Big Data Analytics

UNIVERSITÄT

LUZERN

Lecture 1/A An introduction to text analytics



References



- Grossman, Jonathan, and Pedahzur Ami. 2020. Political Science and Big Data: Structured Data, Unstructured Data, and How to Use Them, *Political Science Quarterly*, 135(2): 225-257
- ✓ 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



- The amount of data generated as a by-product in society is growing fast including data from satellites, sensors, transactions, social media and smartphones, just to name a few
- "Every day 2.5 <u>quintillions</u>* of bytes are being created...so much that the 90% of today's available data have been created in the last 2 years" IBM, January 2012
- *=1,000,000,000,000,000,000



Such data are often referred to as Big Data

- Quite interestingly, while the *information overload* is not a recent phenomenon, the term Big Data is relatively new
- It first emerged in the information technology industry in the mid-1990s and made its academic debut in a 1998 computer science paper
- In the two decades that followed, it gained popularity rapidly

But what do you mean by Big Data?



What Big Data are not

Big Data are not just a data collection with a very large-N

That is, a very *large survey of citizen participation* crossnationally is **not**, strictly speaking, Big Data



What Big Data <u>are</u>: 3 main attributes should be present (at the same time!)

- The most common definitions are in fact based on the "three Vs" framework
- 1. volume (the sheer size of the data set is large)
- 2. velocity (data are produced in or almost in real time, i.e., size per unit of time matters)
- *3. variety* (data come in different types and formats and may be structured or, more often, unstructured)
- This last property is a crucial one in terms of the challanges it provides

For most people, the word "data" means this...

	1icrosoft Ex	cel - PS	3_colla	psed_sym	bols_de	sc.t	xt			and the second			
-	Elle Edit R2C1	¥jew	insert fk	Figmat TACC2	Tools	Dat	a <u>Window</u>	Help Adob		e PDF		-	đ ×
10.1	1	aller a		2	3	- 44	4	5	30	6	7	8	-
1	NAME		DESC	RIPTION	786-0		BT-549	CCRF-CE	M	COLO 205	EKVX	HCC-2998	HC
2	TACC2	m marcans	Ina		46	05	82.17	16.	87	98.6	141.03	2 114.32	1
3	C14orf132	Sec. and	na		108	34	59.04	25.	61	33.11	42.5	3 9.12	
4	AGER		na		4	2.2	25.75	76.	01	40.41	32.17	48.28	
5	32385_at		na		7	.43	13.94	8.	55	21.13	15.05	9 19.05	
6	RBM17		na		1	1.4	3	3.	16	2.34	4.4.	3 1.56	
7	DYT1		na		148	09	317.17	316.	66	147.23	125.70	3 261.39	
8	CORO1A		na		8	62	9.12	1572	53	5.91	5.3	1 11.98	
9	WT1		na		206	74	136.71	141.	34	129.09	138.01	1 138.16	
10	SYCP2		na		7	.94	35.68	7	8.	1.97	7.75	5 4.73	
11	SULF1		na		10	45	8.5	4.	05	4.77	2.3	5 3.72	1
12	C19orf21		na		6	22	5.16	3.	95	37.56	110.36	5 208.29	
13	PHYH		na		209	99	253.07	90.	36	61.83	360.49	145.01	
14	31336_at		na		3	35	5.28	2	98	4.82	4.3	5 1.45	-
14	(+) H\P53_collapsed_symbols_desc/										1000		IL
Rea	dy										NU	4	1

These are structured data - data that can fit squarely into a table, where every row is an observation, every column a variable, and the cells at the intersection of rows and columns contain values





Data of this kind, however, are but a fraction of the total amount of data in the world

According to different estimates, 80 to 95 percent of existing data are unstructured data, that is, data that cannot fit easily snugly into rows and columns

Unstructured data may take the form of text, audio, video, or any other observable manifestation (usually in a digital format)



- Because of the messy and eclectic nature of unstructured data, attempts to directly investigate them with conventional statistical methods would often be futile
- And indeed, we use the expression *Big Data* and not *Big Information*, because there is a lot of work for analysts before information can be gained from such data
- To analyze such data in a quantitative way, one needs first of all to impose a **structure** upon them first
- "In analytics there is no such thing as unstructured data, just data that structure has not yet been applied to"



Texts are probably the most common source of Big Data in social science

And it is precisely on them (that is on text-as-data approach) that we are going to focus in our course



Summing up

The Big Data revolution is both a blessing and a curse

- Not only is the volume of data overwhelming, but data must be treated properly
- **Finding the evidence** that we need in an ocean of information is in other words a constant challenge





Having that in mind...either we do like the ostrich...i.e., we can simply ignore the amount of new data currently available for our research!





Or...we can do like Galileo with his telescope: finding **new** patterns in **new** data, with **new** methods (telescope?) available, and, in the best scenario, developing **new** theories thanks to that! After all, the telescope came before Galileo's astronomical theories...





So let's keep this in mind! And let's start our journey in textanalytics!





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



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

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



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?



Text is an example of what we called earlier "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**



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:

- 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 dataanalysis

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



- 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

A Milton Friedman favorite political aphorism:

"There's no such thing as a free lunch."

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 humanempowered 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, 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?



<u>Our Course Map</u>



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)



- 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 to retrieve data also from YouTube & TikTok for example)



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



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)



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...)



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?

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
- 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



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



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_。



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"



For a **token/type** 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 "preprocessing"

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



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





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



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 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

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)

- 2. Normalization: Handling multiple occurrences / representations of the same word is called normalization
- There are two types of normalization: **stemming** and **lemmatization**



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

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





By doing that, stemming reduce the total number of tokens in the data set



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



Form	Suffix	Stem	
studi <mark>es</mark>	-es	study	
study <mark>ing</mark>	-ing	study	> Stemming
niñ <mark>as</mark>	-as	niñ	
niñ <mark>ez</mark>	-ez	niñ	



Form	Morphological information	Lemma
	Third person, singular number, present tense of	
studies	the verb <mark>study</mark>	study
studying	Gerund of the verb <mark>study</mark>	study
niñas	Feminine gender, plural number of the noun niño	niño
niñez	Singular number of the noun niñez	niñez



In our analysis, 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



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!



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 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



- 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)

- 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 is only 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**



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 challange!

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



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)



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



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



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!)



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

Statistical summaries



Once you have your DfM, you can start by running some statistical summaries

Statistical summary methods are essentially quantitative summaries of texts to describe their characteristics on some indicator, and may use (or not) statistical methods based on sampling theory for comparison

We will present some of them in few minutes