-of-words representations
- ▶ They use keywords identified by the analyst to map documents to labels (simple and intuitive, but also highly context-dependent and lack nuance)
- ▶ Even if basic and intuitive, like all other methodologies they require careful validation

Next session:

Topics models and machine-learning
algorithms for classification

References I



Adcock, R. and Collier, D. (2001).

Measurement validity: A shared standard for qualitative and quantitative research.
American Political Science Review, 95(3):529–546.



Benoit, K. et al. (2020).

Text as Data: An overview.

The SAGE handbook of research methods in political science and international relations, pages 461–497.



Busuioc, M. and Rimkutė, D. (2020).

The promise of bureaucratic reputation approaches for the EU regulatory state.
Journal of European Public Policy, 27(8):1256–1269.



Chinn, S., Hart, P. S., and Soroka, S. (2020).

Politicization and polarization in climate change news content, 1985–2017.
Science Communication, 42(1):112–129.



Grimmer, J., Roberts, M. E., and Stewart, B. M. (2022).

Text as data: A new framework for machine learning and the social sciences.
Princeton University Press.

References II



Grimmer, J. and Stewart, B. M. (2013).

Text as data: The promise and pitfalls of automatic content analysis methods for political texts.

Political Analysis, 21(3):267–297.



King, G., Lam, P., and Roberts, M. E. (2017).

Computer-assisted keyword and document set discovery from unstructured text.

American Journal of Political Science, 61(4):971–988.



Trubowitz, P. and Watanabe, K. (2021).

The geopolitical threat index: A text-based computational approach to identifying foreign threats.

International Studies Quarterly, 65(3):852–865.



Watanabe, K. and Zhou, Y. (2020).

Theory-driven analysis of large corpora: Semisupervised topic classification of the UN speeches.

Social Science Computer Review, pages 1–21.