Machine Learning and Programming in Python

Lecture for Master and PhD students

Chair of Data Science in Economics

Ruhr University Bochum

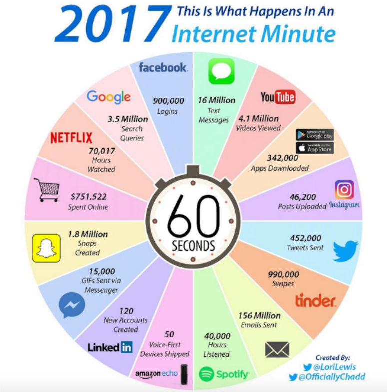
Winter semester 2023/24

Lecture 13
Study materials from 15.12.2023

Natural Language Processing/ Text data analysis

- Natural Language Processing, abbreviated NLP
- The manipulation of human languages by computers
- Includes both text and speech

- Vast amounts of text and speech recorded
  - Historical data (such as speeches, parliamentary records, reports, books, court records, radio and tv programmes)
  - Evolving data (such as web-based text, social media, twitter, email, google searches
  - Large amounts of information inherent in these texts


2017 This Is What Happens In An Internet Minute
facebook 900,000 Logins
16 Million Text Messages
YouTube 4.1 Million Videos Viewed
342,000 Apps Downloaded
46,200 Posts Uploaded Instagram
452,000 Tweets Sent
990,000 Swipes tinder
156 Million Emails Sent
40,000 Hours Listened Spotify
50 Voice-First Devices Shipped amazon echo
120 New Accounts Created LinkedIn
15,000 GIFs Sent via Messenger
1.8 Million Snaps Created
$751,522 Spent Online
NETFLIX 70,017 Hours Watched
Google 3.5 Million Search Queries
60 SECONDS
Created By: @LoriLewis @OfficiallyChadd


2021 This Is What Happens In An Internet Minute
facebook 1.4 Million Scrolling
21.1 Million Texts Sent
YouTube 500 Hours Content Uploaded
414,764 Apps Downloaded
695,000 Stories Shared
200,000 People Tweeting
2 Million Swipes tinder
197.6 Million Emails Sent
2 Million Views twitch
5,000 Downloads Tik Tok
932 Smart Audio Devices Shipped amazon echo
3 Million Images Viewed imgur
69 Million Messages Sent
3.4 Million Snaps Created
$1.6 Million Spent Online
NETFLIX 28,000 Subscribers Watching
LinkedIn 9,132 Connections Made
60 SECONDS
Created By: @LoriLewis @OfficiallyChadd

Applications of Natural Language Processing

- Information retrieval, retrieve documents that are relevant to a query or research, web-based, newspapers, parliamentary records, etc.
- Translation, translate documents from one language to another
- User content moderation, filter inappropriate content Email/Spam, etc.
- Automatic recommendation, targeted advertising, Amazon, Netflix, etc.
- Chatbots and virtual assistants, ChatGPT, Siri, Alexa, etc.

Difference between numerical data and Big Data

- In Econometrics we are used to using numerical data
- These data typically come from survey data and have a relatively orderly structure (i.e. we have a cross section sample of individuals and measures of their wage and demographic characteristics
- Big Data: high-dimensional numerical data; high-dimensional, unstructured text data

- Development in technologies in recent years have made available vast amounts of digital text that is increasingly used in social science research
- All can be used as a rich complement to more standard data
- For example:
  - Text from social media and newspapers has been used to measure policy uncertainty (Baker, Bloom, Davis, 2016, Quarterly Journal of Economics)
  - Transcripts of politicians speeches has been used to study political slant in the media (Gentzkow and Shapiro, 2010, Econometrica)

Baker, Bloom and Davis 2016 - Measuring Economic Policy Uncertainty

- Scrape newspaper text searching for certain phrases to measure economic policy uncertainty
- Build an index of uncertainty over time for the United States
- Add other countries to develop a worldwide index
- Show correlations with "real" economic outcomes over time and across states of the US

Gentzkow and Shapiro 2010 - What drives media slant?

