Applied Scaling & Classification Techniques in Political Science

Lecture 1

An introduction to the course

(and to text analytics!)

Boring (but needed) information

- Two classes each week starting from today, each Wed (theory)/Thur (lab)
- Office Hour: basically when you want. Plz write me in advance to fix an appointment!
- Email: luigi.curini@unimi.it

Boring (but needed) information

How to evaluate you???

Home-assignments!!!

Boring (but needed) information

- All the slides, scripts and datasets that we employ during our classes (for my part) will be made available the day before each lecture at the following URL:

http://www.luigicurini.com/applied-scaling--classification-techniques-in-political-science2.html

Boring (but needed) information

- You should be familiar with R…and if you are not, you will become!

Boring (but needed) information

This course is aimed to:

- ✓ Introduce you some of the new methods developed within the literature in the last years to analyze texts
- ✓ Offer you guidelines on how to effectively (and practically) use text methods for social scientific research

Boring (but needed) information

- Plan of the course:

1. An introduction to text analytics
2. From words to positions: scaling algorithms (unsupervised & supervised)
3. From words to issues: classification algorithms (unsupervised, semi-supervised & supervised)

So let's start!

Really boring information!

- My name is Luigi Curini
- Full Professor of Political Science at Università degli Studi di Milano
- Mainly interested in party competition, legislative behaviour, quantitative methods applied to political science, social media

References

- ✓ Grimmer, Justin, and Stewart, Brandon M. 2013. Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, 21(3): 267-297
- ✓ Benoit, Kenneth 2020. Text as data: An overview. In Luigi Curini and Robert Franzese (eds.), *SAGE Handbook of Research Methods is Political Science & International Relations*, London, Sage, chapter 26

It all began with…

It is no exaggeration to consider text as "*the most pervasive - and certainly the most persistent artifact of political and social behavior*"

Recognizing that language is central to the study of politics and social science is **not new**…

…however scholars have struggled when using texts to make inferences about politics for example

It all began with…

Why? **Volume matters!** There are simply too many texts out there!

Rarely scholars are able (time/resources constrain!) to manually read all the texts

It all began with…

Recent methods have made progress by breaking from traditional (human) content analysis to treat text:

- not as an *object for subjective interpretation*, but…
- …as *objective data from which information about the author can be estimated…i.e.,* **treating words as data!**

What do we mean by that?

It all began with…

Text is an example of “**unstructured data**”, because it is structured not for the purposes of serving as any form of data but rather structured according to the **rules of language**

Because “data” means, in its simplest form, information collected for use, **text starts to become data** when we record it for reference or analysis, and this process always involves imposing **some abstraction or structure that exist outside the text itself**

It all began with…

Absent the **imposition of this structure**, the text remains **informative** - we can read it and understand what it means - but it does not provide a **form of information**

That is, **treating texts-as-data** means:

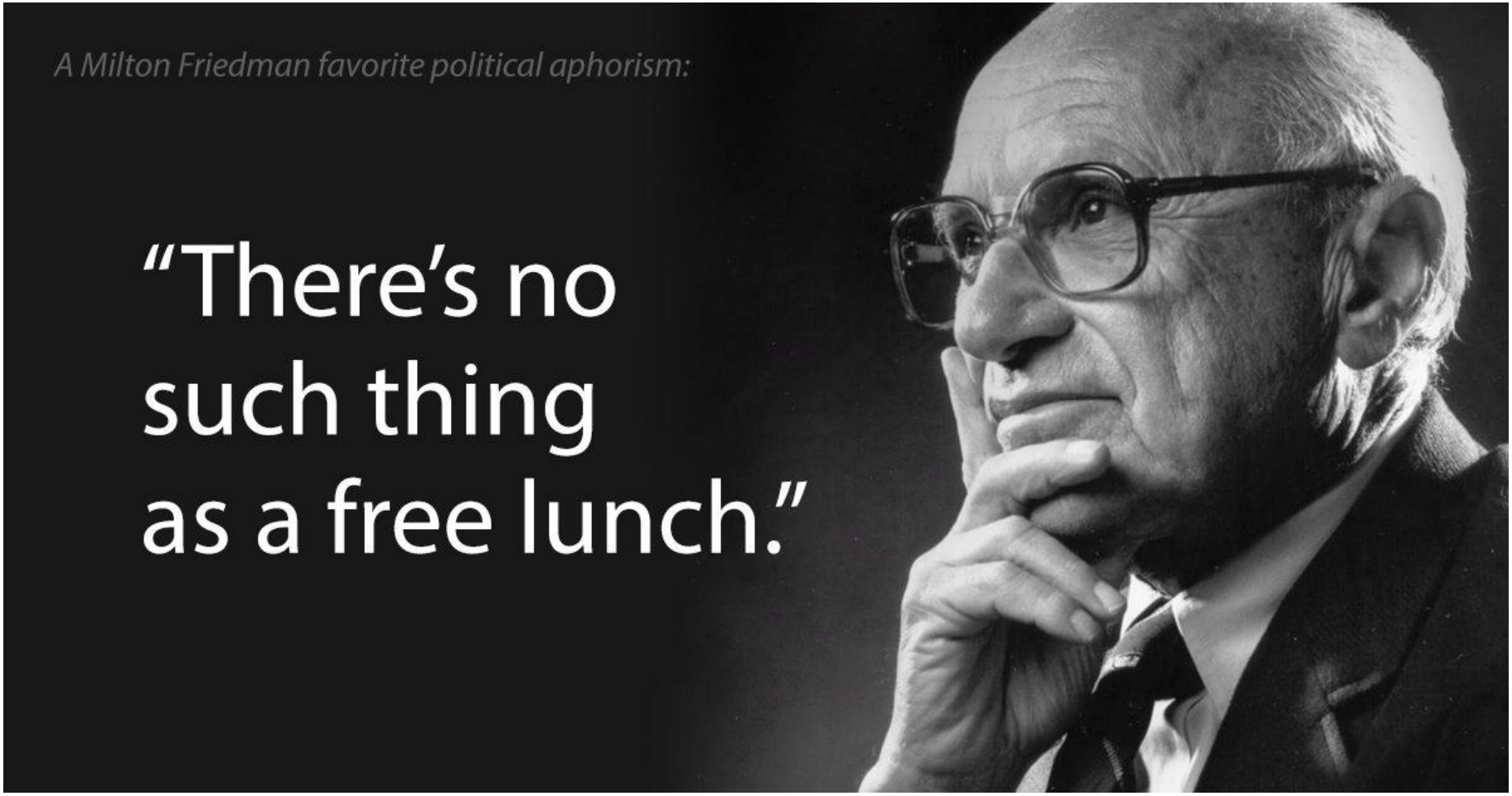
1. arranging texts for the purpose of analysis, using a structure that probably was not part of the process that generated the data itself
2. through that, making texts amenable to the tools of data-analysis

This make possible what was previously impossible: **the systematic analysis** of large-scale text collections that facilitates substantively important inferences from them

It all began with…

The opportunities afforded by vast electronic text archives and algorithms for text analysis are in a real sense unlimited

Yet in a rush to take advantage of the opportunities, it is easy to overlook some important questions and to underappreciate the consequences of some decisions



Just as no body escapes Newton's laws, no technique can escape the following fundamental principles of text analysis

Four principles of Automated Text Analysis to keep in mind (as social scientists!)

1) All quantitative models of language are wrong – but some are useful

The first principle

Data generation process for any text is a **mystery**

If a sentence has complicated structure, its meaning could change drastically with the inclusion of new words (or punctuation…)

The first principle

The Sibyl

"*ibis, redibis, non morieris in bello*"

vs.

"*ibis, redibis non, morieris in bello*"

The first principle

The **complexity of language** implies that all methods necessarily **fail** to provide an **accurate account of the data-generating process** used to produce texts

That all automated methods are based on **incorrect models of language** *therefore* implies that the models **should be evaluated** based on their ability to perform some useful social scientific task

2) Quantitative methods amplify humans, not replace them

The second principle

The **complexity of language** implies that automated content analysis methods will never replace careful and close reading of texts

Rather, such methods are best thought of as **amplifying careful reading and thoughtful analysis**

Researchers still guide the process, make modeling decisions, and interpret the output of the models

«The best technology is **human-empowered** and **computer-assisted**»
(Gary King, Harvard University)

3) There is no a best method for automated text analysis

The third principle

Different datasets and different research questions often lead to different quantities of interest. This is particularly true with text models!

We should simply acknowledging that there are **different research questions** and designs that imply **different types of models**

As a result, every research question and every text-as-data enterprise is **unique**. Analysts should do their own testing to determine how the decisions they are making affect the substance of their conclusions, and be mindful and transparent at all stages in the process

4) Validate, validate, validate

The fourth principle

As told, the **complexity of language** implies that automated content methods are incorrect models of language

This means that the performance of any one method on a new data set cannot be guaranteed, and therefore **validation** is essential when applying automated content methods

We will discuss about validation a lot

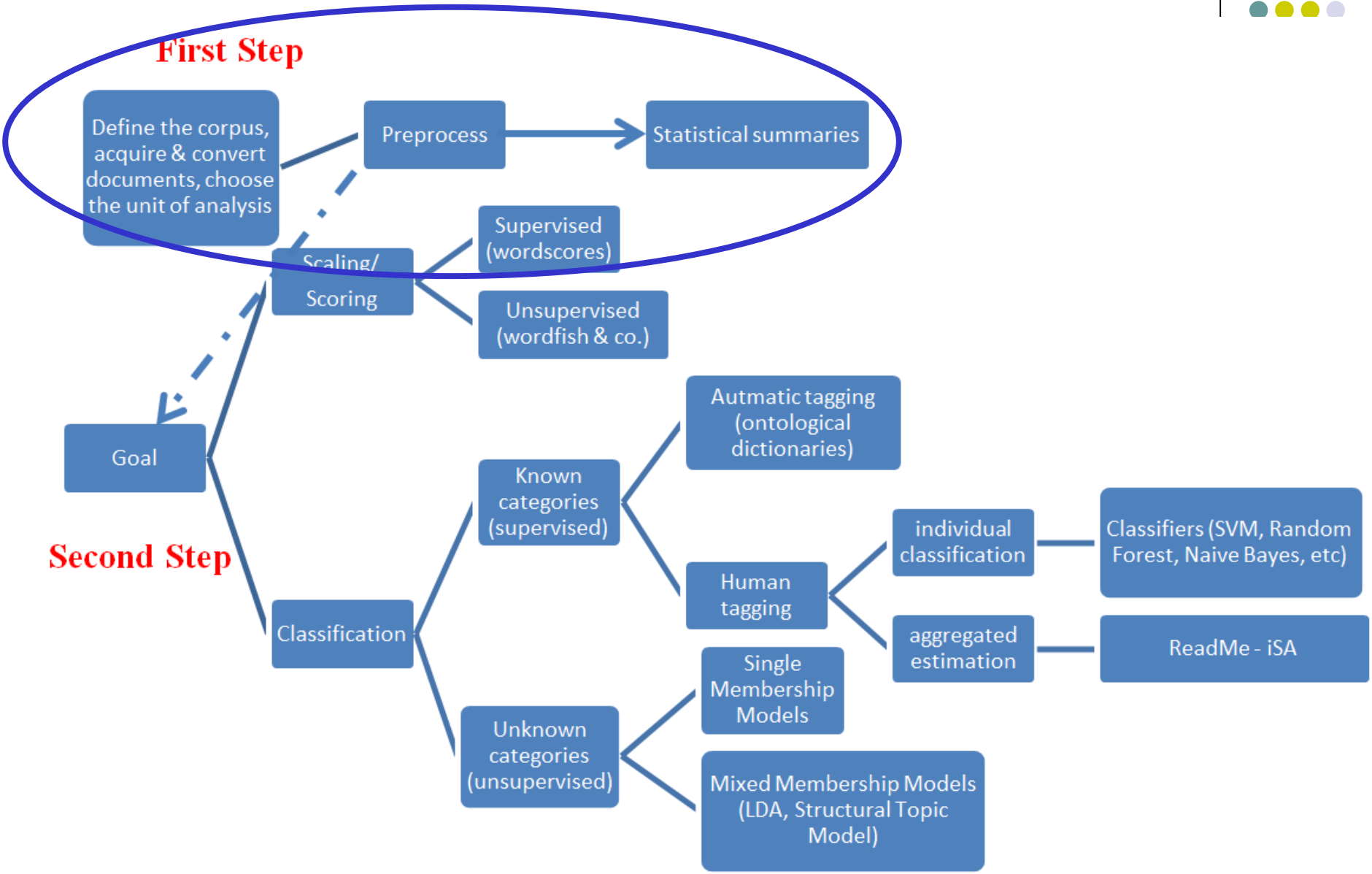
What should be avoided, then, is the **blind use of any method** without a validation step

For analysts using text as data, there are decisions at every turn, and even the ones we assume are benign may have meaningful downstream consequences!!!

Let’s start our journey…

So how to prepare a text for the analysis?

Our Course Map



The First Step: the preparation

Two stages:

1. **Defining the corpus** and the unit of analysis, and then acquiring the texts
2. **Preprocessing stage**: defining and refining textual features (i.e., words) as well as converting them into a quantative matrix

Define the corpus

Jargon: we refer to *text* or *document* as the **unit of analysis** (it could apply to any unit of text: a tweet, a Facebook status, press briefing, sentence, paragraph)

We refer to the population of texts to be analyzed as the *corpus* and a collection of these as *corpora*

Define the corpus

A year of articles about the economy from The New York Times, for instance, could form a **corpus** for analysis, where the **unit** (text or document) of analysis is an article

A set of debates during (one of the many) votes on Brexit in the UK House of Commons could form another **corpus**, where the **unit** of analysis is a speech act (one intervention by a speaker on the floor of parliament)

Acquire the texts

The burst of interest in automated content methods is mainly due to the proliferation of **easy-to-obtain** digital texts

Some of these texts are already available (for example, legislative speeches), others should be recollected by you, by scraping or via API query

Later on we will discuss how to retrieve data from social media (i.e., Twitter, but you can easily employ API via R packages to retrieve data also from Redditt, YouTube & TikTok for example. If you are interest, plz drop me an email!)

Moreover, if you are interest in getting data from Facebook and/or Instagram, you can apply to get a research account from CrowdTangle. And if you are lucky…

greenpass

Share Save

Facebook | Pages | Last 24 Hours | Post Type | Pages | Language | Branded Content | Verified Status | Lists

Local Relevance | Page Category | Page Admin Country | Meme Search

Posts with the most interactions do not equal posts with the most content views or reach.

Interested in seeing content ranked by content views instead of interactions?
Check out the first quarterly Widely Viewed Content Report at Facebook's Transparency Center.

INTERACTIONS
624,410

POSTS
3,022

EXTENDED 7-DAY TIME PERIOD

Showing **25** of **660** public posts since Sep 15, 2021 3:34 PM

SORT BY Total Interactions

Pastorizia Never Dies
Sep 16, 2021 at 8:48 AM

Da Pnd HOT - Paolo

Nessuno

MoVimento 5 Stelle
Sep 15, 2021 at 6:04 PM

Il vaccino salva la vita, non le bugie di chi insegue il consenso

Il vaccino ha salvato milioni di persone. Le ha

Nicola Porro
Sep 15, 2021 at 5:49 PM

#ZuppaDiPorro sul ##greenpass La questione del lasciapassare esteso a tutti continua a non convincermi. Ne faccio innanzitutto una questione di principio, ma

Acquire the texts

As a researcher, when you acquire your corpus you need to ensure that the texts under examination are related to the **research question you are interest about** and have **theoretical consistency**

For example, imagine that you want to retrieve your corpus from Twitter by using a list of keywords

Acquire the texts

In this case you want to generate a *list of keywords* expected to distinguish between tweets relevant to the topic you are interest about (say, *Donald Trump*) compared to irrelevant tweets

It is however critical that the analyst *pay attention* to selecting keywords that are both *relevant to the population of interest* (given the topic you care about) and *representative of the population of interest* (i.e., not being too narrow and selecting only the tweets pro or against Donald Trump via a biased list of keywords)

Acquire the texts

In other words, in our attempt to acquire our corpus, we want **to include** in the corpus all relevant texts (i.e., *minimize false negatives*) and **exclude** any irrelevant texts (i.e., *minimize false positives*)

Convert the texts

The step of converting the texts into a common electronic format is a purely technical one, involving no research design decisions, but it can nonetheless poses one of the stickiest problems in text analysis (pdf as image…)

In R, we will use the `readtext` command in this regard and then the `corpus` command to declare that a set of texts belong to the *same collection* you want to analyze (i.e. to the same corpus)

Preprocessing stage

But then…how to move from words to number? That is:

- *how a text can be transformed into digital data so that an algorithm can then treat it?*

Preprocessing stage

Introducing some terms…

Words as they occur in a text are commonly known as **tokens**, so that the text "*one two one two*" contains four tokens

Tokenization is the process of **splitting a text** into its constituent tokens

In R, we will very often use the `tokens` command in this regard

Tokenization usually happens by recognizing the delimiters between words, which in most languages takes the form of a space

In more technical language, inter-word delimiters are known as **whitespace**, and include additional machine characters such as newlines, tabs, and space variants

Preprocessing stage

However in some major languages, such as Chinese and Japanese, sentences are only distinguished by commas and periods, and words are put in sequence without spaces in between. And so?

Tokenizing these languages requires a *set of rules* to recognize word boundaries, usually from a listing of common word endings

Preprocessing stage

私は、日本社会党を代表して、当面する内外の諸問題につき、佐藤総理大臣にその所見をたださんとするものであります。

↓ after tokenization

私_は_、_日本_社会党_を_代表_し_て_、_当面_する 内外_の_諸問題_に_つき_、_佐藤_総理_大臣_に_その_所見_を_たださん_と_する_もの_で_あり_ます_。

Preprocessing

In R, we will use the `tokens` command to tokenize a text

The nice thing about `tokens` is that it allows to directly tokenize also Japanese/Chinese etc.

Preprocessing stage

To introduce another term, **word types** refer to uniquely occurring words

So that the text "*one two one two*" contains four tokens, but only two word types, "one" and "two"

Preprocessing stage

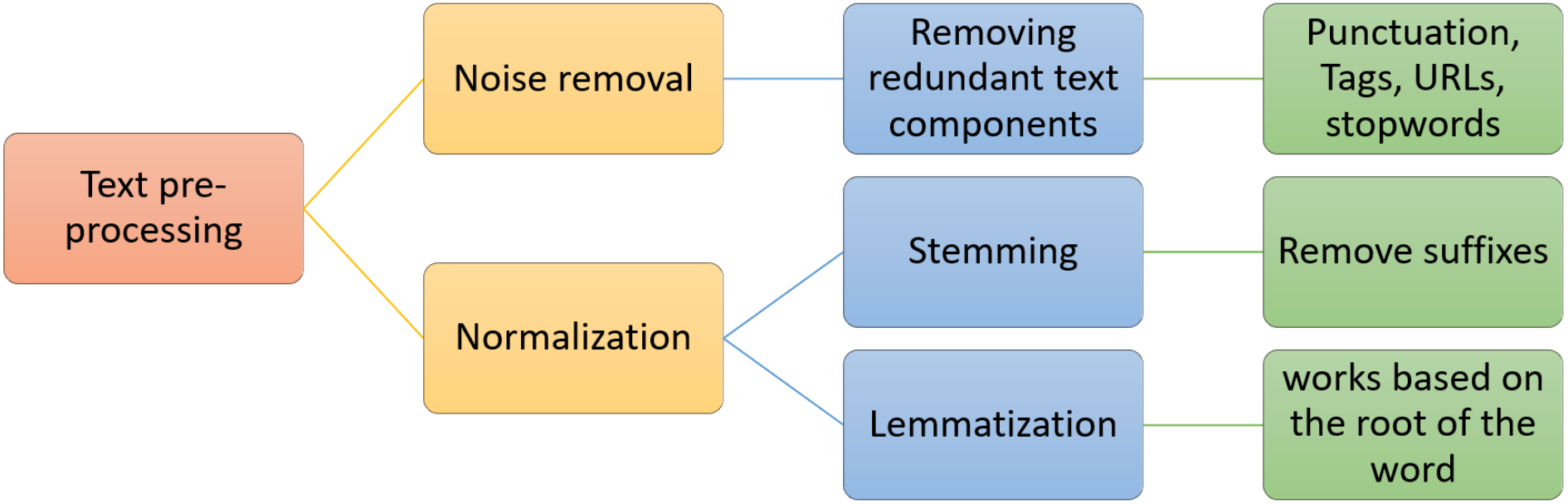
For a **token** to become a **feature** of textual data (our basic unit of analysis), it typically undergoes a process of selection and transformation in a step often called “pre-processing”

Why do we need such process? Cause language is **complex**! But not all of language’s complexity is necessary to effectively analyze texts (REMEMBER?)

We should **retain information** (i.e., tokens) that will be used by the automated methods, **while discarding information** (i.e., tokens) that will likely be unhelpful, ancillary, or too complex for use in a statistical model

Preprocessing stage

Text pre-processing can be divided into two broad categories—**noise removal & normalization**



Preprocessing stage

1. **Noise removal**: Data components that are redundant to the core text analytics can be considered as **noise**

Such as?!?

The First Step: the preparation

Stopwords! They include the large number of prepositions, pronouns, conjunctions etc. in sentences such as *the, is, at, which,* and *on* in English that occur in the greatest frequency in natural language texts

These words can be considered **unlikely** to contribute useful information for analysis, adding little specific political meaning to the text

However…

The First Step: the preparation

…the pronoun “**her”**, as Monroe, Quinn and Colaresi (2008) found, has a decidedly partisan orientation in debates on abortion in the U.S. Senate

For this reason, when preparing textual data for analysis, always check the impact on your final results of dropping stopwords

The First Step: the preparation

We also typically discard:

- **Punctuation**
- **Capitalization**: we apply lower-casing, which treats words as equivalent regardless of how they were capitalised
- We can also decide to eliminate words through the use of **predefined lists of words to be ignored** (for example: tags, URLs, etc.) or based on **their relative infrequency** (words that appear only once or twice in the corpus are unlikely to be discriminating)

The First Step: the preparation

2. **Normalization**: Handling multiple occurrences / representations of the same word is called normalization

There are two types of normalization: **stemming** and **lemmatization**

The First Step: the preparation

Stemming normalizes text by reducing words to their stems, which is a cruder algorithmic means of equating a word with its canonical (dictionary) form, i.e., stemming treats **words as equivalent** when they differ only in their inflected forms

The First Step: the preparation

For example, the different words *taxes, taxation,* and *taxable* are all converted to their word stem "**tax**"

Form	Suffix	Stem
taxes	-es	tax
taxation	-axation	tax
taxable	-able	tax

Stemming of course reduce the total number of tokens in the corpus

The First Step: the preparation

Lemmatization is a more advanced technique which works based on the **root of the word** taking into consideration the **morphological analysis of the words**

To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma

The First Step: the preparation

For example, *runs, running,* and *ran* are all forms of the word *run*, therefore **"run"** is the lemma of all these words

Form	Suffix	Stem
runs	-s	run
running	-ning	run
ran	--	ran

Stemming

Lemmatization

Form	Morphological information	Lemma
runs	Third person, present tense of the verb run	run
running	Present participle of the verb run	run
ran	Past tense of the verb run	run

The First Step: the preparation

Quite often (in particular in all non-positional analyses), **we also discard the order** in which words occur in documents, i.e., we assume that documents are a **bag of words**, where order does not inform our analyses

Is it a problem?

For instance, the expressions '*We are against lowering taxes, and for tax increases*' and '*We are for lowering taxes, and against tax increases*' use the exact same words, even though the meaning is reversed

The First Step: the preparation

While it is easy to construct sample sentences where word order fundamentally changes the nature of the sentence, empirically these sentences are rare

As a result, a simple list of words, which we call **unigrams**, is often sufficient to convey the general meaning of a text

And consistently across applications, scholars have shown that a simple representation of text such as the one we get via a bag-of-words approach is sufficient to infer **substantively interesting properties of texts**!

The First Step: the preparation

We can also *retain some word-order* by including **bigrams** (word pairs, for example to distinguish the “White House” from the color and the domicile) or any other (defined as sequences of n consecutive tokens to form not words but phrases)

In practice, for common tasks, n-grams do little to improve the performance of text analysis

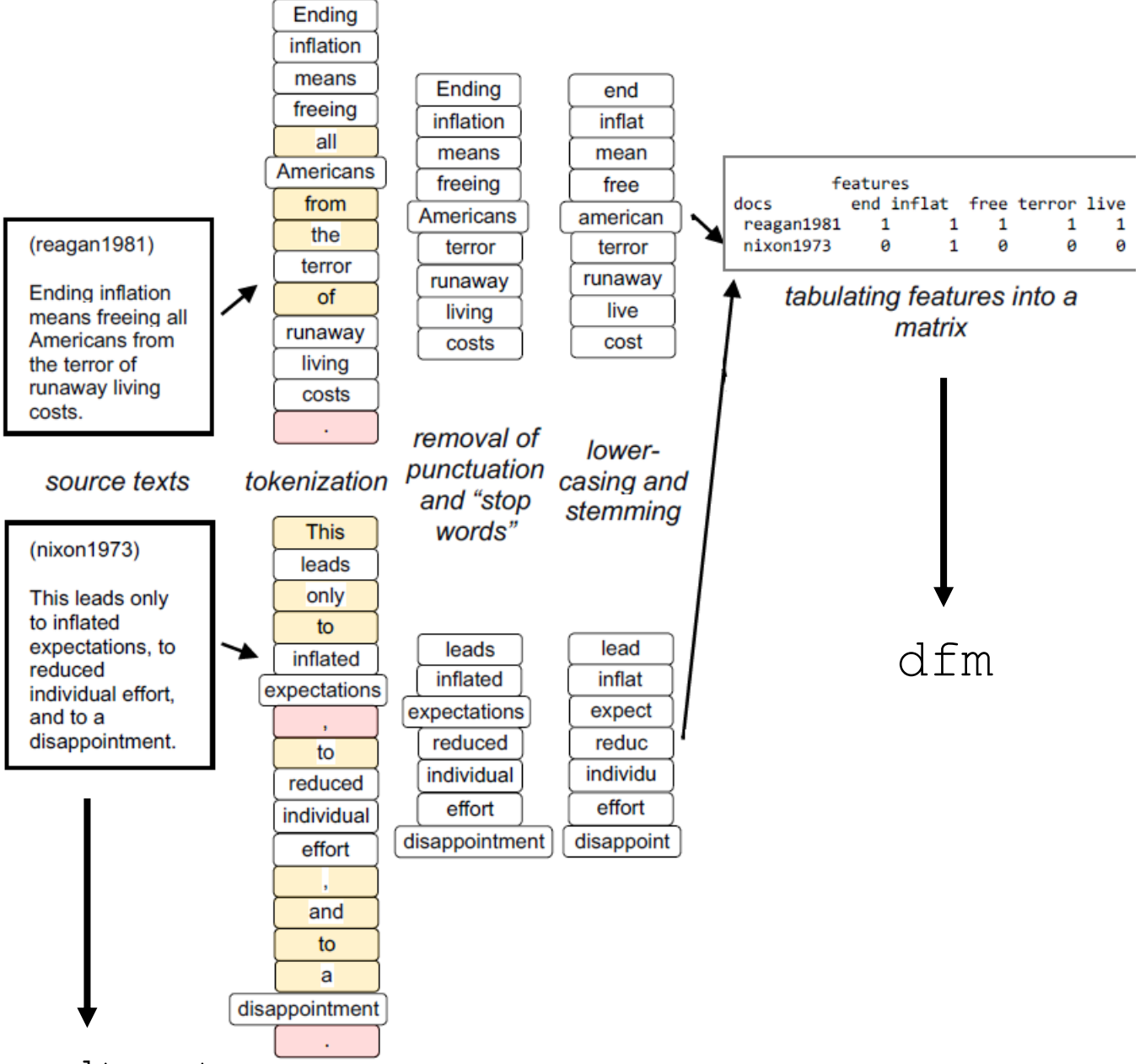
The First Step: the preparation

The result of the *preprocessing steps* is that each document can be represented as a **vector that counts** the number of times each of the unique words occur in each document

This the *bag-of-words* approach!

Multiple document vectors are then put together in a **document-term matrix (or document-feature matrix)**, where each **row** represents a document and each **column** represents a unique word, or term

In R, we will use the `dfm` command in this regard


(reagan1981)
Ending inflation means freeing all Americans from the terror of runaway living costs.
(nixon1973)
This leads only to inflated expectations, to reduced individual effort, and to a disappointment.
source texts
Ending inflation means freeing all Americans from the terror of runaway living costs .
This leads only to inflated expectations , to reduced individual effort , and to a disappointment .
tokenization
Ending inflation means freeing Americans terror runaway living costs
leads inflated expectations reduced individual effort disappointment
removal of punctuation and "stop words"
end inflat mean free american terror runaway live cost
lead inflat expect reduc individu effort disappoint
lower-casing and stemming
features
docs end inflat free terror live
reagan1981 1 1 1 1 1
nixon1973 0 1 0 0 0
tabulating features into a matrix
dfm
readtext & corpus
tokens

The First Step: the preparation

The matching between row and column will report either the frequency of that word in that document (as shown above)….

….or alternatively a list of 0/1: where 0 = word not present in that document and 1 viceversa

This latter procedure is called **one-hot-encoding**

We will mainly deal with the former procedure - but not only: for example, a one-hot-encoding could be advisable given very short texts (such as tweets)

The First Step: the preparation

This matrix form of textual data can then be used as input into a variety of **analytical methods** for describing the texts

Ironically, generating insight from text as data becomes possible **once we have destroyed** our ability to make sense of the texts directly

We should not lose any sleep over it, because the point in analysing text as data is **never to interpret the data but rather to mine it for looking for patterns**

The First Step: the preparation

A bag-of-words approach therefore discards much linguistic information regarding the **surrounding syntactic and semantic context** of a given word in a sentence

Of course, in some contexts bringing back the context in which a word appears, can be very important…

Positional analysis, such as **word embeddings**, allows us to do precisely that (and if have time, we will discuss about such type of analysis at the end of the course)

The First Step: the preparation

DfMs are affected by what is known as the **curse of dimensionality**: new observations tend to grow the feature set, and *each new term found in even one single document adds a new column to the matrix*

This usually creates a **problem of sparsity** in your dfm (a matrix with lots of 0s!) – often a statistical challenge!

$$\begin{bmatrix} 0 & 1 & 5 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 9 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 2 & 0 & 0 & 8 \end{bmatrix}$$

The First Step: the preparation

Several of the pre-processing techniques just discussed allows to minimize precisely the sparsity problems

One further strategy for mitigating the problem of exponentially increasing dimensionality is to **trim** the document-feature(term) matrix

Trimming can be done on various criteria, but usually takes the form of a filter based on some form of feature frequency (i.e., keeping only features that appear just in 10% of documents for example)

The First Step: the preparation

Under some given circumstances, you could also prefer to **weight** your document-feature(term) matrix

Weighting schemes convert a matrix of counts into a matrix of weights

The most common of these is **relative term frequency**, a weighting process also known as document *normalisation* because it homogenises the sum of the counts for each document

Since documents in a typical corpus vary in length, this provides a method for comparing frequencies more directly than counts, which are inflated in longer documents

The First Step: the preparation

Words may also be weighted according to how *rare or frequent* they are in the corpus via a ***tf-idf*** **(term frequency-inverse document frequency)** matrix

tf-idf is a method in information retrieval for down-weighting the terms that are *common* to documents

tf-idf adds a weight that approaches zero as the number of documents in which a term appears (in any frequency) approaches the number of documents in the collection

In texts of debates over **health care**, for instance, tf-idf weighting is likely to eliminate all words related to health care, even when they might occur at very different rates across different documents

The First Step: the preparation

Note that several of the models we will discuss only work with counts as inputs, so that tf-idf or other weighting schemes are inapplicable (but trimming always yes!)

The First Step: the preparation

Never underestimate the *power* of the preprocessing stage!

Preprocessing has tremendous consequences for the quality of automated text analysis

The First Step: the preparation

Denny and Spirling (2018) replicated several published text analyses from political science using a variety of alternative feature processing steps

Their results shows that "*under relatively small perturbations of of preprocessing decisions...very different substantive interpretations would emerge*"

Researchers in practice should be aware of these decisions, critically examine the assumptions of their methods and how these relate to feature selection, and test the robustness of these results

Before our first Lab class

If you **have a laptop** with you:

1) Install the latest version of R
2) For **Windows platforms**: install the latest version of Rtools (i.e., Rtools 4) from here (https://cran.r-project.org/bin/windows/Rtools/)

For **OS X**, do the following:

a) First try to install Quanteda directly
b) If you fail in doing that, install XCode from the App Store

Before our first Lab class

c) To install XCode, follow these simple rules:

1 Access to “Apple Developer”

https://developer.apple.com/download/more/

(You need Apple ID and password)

2 Insert “Xcode” in “Search Downloads” located on the left side of the page.

3 Choose “Xcode 12” and download.

4 After finishing download, click "Finder" and then "download." Double click “Xcode 12”. It may take a while to open this file

d) If you have problems to install the **latest version** of Xcode, **uses an earlier one**, such as Xcode 9!

Before our first Lab class

e) To make things even more complicated for Mac users: the latest R could not be compatible with the most recent Xcode. In that case, they the second most recent version of R

Before our first Lab class

Install the following packages by running these lines (1):

install.packages('quanteda', repos='http://cran.us.r-project.org')

install.packages('quanteda.textstats', repos='http://cran.us.r-project.org')

install.packages('quanteda.textplots', repos='http://cran.us.r-project.org')

install.packages('readtext', repos='http://cran.us.r-project.org')

install.packages('devtools', repos='http://cran.us.r-project.org')

devtools::install_github("quanteda/quanteda.corpora")

devtools::install_github("quanteda/quanteda.textmodels")

Before our first Lab class

Install the following packages by running these lines (2):

install.packages('ggplot2', repos='http://cran.us.r-project.org')

install.packages('SnowballC', repos='http://cran.us.r-project.org')

install.packages('corrplot', repos='http://cran.us.r-project.org')