- Examine text from us newspapers to ascertain the political slant of the paper
- Search for certain words or phrases to indicate left/right balance of newspaper
- Build a dictionary of these words by analyzing speeches by politicians
- Categorize politicians as left/right wing according to their voting record
- Then find common words used by left wing versus right wing politicians

TABLE I

MOST PARTISAN PHRASES FROM THE 2005 *CONGRESSIONAL RECORD*[a]

Panel A: Phrases Used More Often by Democrats

Two-Word Phrases		
private accounts	Rosa Parks	workers rights
trade agreement	President budget	poor people
American people	Republican party	Republican leader
tax breaks	change the rules	Arctic refuge
trade deficit	minimum wage	cut funding
oil companies	budget deficit	American workers
credit card	Republican senators	living in poverty
nuclear option	privatization plan	Senate Republicans
war in Iraq	wildlife refuge	fuel efficiency
middle class	card companies	national wildlife
Three-Word Phrases		
veterans health care	corporation for public broadcasting	cut health care
congressional black caucus		civil rights movement
VA health care	additional tax cuts	cuts to child support
billion in tax cuts	pay for tax cuts	drilling in the Arctic National
credit card companies	tax cuts for people	victims of gun violence
security trust fund	oil and gas companies	solvency of social security
social security trust	prescription drug bill	Voting Rights Act
privatize social security	caliber sniper rifles	war in Iraq and Afghanistan
American free trade	increase in the minimum wage	civil rights protections
central American free	system of checks and balances	credit card debt
	middle class families	

TABLE I—*Continued*

Panel B: Phrases Used More Often by Republicans		
Two-Word Phrases		
stem cell	personal accounts	retirement accounts
natural gas	Saddam Hussein	government spending
death tax	pass the bill	national forest
illegal aliens	private property	minority leader
class action	border security	urge support
war on terror	President announces	cell lines
embryonic stem	human life	cord blood
tax relief	Chief Justice	action lawsuits
illegal immigration	human embryos	economic growth
date the time	increase taxes	food program
Three-Word Phrases		
embryonic stem cell	Circuit Court of Appeals	Tongass national forest
hate crimes legislation	death tax repeal	pluripotent stem cells
adult stem cells	housing and urban affairs	Supreme Court of Texas
oil for food program	million jobs created	Justice Priscilla Owen
personal retirement accounts	national flood insurance	Justice Janice Rogers
energy and natural resources	oil for food scandal	American Bar Association
global war on terror	private property rights	growth and job creation
hate crimes law	temporary worker program	natural gas natural
change hearts and minds	class action reform	Grand Ole Opry
global war on terrorism	Chief Justice Rehnquist	reform social security

Text as Data

- The first challenge we face in using text in our quantitative models is that text is very clearly not quantitative data
- As such we need to find ways to convert text for use in such models
- Text is very high dimensional
- Places restrictions on how text can be used as quantitative data

- Imagine a sentence with 15 words. Will yield a very large number of possible combinations of words in this sentence ($15! = 1307674368000$, if each word is unique)

- In how many ways can the letters in the word MISSISSIPPI be arranged? From the number of letters (1x M / 4x I / 4x S / 2x P) follows: $\frac{11!}{1!4!4!2!} = 34650 \Rightarrow$ There are 34650 potential combinations to arrange the letters of the word MISSISSIPPI

Does complexity matter?

- "Time flies like a bird."
- "Fruit flies like a banana."

- The complexity in this sentence is crucial to inferring the meaning

- Quantitative techniques make simplifications that map the text to a numerical array that in general ignores the complexity and interdependence between words
- For example we might just count up the occurrence of key words ⇒ Flies 2 like 2 a 2 bird 1 fruit 1 banana 1 time 1

The steps to quantitative text analysis

- 1. We seek to represent the raw text T as a numerical array X (a matrix or vector)
- 2. We then seek to map X to the predicted values $\hat{y}$ of some unknown outcome y
- 3. We then use $\hat{y}$ in some descriptive or causal analysis

Step 1: Dimension reduction/ pre-processing

- Here we impose some restrictions to reduce the dimensions of the text T to be more manageable
- In mapping from text T to a numerical form X it is common to carry out a range of pre-processing steps
- These commonly involve dropping punctuation, very common or very uncommon words, "stop" words (a, and, the, etc.), names, numbers.
- Counting of words or counts of combinations of words or n-grams

Step 2 Prediction of $\hat{y}$

- In this step we use our high dimensional Machine Learning methods to predict $\hat{y}$ given our numerical representation of the text X
- For example, email spam filters will use the text to predict whether the email should be classified as spam or not
- In this step we are less interested in the relationship between $\hat{y}$ and X but just that X can predict $\hat{y}$
- In the spam email example we don't really care why certain words are more likely to appear in a spam email, just that they help us predict whether an email is spam or not.
- Here one can use the usual Machine Learning techniques, but also non-Machine Learning type approaches, e.g. dictionary methods

Step 3 Causal inference

- In many applications of text analysis the prediction step is the goal
- However, increasingly in the social sciences the outcome measures can be taken a step further and used in a causal analysis

Some definitions

- A Corpus is a collection or body of structured text, i.e. a collection of Facebook posts, a collection of newspaper reports
- A Document is a unit of the corpus; how we define a document will be specific to the research question. i.e. to detect spam emails we might define each document to be an email
- A Token is any word in the document

Pre-processing

- In order to generate something more manageable from our corpus a common first step is the strip away from the raw text anything other than words.
- Remove punctuation, make lowercase, remove names and any HTML tags, symbols (#) etc.
- Remove very common or "stop words"
- Stop words might be important to the grammatical structure of the text but generally convey little meaning
- E.g. the frequency of the word "a" is not likely to be diagnostic of whether an email is spam or not
- Again this step will depend on the research task to hand
- For example in author identification the stop word count might contain important information

Stop words

- For various lists of stop words in different languages see: https://www.ranks.nl/stopwords
- In Python the module we will be using NLTK has a dictionary of stop words

Stopwords from NLTK

- ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

Stemming words

- It is also common to stem words or take them back to their root
- For example, economic , economics , economically replaced by the stem economic
- This reduces further the number of words in a document but retains the basic concept in each word
- The Porter (1980) stemmer is the standard tool used in English and this is present in Python's NLTK
- Essentially removes word endings; ly , ed, ing , ment
- The stemmer can be wrong: E.g. policy and police have the same stem but clearly very different meanings

Stemming

- A few examples from *Pride and Prejudice* (using NLTK)

affect	affect affectation affected affecting affection affections affects
amus	amuse amused amusement amusements amusing
close	close closed closely closing
grate	grate grateful gratefully

- Stopwords removal eliminates very common words that we think add little meaning to the text (and, the)
- It is also common to remove rare or infrequent words
- Rare words can add meaning but the added computational costs from including rare words often exceeds their diagnostic value
- One approach to this is to filter both very rare and very common words using "term frequency inverse document frequency (Tf-idf)"

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Figure 6.2 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

- Here a document is represented as a vector of word counts
- The matrix below reports frequencies of the words (each row) in each document (each column)

Term frequency

- Here we count the frequency of word j in document i:tf_{ij}
- More frequent terms may be more important and contain more information about a document
- But high frequency words (a, and, the) are not always discriminating
- E.g. "Good" appears regularly in all of the plays above

Inverse Document frequency

- Terms or words which occur in fewer documents may be more important
- The term "battle" is unevenly distributed across the plays
- Inverse document frequency: $idf_j = log(\frac{N}{d_j})$
- d_j is the number of documents where term occurs
- N is the total number of documents

Term frequency inverse document frequency

- tf-idf= $tf_{ij} * idf_j$
- Very rare words will have low tf-idf scores because tf_{ij} will be low
- Very common words that appear in most or all documents will have low tf-idf scores because idf_j will be low
- Better than excluding words that occur frequently because it will keep words that occur frequently in some documents but do not appear in others
- these often provide useful information
- Common practice to keep only the words within each document i with tf-idf scores above some cutoff

Jurafsky, D.; Martin, J. (2023), Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition