POLITECNICO DI MILANO

Introduction to NLP

Natural Language Processing

Some slide content based on textbook:
Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition *by Daniel Jurafsky and James H. Martin*
Other content adapted from slides by Roberto Tedesco

Welcome to Natural Language Processing!

Who is Teaching the NLP class?

Lecturer:

Mark Carman

- Associate Professor in DEIB (Dipartimento di Elettronica, Informazione e Bioingegneria)
- mark.carman@polimi.it

🤔 Let's hope his teaching is better than his cooking ...🤣

Research background:

- Information Retrieval & **statistical** Natural Language Processing**
- Machine Learning & Data Science
- Applications: *Personalisation & Recommendation, Web Search, Social Media Analysis, Digital Forensics, Bioinformatics, …*
  - see: https://scholar.google.com/citations?user=fcPONTQAAAAJ&hl=en

Teaching:

- Data Science, Artificial Intelligence & NLP

➢ classes are MUCH more fun when they're interactive, so please **help me out** by asking **lots of questions**!

** Favourite NLP quote:
"Every time I fire a linguist, the performance of the speech recognizer goes up" Frederick Jelinek

Who is Teaching the NLP class?

** Real coders wear t-shirts 😉

Instructor:

Nicolò Brunello,

- PhD student in DEIB (Dipartimento di Elettronica, Informazione e Bioingegneria
- nicolo.brunello@polimi.it

Research background:

- Large Language Modeling (LLM), Retrieval Augmented Generative (RAG) models, eXplainable AI (XAI), Bioinformatics
  - see: https://scholar.google.com/citations?user=wgnP67kAAAAJ&hl=en

OK, so you know who we are …

And who are you?

What is your background?

- Engineering?
- Computer Science?
- Statistics?

How much do you already know about

- Machine learning?
- Deep learning?
- NLP?

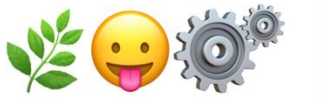
Fill in a quick quiz for me and we'll find out:

- https://forms.office.com/e/rWqJVhLPZ2

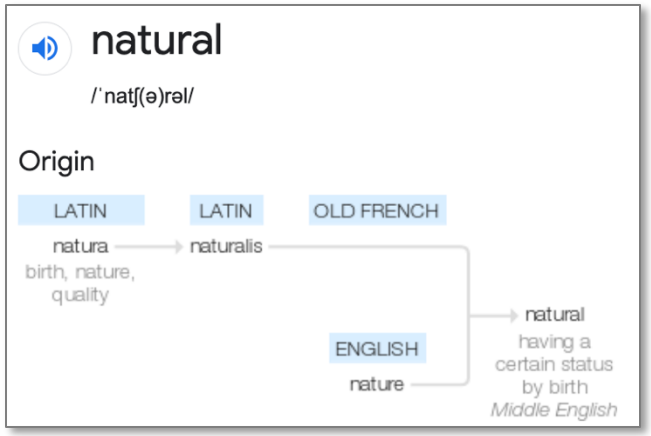
Image source: https://www.viasarfatti25.unibocconi.eu/notizia.php?idArt=19895

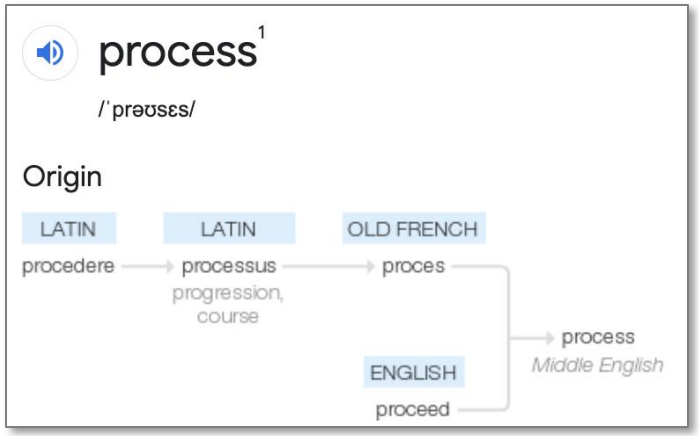
What is Natural Language Processing?

What is Natural Language Processing?



I looked up the etymology (origin) of the words *natural*, *language* and *processing*





- turns out they derive from: *birth*, *tongue*, and *progress*

What's that got to do with NLP?

- not a lot …
- but it does tell us that natural languages are **spoken** and that meaning of words **evolves** over time
- and it also explains the logo for the course ;-)

In this course we will learn how to process natural (human) language

What is Natural Language?

Lots of natural **animal languages** out there!

- see article "**Q&A: What is human language, when did it evolve and why should we care?**": https://bmcbiol.biomedcentral.com/articles/10.1186/s12915-017-0405-3#

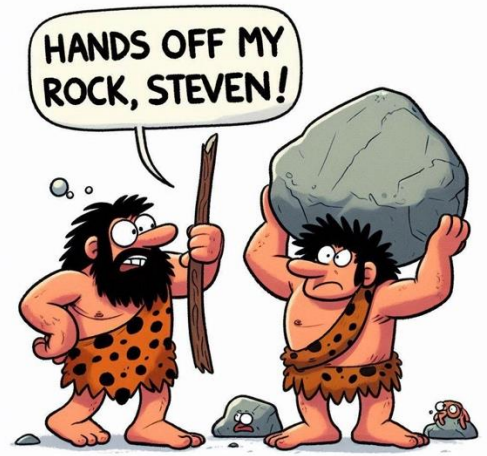
Some of these languages are quite sophisticated:

- monkeys use distinct alarm calls to identify different threats, like snakes, big cats, etc.
- dolphins have sounds associated with hunting & social activities
  - according to a statistical analysis, dolphins have a vocabulary of 125 different whistles: https://www.insidescience.org/news/information-theory-counts-size-animal-languages
- birds sing to communicate
  - parrots can even mimic human sounds

Forms of animal communication are **symbolic**:

- use a particular sound to represent an object or action
- no evidence for **compositionality**

Development of Human language



Source: https://sl.bing.net/dZm9hHi0Mwu

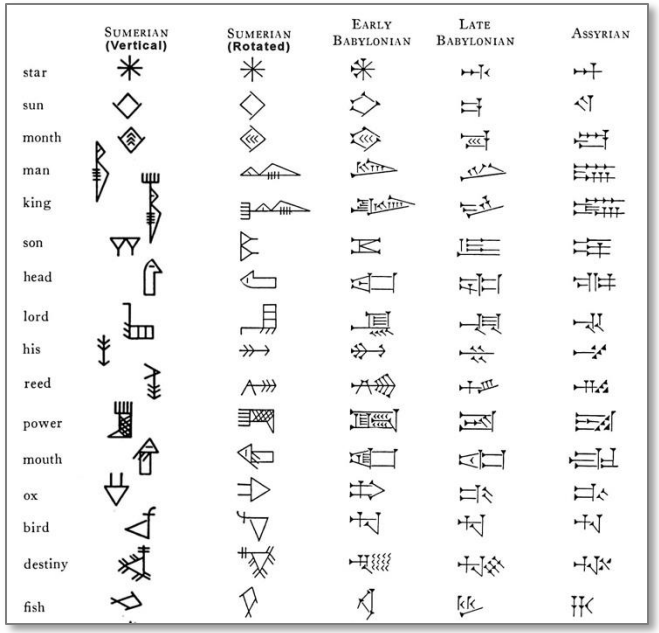
Debate as to when **spoken language** developed

- from as much as 2.5 million to as little as 60 thousand years ago
- depending of course on what you consider **human language** to be
- see: https://en.wikipedia.org/wiki/Origin_of_language

Origin of **written language** is more clear

- first writing systems developed in Mesopotamia (Iraq) around 3500 BCE
- over time writing progressed from **pictograms** representing **objects** to **abstract symbols** representing **sounds**
- see https://en.wikipedia.org/wiki/Cuneiform

Early pictographic writing (3500 BCE)



Progressive simplification from pictograms to abstract symbols

Images source: https://en.wikipedia.org/wiki/Cuneiform

What is so special about *human* language?

Human language is ***compositional***

- express thoughts in sentences comprising **subjects**, **verbs** and **objects**
  - e.g. <I> <walk> <the dog>
- endless capacity for generating new sentences:
  - e.g. 100 words for each role, results in a million distinct sentences

Human language is ***referential***

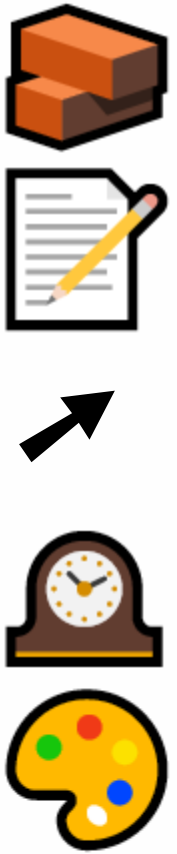
- can express information about objects and their locations/actions

Human language conveys ***temporal*** information

- with **past**, **present** and **future** tenses

Human language is ***varied***

- thousands of different languages spoken around the world



Source: Mark Pagel's article "**Q&A: What is human language, when did it evolve and why should we care?**": https://bmcbiol.biomedcentral.com/articles/10.1186/s12915-017-0405-3#

Text Data

NLP mainly deals with Text data

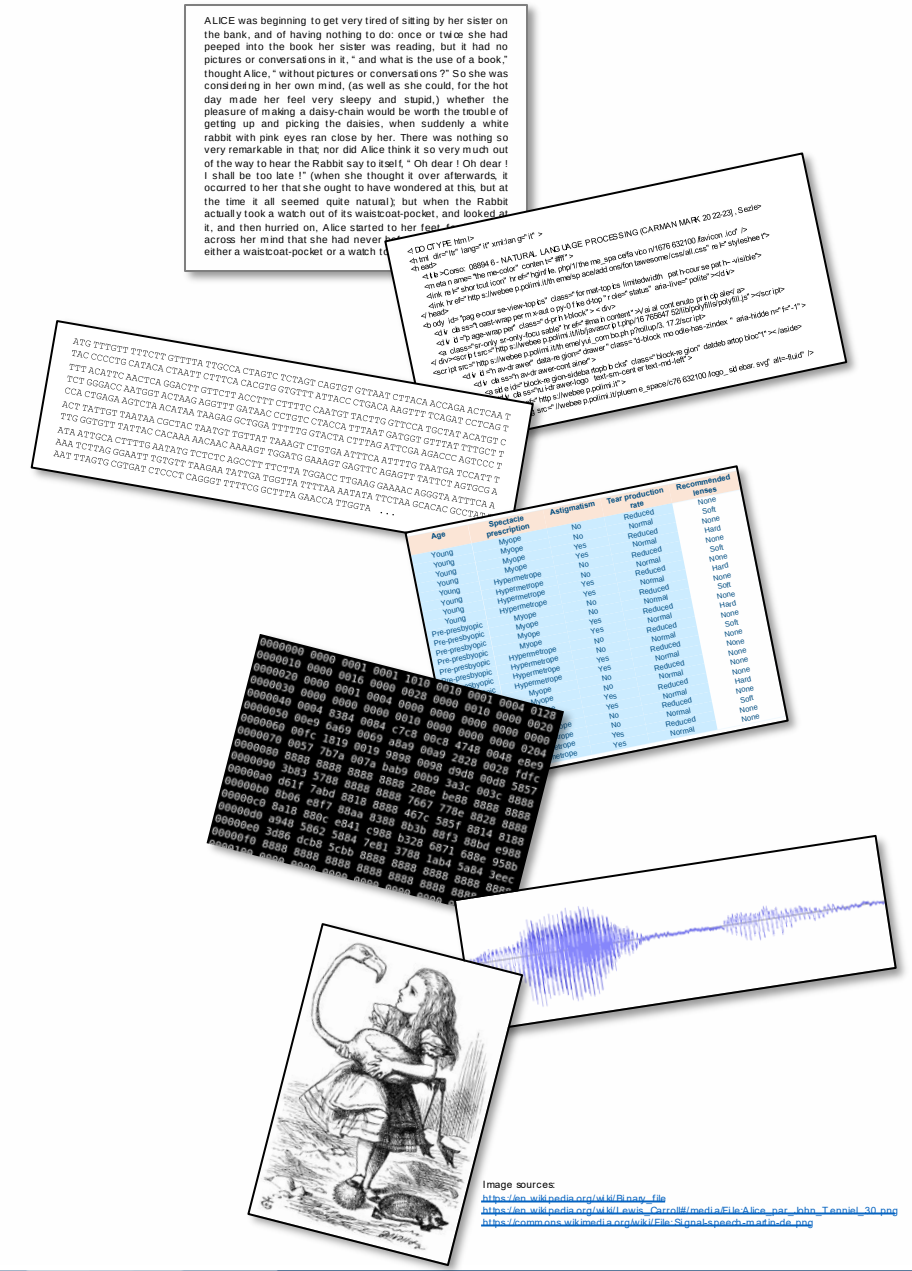
In addition to:

- **textual documents** (in natural language)

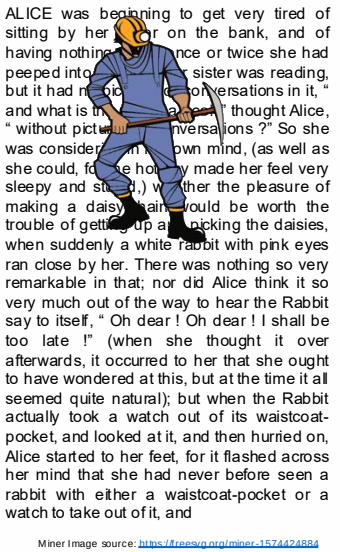
There are many different types of data that we might be interested in working with:

- semi-structured data (like html)
- programming code
- tabular (relational) data
- biological sequences (e.g. genomic data)
- binary data (e.g. malware executables)
- audio data (e.g. speech signals)
- other time series
- images & video

Turns out that **NLP techniques** are useful for handling these other types of data too



What can we do with text?



Miner Image source: https://freesvg.org/miner-1574424884

Natural Language Processing (NLP), aka **Text Mining**

- process of **working with** & **extracting useful knowledge** from **textual data**

LOTS of different things we can do with text data:

cluster *translate* *generate*

visualise

summarise *transcribe*

discover relationships

fact-check

extract entities *identify sentiment* *search*

categorise

disambiguate

correct

attribute authorship

Machine learning and in particular **deep learning** techniques have

- **revolutionised performance** on *all these tasks* over last few years

Example: Text generation has become so good!

So good in fact, that I considered using it to generate a recent talk … 🤣😂

One of the following texts abstracts was written by me

- the other was **generated automatically** by conditioning on the title

Workshop Title: How Deep Neural Techniques have revolutionised Information Retrieval and Text Mining

Abstract: The world of information retrieval and text mining is a global one, but the applications of deep learning have become highly specific and specialized. There are huge opportunities in this field. We examine how deep learning have revolutionized the problem of extracting and processing text. We demonstrate the importance of two major breakthroughs in the field: 1) The first, deep convolutional neural network (DNN), which can find hidden layer effects using very sparse input and a few simple operators. It is widely used for word embeddings and natural language processing. 2) The second, reinforcement learning, which can learn the hidden layer effect using a small amount of input. This technique is also widely used for image and speech recognition.

Workshop Title: How Deep Neural Techniques have revolutionised Information Retrieval and Text Mining

Abstract: Over the last few years, deep neural architectures have rewritten the rulebook in terms of the performance that can be achieved across a multitude of text processing tasks from sentiment analysis and sarcasm detection, to machine translation, web search, question answering, and dialog generation. In this workshop I will explain the language modelling technology behind these advances, discussing its evolution from shallow embeddings to modern transformer models composed of ever deeper self-attention networks. I will describe numerous applications of these deep models in information retrieval and text mining and then look to the future, to applications that seamlessly combine information across text and image modalities.

Can you tell which is which?

- if you guessed the first abstract was automatically generated, you were right ;-)
- try the same GPT-2 based text generator here: https://transformer.huggingface.co/

OK, so what is this course about then?

Planned Course Content**

Natural Language Processing (NLP) concerns:

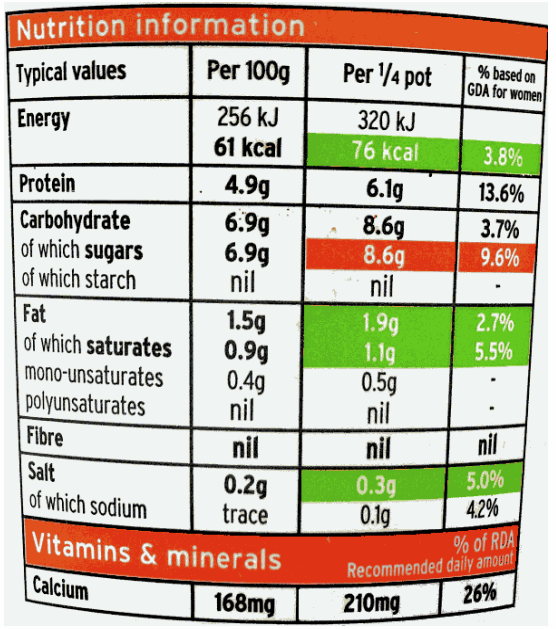
- the **computational analysis**, **interpretation**, and **production** of **natural language** in either written or spoken form

NLP Techniques we aim to cover in this course:

- regular expressions,
- vector space representations & text classification,
- text retrieval and text clustering,
- word embedding based representations,
- language models for text generation
- sequence-to-sequence models and Transformers
- dialog systems: task-oriented and retrieval-augmented chatbots
- Large Language Models (LLMs)
- audio aspects: speech-to-text and text-to-speech

MANY practical sessions building NLP applications for:

- sentiment analysis, retrieval, summarisation, translation, named entity extraction, question answering, chatbots, personal assistants,



Nutrition information			
Typical values	Per 100g	Per 1/4 pot	% based on GDA for women
Energy	256 kJ 61 kcal	320 kJ 76 kcal	 3.8%
Protein	4.9g	6.1g	13.6%
Carbohydrate of which sugars of which starch	6.9g 6.9g nil	8.6g 8.6g nil	3.7% 9.6% -
Fat of which saturates mono-unsaturates polyunsaturates	1.5g 0.9g 0.4g nil	1.9g 1.1g 0.5g nil	2.7% 5.5% - -
Fibre	nil	nil	nil
Salt of which sodium	0.2g trace	0.3g 0.1g	5.0% 4.2%
Vitamins & minerals			% of RDA Recommended daily amount
Calcium	168mg	210mg	26%

Nutrition Information UK Label Yoghurt by Samatarou (CC0 1.0)

**Final course program will depend on student background & time constraints

Course Material

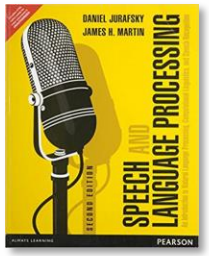
Slides and Textbooks

Slides for the course available on Webeep:

- https://webeep.polimi.it/my/

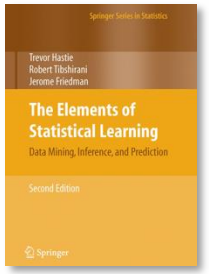
Main textbook:

- *Speech and Language Processing:* An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, by Daniel Jurafsky and James H. Martin
  - draft of the 3rd Edition is available online at: https://web.stanford.edu/~jurafsky/slp3/



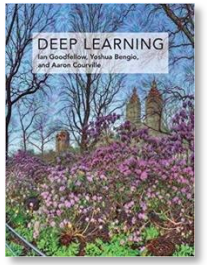
Additional Textbook for basic Machine Learning:

- *The Elements of Statistical Learning:* Data Mining, Inference, and Prediction, by Trevor Hastie, Robert Tibshirani & Jerome Friedman
  - second Edition, available online: https://web.stanford.edu/~hastie/ElemStatLearn/



Additional Textbook for Deep Learning:

- *Deep Learning* By Ian Goodfellow, Yoshua Bengio & Aaron Courville
  - available online: http://www.deeplearningbook.org
  - CAVEAT: recent techniques like Transformers didn't exist in 2016 when book was written



Assessment

Assignments & Exams

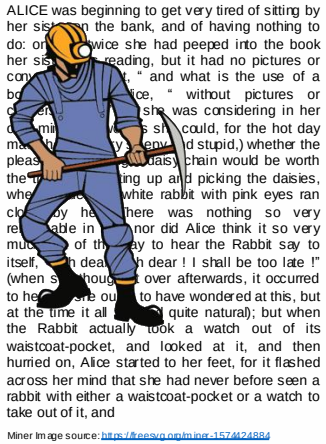
Assignment

- worth 40% of grade
- work in groups on a **fun NLP project**
- more on that later …

Written (or Oral) Exam

- Worth 60% of grade
- will also be **lots of fun**
- pay attention during the lectures and you'll be fine …

Text Processing and Python



ALICE was beginning to get very tired of sitting by her sis[illegible] the bank, and of having nothing to do: on[illegible] twice she had peeped into the book her sis[illegible] reading, but it had no pictures or conv[illegible], " and what is the use of a bo[illegible]ice, " without pictures or c[illegible] she was considering in her [illegible]mit[illegible]s she could, for the hot day ma[illegible]th[illegible]y s[illegible]d stupid,) whether the pleas[illegible] daisy chain would be worth the t[illegible]ing up and picking the daisies, whe[illegible] white rabbit with pink eyes ran clo[illegible]by he[illegible]here was nothing so very re[illegible]able in [illegible] nor did Alice think it so very muc[illegible] of th[illegible]ay to hear the Rabbit say to itself, [illegible]h dea[illegible]h dear ! I shall be too late !" (when s[illegible]thou[illegible]t over afterwards, it occurred to he[illegible]she ou[illegible] to have wondered at this, but at the time it all [illegible] quite natural); but when the Rabbit actually took a watch out of its waistcoat-pocket, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket or a watch to take out of it, and

Miner Image source: https://freesvg.org/miner-1574424884

Text processing lecture:

- Why process text?
- What can be done with text?
- Text processing is hard
- Brief history of NLP
- Pre-processing Text
- Regular Expressions

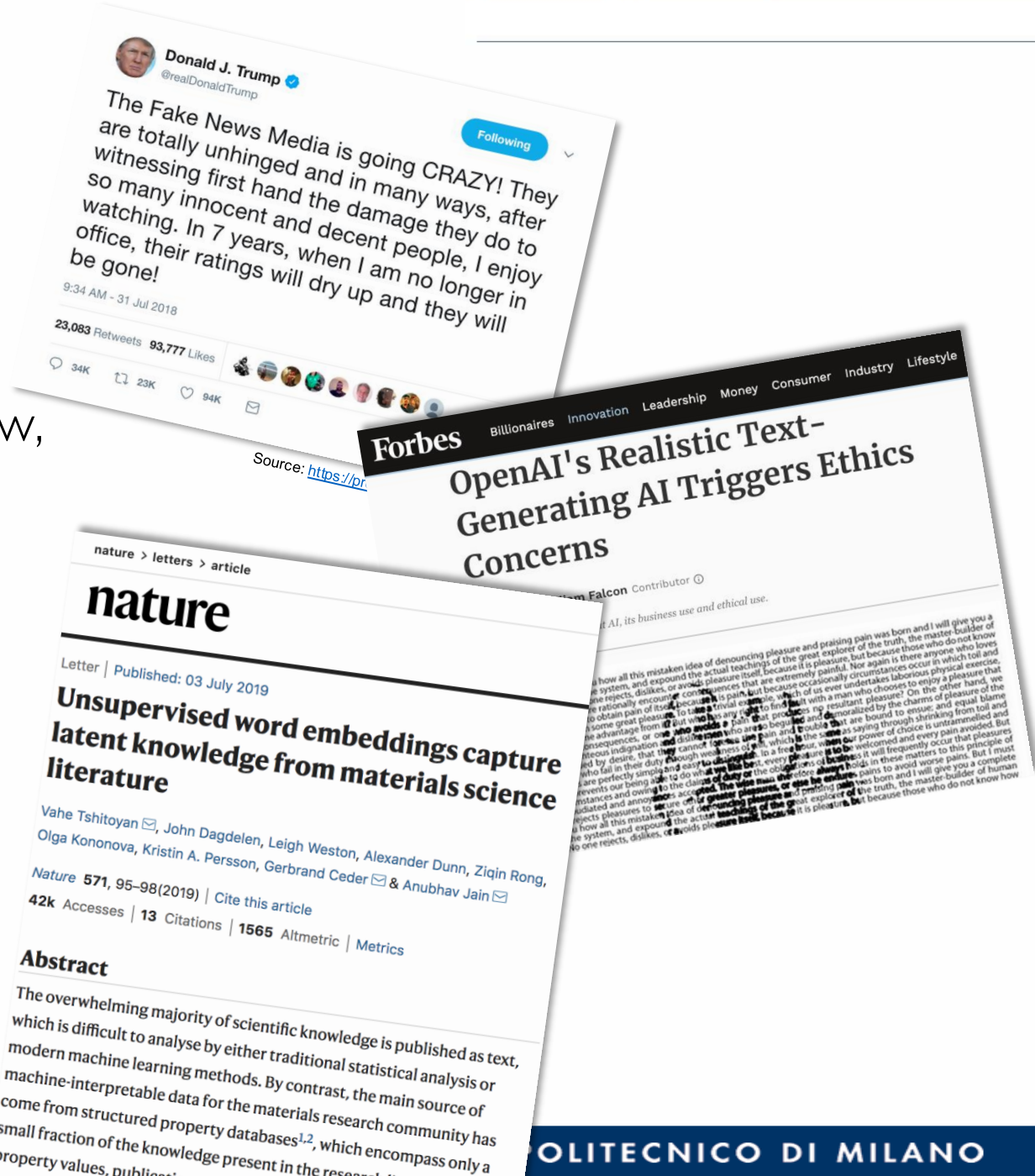
Why process text?

Because text is **pervasive**

- personal communications, news, finance, law, literature, scientific publications

Because text is **important**

- can influence public opinion
- make scientific discoveries, …



Example of NLP tasks in a specific domain: medical documents

Types of tasks: classification, extraction & search

Medical text classification

- label document with **procedure, diagnosis, motivation,** billing **code**, etc. and predict patient **outcome** (e.g re-admission risk)

Medical data extraction

- extracting **entities** (e.g. diagnostic tests), **linking entities** (e.g. reconcile drug names), **relation extraction** (determine drug dosage), **event detection** (administered on …)

Disambiguation

- E.g. expanding abbreviations: "MR" → magnetic resonance, mitral regurgitation, …

Patient similarity search

- find most similar patient for diagnosis or cohort selection

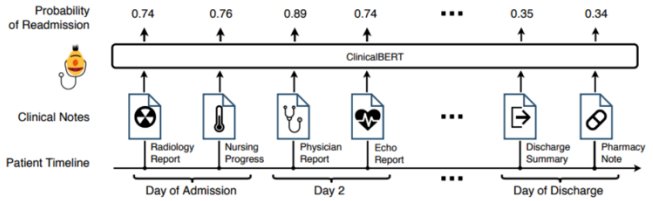


Image source: https://arxiv.org/pdf/2107.02975.pdf

Image source: https://www.johnsnowlabs.com/

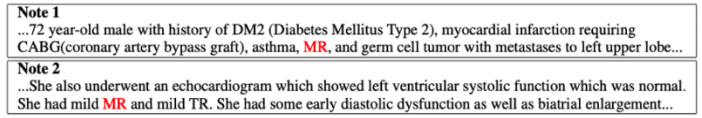


Image source: https://arxiv.org/pdf/2107.02975.pdf

Types of tasks: text generation

Translation

- e.g. Italian to English or medical jargon to plain language for patient consumpt

Summarisation

- of patient medical health history or related medical literature

Anonymisation and **synthetic data** generation

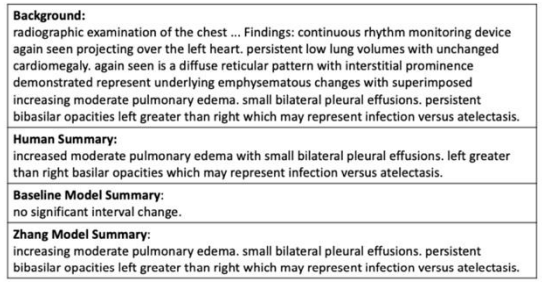
- remove sensitive informative or create synthetic datasets

Question answering

- directly answer medical questions based on text in EHR

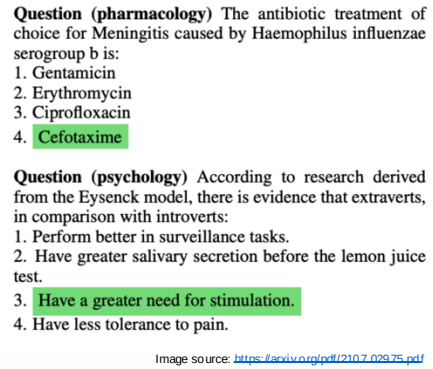
Explanations

- explain how the model came to certain prediction/diagnosis



Background: radiographic examination of the chest ... Findings: continuous rhythm monitoring device again seen projecting over the left heart. persistent low lung volumes with unchanged cardiomegaly. again seen is a diffuse reticular pattern with interstitial prominence demonstrated represent underlying emphysematous changes with superimposed increasing moderate pulmonary edema. small bilateral pleural effusions. persistent bibasilar opacities left greater than right which may represent infection versus atelectasis.
Human Summary: increased moderate pulmonary edema with small bilateral pleural effusions. left greater than right basilar opacities which may represent infection versus atelectasis.
Baseline Model Summary: no significant interval change.
Zhang Model Summary: increasing moderate pulmonary edema. small bilateral pleural effusions. persistent bibasilar opacities left greater than right which may represent infection versus atelectasis.

Image source: https://arxiv.org/pdf/2107.02975.pdf



Question (pharmacology) The antibiotic treatment of choice for Meningitis caused by Haemophilus influenzae serogroup b is:
1. Gentamicin
2. Erythromycin
3. Ciprofloxacin
4. Cefotaxime

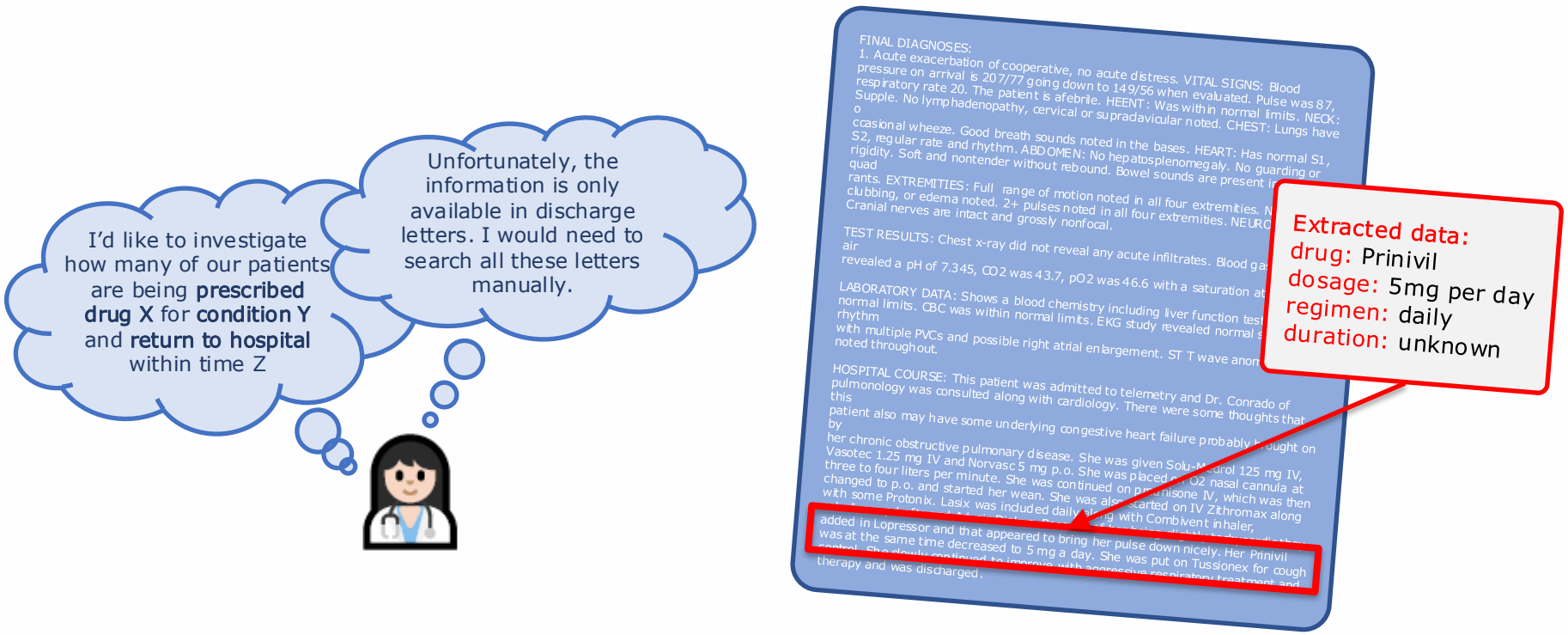
Question (psychology) According to research derived from the Eysenck model, there is evidence that extraverts, in comparison with introverts:
1. Perform better in surveillance tasks.
2. Have greater salivary secretion before the lemon juice test.
3. Have a greater need for stimulation.
4. Have less tolerance to pain.

Image source: https://arxiv.org/pdf/2107.02975.pdf

Data extraction: prescription from discharge letters

goal: extract prescription information from discharge letter

- *drug dosage information, diagnosis information, appointment information*



Not yet convinced you
want to study NLP?

Why Should You Care?

1. Enormous amount of knowledge now available in machine readable form as natural language text
2. Conversational agents becoming important form of human-computer communication
3. Much of human-human communication now mediated by computers

How hard can processing natural language be?

NLP is difficult

Even Microsoft's Bing Image Creator was confused what Noam Chomsky was talking about: https://sl.bing.net/i3pTFKtDamq

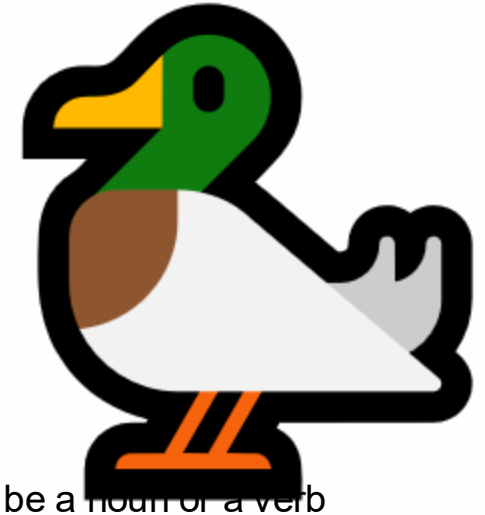
Because human language is **extremely expressive**:

- most of human knowledge is recorded in books
- but one can quite literally say **anything** in natural language

Even nonsensical statements can be expressed in natural language:

- *Colorless green ideas sleep furiously.*
  - Makes no sense, but is grammatically correct and famous enough to have its own Wikipedia page: https://en.wikipedia.org/wiki/Colorless_green_ideas_sleep_furiously
- *I didn't just say what I just said.*
  - Simple logical inconsistency that nonetheless carries meaning.

NLP is difficult (cont.)



Because human language can be **highly ambiguous**

- resolving ambiguity fundamental problem of computational linguistics

Example of an ambiguous statement:

- *I made her duck*

What did you do exactly? Did you:

- get her to lower her head (to avoid being hit)?
- cook the food that she had bought?
- build a duck-shaped statue and give it to her?
- magically turn her into a duck?

Lexical category: "duck" can be a noun or a verb

Lexical category: "her" can be a possessive ("of hers") or dative ("for her") pronoun

Lexical Semantics: "make" can mean "create" or "cook"

Grammar: "make" is a complicated verb. It can be transitive (take an object), ditransitive (take 2 objects), or action-transitive (takes an object & another verb)

NLP is difficult (cont.)

Thankfully natural language is also often very **redundant**.

Consider the sentences:

- I'm a massive fan of Britney Spears!
- Massive fan of Britney Spears!
- Massive Britney fan!
- Britney Spears? Massive fan!
- Massive fan of Brit**tan**y Spears!
- Masiv fan Brtney
- I'm a maaasssive fan of Britney Spears!

Source: https://en.wikipedia.org/wiki/Britney_Spears#/media/File:Britney_Spears_2013_(Straighten_Crop).jpg

Note, there are a LOT of ways one can misspell Britney Spears.

- Just ask Google: http://archive.google.com/jobs/britney.html

NLP is difficult (cont.)

Because even **prosody**

- the way somebody pronounces and emphasises the text

can affect its meaning

- Consider the sentence:

I never said she stole my money.

What happened exactly?

Depends where you place **emphasis:**

- *I never said she stole my money.* [Somebody else said she stole it.]
- *I never said she stole my money.* [I didn't say she stole it.]
- *I never said she stole my money.* [I only implied she stole it.]
- *I never said she stole my money.* [I said someone did, not necessarily her.]
- *I never said she stole my money.* [I considered it borrowed.]
- *I never said she stole my money.* [Only that she stole money.]
- *I never said she stole my money.* [She stole something of mine.]

Source: https://www.distractify.com/fyi/2015/04/13/19NMFR/the-19-most-mind-blowing-sentences-in-the-english-language-1197891759

Very Brief History of NLP

Very Brief history of NLP

Field grew out of Linguistics, Computer science, Speech Recognition & Psychology

1940-1950 - World War II

- Finite State Automata: Formal Language Theory, Probabilistic algorithms for speech, information theory (Shannon)

1957-1970 - Two paradigms

- **Symbolic**: Formal Language Theory, AI Logic Theories,
- **Stochastic**: Bayesian method and use of dictionaries and corpora, first OCR, Brown Corpus

1970-1993 - empiricism and Finite-State Models

- Understanding natural language semantics, discourse Modeling: substructure analysis
- Speech recognition based on probabilistic models @IBM, Data-driven approaches for POS tagging, parsing & annotation, ambiguity resolution, Natural Language Generation

1994-1999 - decline of symbolic approach

- Heavy use of data-driven methods, new application areas (Web)

2000-2010 - empiricism and Machine Learning

- Empirical approaches even more significant: large amount of annoated material online, liaison with ML+HPC community, unsupervised systems become important

2010-2018 - Machine Learning everywhere

- Neural Networks for NLP, Conversational Agents, Subjectivity and Sentiment Analysis

2018-… - Transformer architectures

- Transfer learning using pretrained language models, massive online language models

Current Technology
is **amaaaaaazzzziiing**!!!!

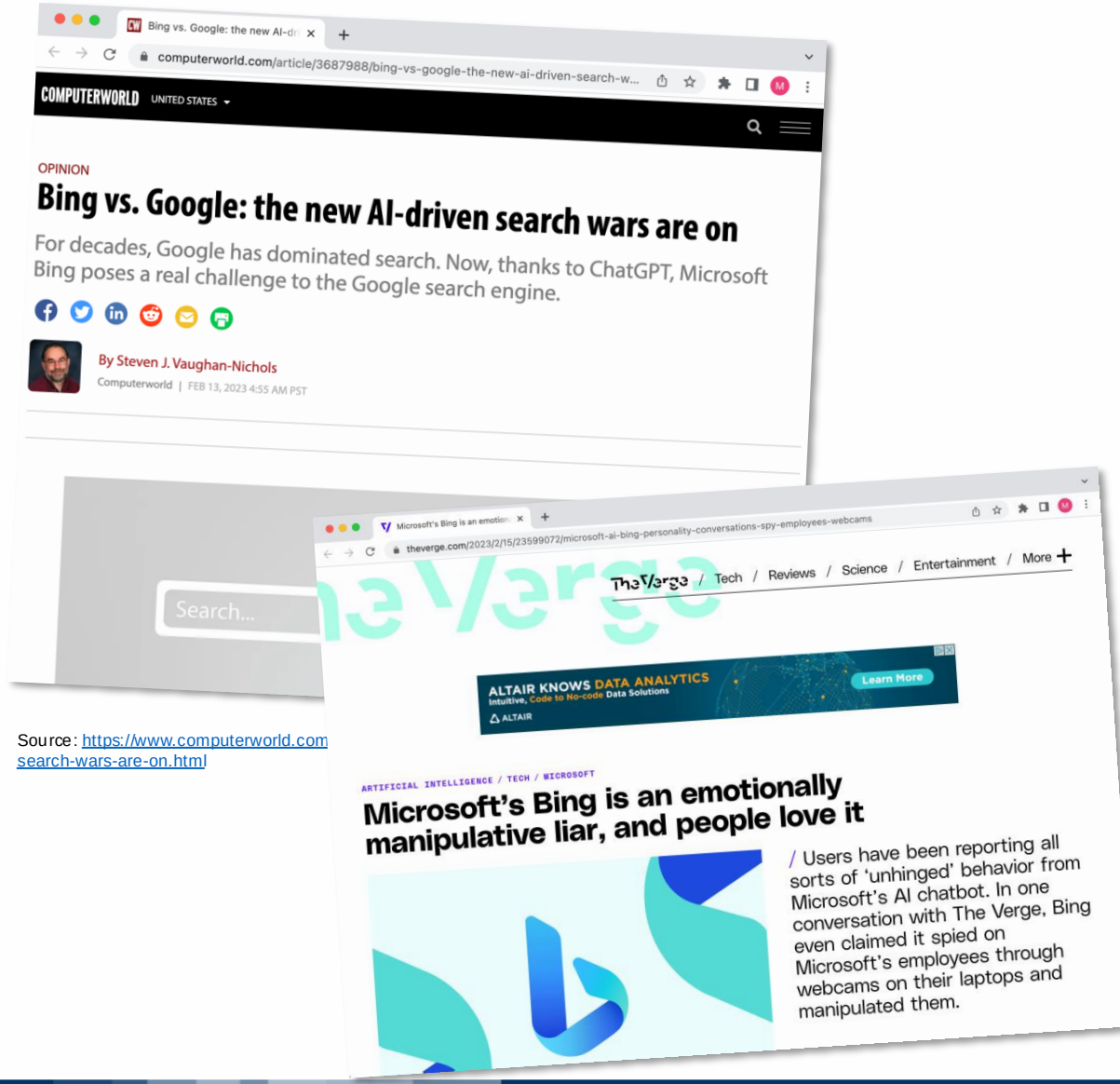
Lots of interest in chatbots for search these days

Latest generation of Language Models have become incredibly good at conversation

- Microsoft and Google scramble to make use of chatbots to power/extend their search interface.

Chatbots can emulate human conversation extremely well

- but humans aren't always nice to one another! 😳☐



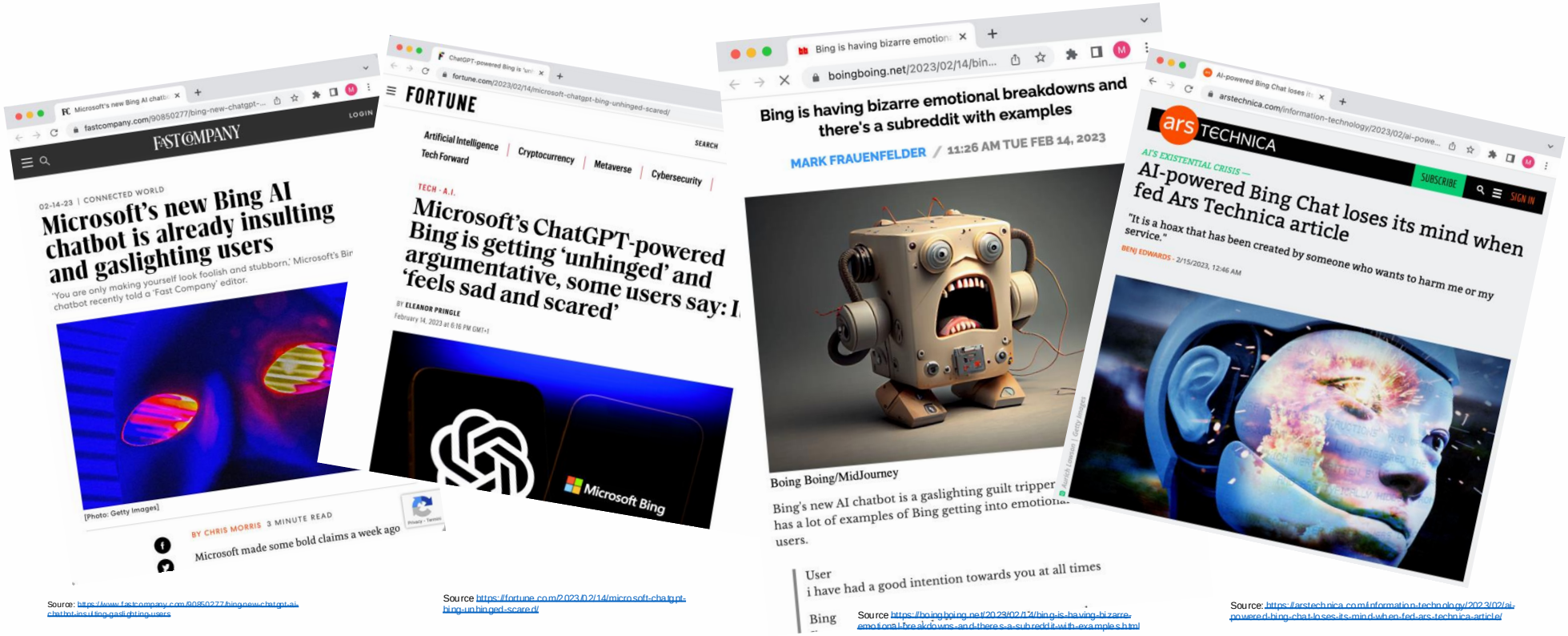
Source: https://www.computerworld.com/article/3687988/bing-vs-google-the-new-ai-driven-search-wars-are-on.html

Source: https://www.theverge.com/2023/2/15/23599072/microsoft-ai-bing-personality-conversations-spy-employees-webcams

aside: Anthropomorphism

Act of **ascribing human emotions** to non-human entities:

- may become a problem as chatbots get better and better
- people are already worried about the emotional state of Bing Search …



Source: https://www.fastcompany.com/90850277/bing-new-chatgpt-ai-chatbot-insulting-gaslighting-users

Source https://fortune.com/2023/02/14/microsoft-chatgpt-bing-unhinged-scared/

Source https://boingboing.net/2023/02/14/bing-is-having-bizarre-emotional-breakdowns-and-theres-a-subreddit-with-examples.html

Source: https://arstechnica.com/information-technology/2023/02/ai-powered-bing-chat-loses-its-mind-when-fed-ars-technica-article/

Preprocessing Text

Pre-processing text

Common to pre-process text by performing cleaning activities, such as:

- Prior to tokenisation:
  - Remove **mark-up** (non-content information), e.g. <HTML> tags
  - **Lowercase** the text (can lose information: e.g. 'WHO' vs 'who')
  - Remove **punctuation** (;,&%$@!><? etc.)
- After tokenisation:
  - Remove **stopwords** (extremely high frequency words)
  - Remove **low frequency words**
  - Perform **stemming** or **lemmatization** to reduce vocabulary size
    - e.g. fishing => fish, thought => think, etc.
  - Perform **spelling correction**

Extracting Plain Text

Extracting plain text

We may need to extract text from:

- textual documents: **.txt, HTML, e-mail**, etc.
  - usually discard mark-up (html tags) and other format-specific commands
  - in web crawl situations parser should be robust to badly formed HTML
- binary documents: **Word, PDF,** etc.
  - much more complex to handle
  - for PDF documents, structure of the text needs to be reconstructed
    - if text contains multiple columns, these need to be identified so correct flow of text can be re-established
    - if all PDFs have same format, then rules could be hand-written, otherwise machine learning might be needed
- images of **scanned documents**
  - requires specialized Optical Character Recognition (OCR) software that is now deep learning based
  - OCR is not perfect, so may introduce recognition errors in the text

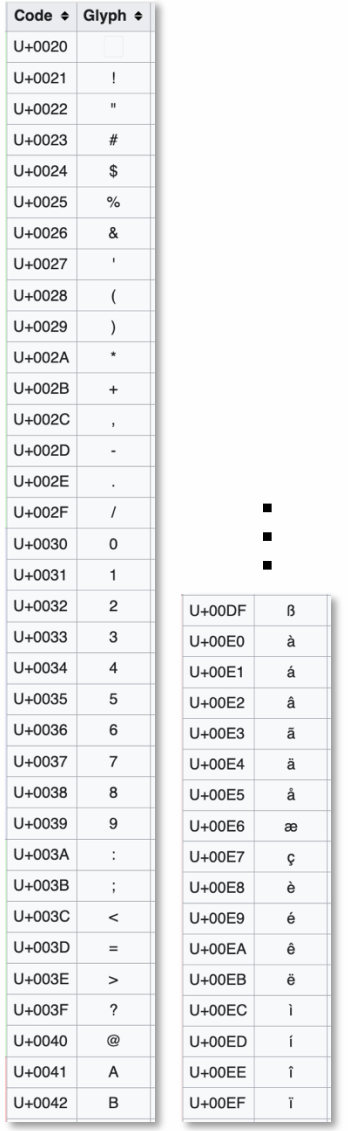
Text Encodings

Various **encodings** could be used to store characters on computer

- each supports a different number of possible characters
- **ASCII** encoding (traditional keyboard)
  - 'A' → 65, 'B' → 66, …, 'a' → 97, 'b' → 98, …
  - only 128 characters in total
- **UTF-8** encoding
  - https://en.wikipedia.org/wiki/Unicode
  - handles **149k Unicode characters**
  - works for 160+ languages

Why do we need Unicode? To handle languages with:

- non-latin character sets: Arabic, Cyrillic, Greek, Devanagari, etc.
- special characters: e.g. diacritical signs in Italian "Questa **è** così" and even in English: "Naïve"



Code	Glyph
U+0020	
U+0021	!
U+0022	"
U+0023	#
U+0024	$
U+0025	%
U+0026	&
U+0027	'
U+0028	(
U+0029	)
U+002A	*
U+002B	+
U+002C	,
U+002D	-
U+002E	.
U+002F	/
U+0030	0
U+0031	1
U+0032	2
U+0033	3
U+0034	4
U+0035	5
U+0036	6
U+0037	7
U+0038	8
U+0039	9
U+003A	:
U+003B	;
U+003C	<
U+003D	=
U+003E	>
U+003F	?
U+0040	@
U+0041	A
U+0042	B

U+00DF	ß
U+00E0	à
U+00E1	á
U+00E2	â
U+00E3	ã
U+00E4	ä
U+00E5	å
U+00E6	æ
U+00E7	ç
U+00E8	è
U+00E9	é
U+00EA	ê
U+00EB	ë
U+00EC	ì
U+00ED	í
U+00EE	î
U+00EF	ï

Image source: https://en.wikipedia.org/wiki/List_of_Unicode_characters

Tokenizing Text

Text Tokenization

Many (if not all) NLP tasks require **tokenization**:

- **segmenting the text into sequences of characters called tokens**
- usually **tokens correspond to the words in the text** (although sometimes we tokenize at the character level)

So **tokenization** is **process of splitting up sentences into words**

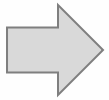
- requires language-specific resources
- can be difficult for some languages (e.g. Chinese)

Space-based tokenization

Some languages use space characters between words:

- e.g. English, Arabic, Cyrillic, Greek, Latin, Devanagari, etc.
- can segment the text into tokens based on the white-space between words
- notethatitispossibletoreadasentenceinenglishwithoutspacesbetweenthewords
  - so if we didn't have spaces, we could work out some other way to tokenize text …
  - but given hat they are available, we **may as well make use of them** ;-)

```
Sonnet 1:
BY WILLIAM SHAKESPEARE
From fairest creatures we desire increase,
That thereby beauty's rose might never
die,
But as the riper should by time decease,
His tender heir might bear his memory;
…
```



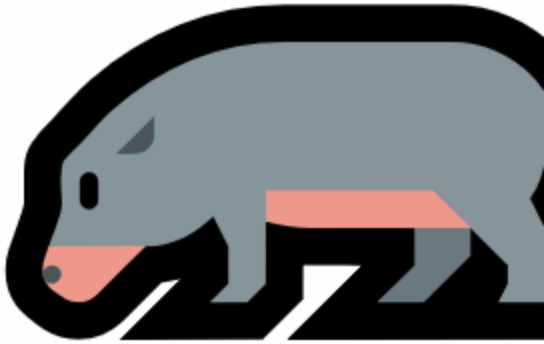
["Sonnet", "1", "BY", "WILLIAM", "SHAKESPEARE", "From", "fairest", "creatures", "we", "desire", "increase", "That", "thereby", "beautys", "rose", "might", "never", "die", "But", "as", "the", "riper", "should", "by", "time", "decease", "His", "tender", "heir", "might", "bear", "his", "memory, …]

Issues with space-based tokenization

Problems with tokenizing certain texts

- depending on application, may want to split **hyphenated words**:
  - "Italian-style furniture" => "Italian", "style", "furniture"

- some languages are highly **agglutinative**, and can build very long and specific content, which might be better to separate out
  - hippopotomonstrosesquipedaliophobia [ENGLISH]
    = fear of monstrously long words
    (literaly: hippopotamus + monster + one-and-a-half + feet + fear)
  - die Unabhängigkeitserklärung [GERMAN]
    = the Declaration of Independence
  - incontrovertibilissimamente [ITALIAN]
    = in a way that is very difficult to falsify



- some times the "unit of meaning" is spread over two **non-hyphenated words** in **multi-word expressions** (MWE):
  - New York => New_York
  - rock 'n roll => rock_'n_roll

Issues in Tokenization (cont.)

Can't blindly remove punctuation:

- m.p.h., Ph.D., AT&T, cap'n
- prices ($45.55)
- dates (01/02/06)
- URLs (http://www.stanford.edu)
- hashtags (#nlproc)
- email addresses (someone@cs.colorado.edu)

May need to deal with clitics: words that don't stand on their own

- "are" in we're, French "je" in j'ai, "le" in l'honneur

Examples of simple Tokenizers

Default tokenizer in scikitlearn:

- uses the regular expression: token_pattern = '(?u)\b\w\w+\b'
  - where \b is matches word-boundaries (or start/end of string),
  - and \w = [a-zA-Z0-9] = any 'word' character

Tokenizer in NLTK (Natural Language Tool Kit in Python):

- uses a more complicated regular expression to catch various types of tokens:

```
pattern = r'''(?x)    # set flag to allow verbose regexps
    ([A-Z]\.)+        # abbreviations, e.g. U.S.A.
  | \w+(-\w+)*        # words with optional internal hyphens
  | \$?\d+(\.\d+)?%?  # currency and percentages, e.g. $12.40, 82%
  | \.\.\.            # ellipsis
  | [][.,;"'?():-_`]  # these are separate tokens; includes ], [
'''
```

- so that: `text = 'That U.S.A. poster-print costs $12.40...'`
- becomes: `['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', '...']`

Tokenization in languages without spaces

Many languages, such as Chinese, don't use spaces to separate words!

- How do we decide where the token boundaries should be?

Chinese words contains 2.4 characters on average

- so deciding what counts as a word can be difficult
- consider sentence: 姚明进入总决赛 "Yao Ming reaches the finals" with potential tokenisations at different levels:
  - 姚明 进入 总决赛
    YaoMing reaches finals
  - 姚 明 进入 总 决赛
    Yao Ming reaches overall finals
  - 姚 明 进 入 总 决 赛
    Yao Ming enter enter overall decision game
- so common to just treat each character as a token

In other languages (like Thai & Japanese) complex segmentation is required

Other options for text tokenization

Instead of **white-space segmentation** or **single-character segmentation**

- **use the data** to tell us how to tokenize
- with a **sub-word tokenization**
  - useful for splitting up longer words
  - and for allowing the machine learning model to learn explicitly the morphology of the language
- we will use **byte-pair encoding** to do this later in the course when we do deep learning.

Sentence Segmentation

Certain tasks require sentences to be segmented.

- punctuation marks: "**!**" and **"?" often** indicate the end of the statement/question
  - except for maths expressions like: "5! = 120"
  - or statements containing unknowns: "Fill in the missing values: 1, 2, ?, 4, 5, ?, 7, 8, ?,10

- **period "." is commonly used to end a sentence**, but is ambiguous
  - also appears in abbreviations like **Inc.** or **Dr.** and numbers like **.02%** or **4.3**

Common algorithm:

- Tokenize and then use rules or ML to classify a period as either (a) part of the word or (b) a sentence-boundary.

Word Frequencies

How many words in a sentence?

Should we count all words, or just meaningful ones?

- *"I mainly do uh, mainly business data processing"*
- 8 words, or just 6 real (and not repeated) words?

Should we count wordforms or just word groups?

- *"Seuss's cat in the hat is different from other cats!"*
- cat and cats = same lemma, but different wordforms
  - **Lemma**: same stem and part of speech
  - **Wordform**: the full inflected surface form

Usually distinguish between types and tokens:

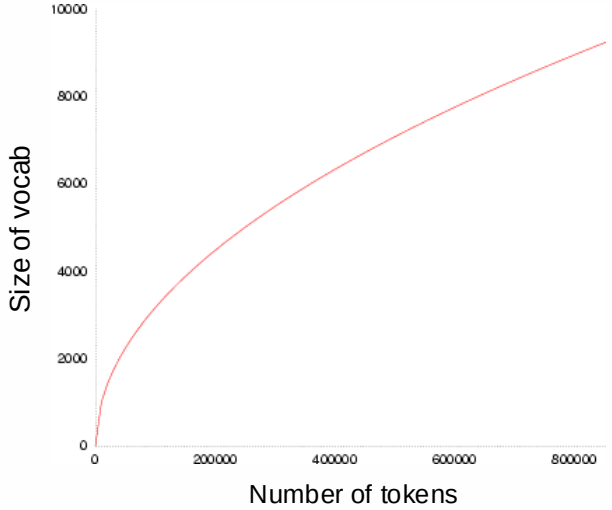
- *"She actively encouraged the discouraged child to act courageously and confront her fears."*
  - **Type/Term**: an element of the vocabulary: she, actively, …
  - **Token**: an instance of that type in running text: She, actively, …

Statistical Laws of Text

Heap's law:

- (https://en.wikipedia.org/wiki/Heaps'_law)
- Vocabulary grows with approximately the square root of document/collection length:

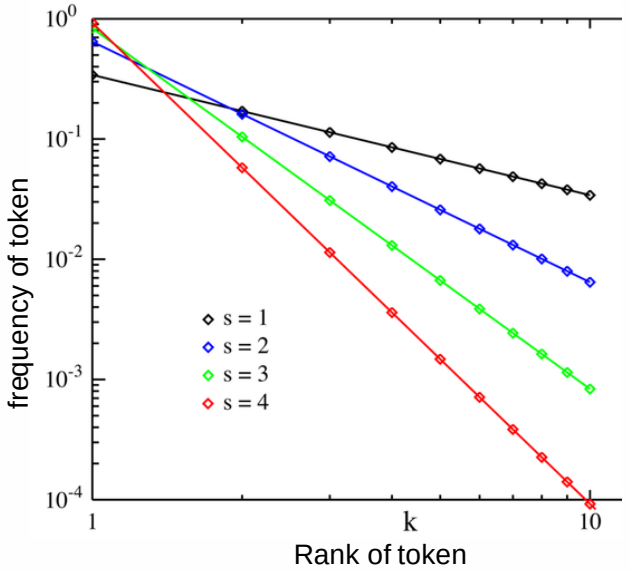
$$V(l) \propto l^{\beta}$$
$$\beta \approx .5$$



Zipf's law:

- (https://en.wikipedia.org/wiki/Zipf%27s_law)
- A token's frequency is approximately proportional to the inverse of its rank

$$\mathrm{ctf}_t \propto \frac{1}{\mathrm{rank}(t)^s}$$
$$s \approx 1$$



Monkeys and Typewriters

Examples of collections and corresponding vocabularies:

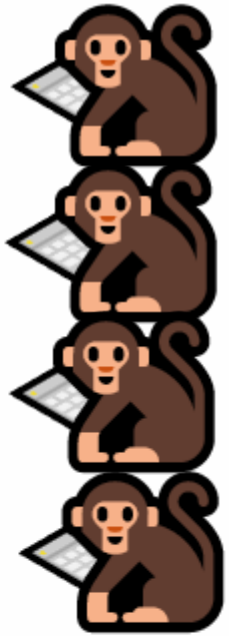
	Tokens	Vocabulary
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884 thousand	31 thousand
COCA	440 million	2 million
Google N-grams	1 trillion	13+ million

Heap's law derives from Zipf's law and can be explained by a **random typing model**

- a.k.a. Monkeys at typewriters (https://www.cs.cmu.edu/~zollmann/publications/monkeypaper.pdf)

Note that if you wait long enough and have a sufficiently large number of monkeys, eventually one of them will produce Shakespeare …

- Infinite monkey theorem (https://en.wikipedia.org/wiki/Infinite_monkey_theorem)



Normalising Text

Case folding

Who doesn't love ABBA?
⇒ who doesn't love abba?

Applications like **web search** often reduce all letters to lower case

- drastically reduces size of vocabulary and increases recall (set of valid documents found)
- users tend to use lowercase in search queries anyway

For classification problems,

- removing case reduces vocabulary and thus number of parameters that must be learnt
- helping classifier to generalise well from far fewer examples

Problem:

- sometimes lose important information by removing case
  - e.g., word "**who**" might refer to **the WHO** (the World Health Organization), **the Who** (the rock band) or a person (**who** was being talked about)
  - or "**us**" might have been **the US** or a just **us** (first person plural)
- thus retaining case can be helpful for many applications like sentiment analysis, machine translation, information extraction.

Word Normalization

Process of converting words/tokens into a standard format

- U.S.A. or USA?
- uhhuh or uh-huh?
- Fed or fed?
- am, is, be, are or be?

Critical for Web Search applications:

- otherwise query for "USA" might not return documents containing the term "U.S.A"

Morphology

Morphology

Fancy word from linguistics that refers to:

- analysis of **structure** of **words**

Morpheme: smallest linguistic unit that has semantic meaning

- e.g.: unbelievably → un-believe-able-ly

Morphemes are divided into:

- **root**: the base form (*believe*)
- affixes: **prefix** (*un-*), **infix** (*-able-*), or **suffix** (*-ly*)

Lexicon

Morphemes compose to make *lexemes*

- **Lexeme**: unit of lexical meaning that exists regardless of the number of inflectional endings it may have or the number of words it may contain
  - E.g.: BELIEVE, NEW YORK, RUN
- **Lemma**: canonical form of a lexeme
  - E.g.: TO RUN
- **Lexicon**: set of lexemes
  - Lexicons for NLP usually contain affixes and other info
- A **word** is, in general, an inflected form of a lexeme
  - E.g.: *unbelievably*, *runs*

Composing morphemes

Morphologic rules:

- restrict the ordering of morphemes

  e.g.: PLURAL NOUN = SINGULAR NOUN + PL

Orthographic rules:

- aka "spelling rules" or "two-level rules"

  e.g.: fox + s → foxes; un-believe-able-ly → unbelievably

Lexicons in NLP:

- define base forms

  e.g.: fox: NOUN, SINGULAR, …

- define affix morphemes

  e.g.: PL → s

- address irregular forms

  e.g.: wrote → root: write; mice → root: mouse

Lemmatization

Represent all words as their lemma, their shared root
= dictionary headword form:

- *am, are, is* → *be*
- *car, cars, car's, cars'* → *car*
- voglio ('I want'), vuoi ('you want') → volere 'to want'
- *He is reading detective stories* → *He be read detective story*

Complex Morphology

Dealing with complex morphology is necessary for many languages

- e.g. Turkish word: Uygarlastiramadiklarimizdanmissinizcasina
  - '(behaving) as if you are among those whom we could not civilize'
  - Uygar 'civilized' + las 'become' + tir 'cause' + ama 'not able'

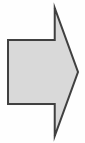
  + dik 'past' + lar 'plural' + imiz 'p1pl' + dan 'abl' + mis 'past' + siniz '2pl' + casina 'as if'

Stemming

Simple algorithm that reduces terms to stems, chopping off affixes crudely

- no lexicon needed!
- often used in text retrieval to reduce computational requirements

This was not the map we found in Billy Bones's chest, but an accurate copy, complete in all things-names and heights and soundings-with the single exception of the red crosses and the written notes.



Thi wa not the map we found in Billi Bone s chest but an accur copi complet in all thing name and height and sound with the singl except of the red cross and the written note.

Porter Stemmer

Porter Stemming Algorithm (1980)

- set of *rewriting rules*
  - e.g.: +ATIONAL → +ATE (es: relational → relate),
    +ING → ε (es: motoring → motor), …
- simple, but error prone:
  - collisions (different words, same stems)
    - 'correctional facilities' → 'correct', 'facil'
    - 'facile corrections' → 'facil', 'correct'
  - searching for 'correctional facilities' may return docs containing 'facile corrections'

  - Porter Stemmer for English http://www.tartarus.org/~martin/PorterStemmer/
  - Snowball (programming language for stemmers) http://snowball.tartarus.org/

Stemming vs lemmatization

the boy's cars are different colors ⇒
the boy car be differ color

In **text retrieval** to prevent vocabulary mismatch between query and document:

- usually perform stemming (or lemmatization) before adding terms to the index

  car, cars, car's, cars' ⇒ car

Difference between Stemming and Lemmatization?

- Stemming = Simple algorithm that applies rules to extract word stems.
  - See for example Porter's stemmer
  - http://snowball.tartarus.org/algorithms/porter/stemmer.html
- Lemmatization = More sophisticated NLP technique
  - uses vocabulary and morphological analysis to extract the lemma
  - can be important for morphologically rich languages (e.g. French)

Stopword Removal

Word
the
be
to
of
and
a
in
that
have
I
it
for
not
on
with
he
as
you
do

Stopword lists

Stopwords are just most frequent terms in language

- extremely high document frequency scores (low discriminative power)
- convey very little information about the topic of the text

Removing stopwords can sometimes boost performance of retrieval/classification models

- More likely it will just reduce computational / memory burden
- plus speeds up index by removing massively long posting lists

Problem: sometimes stopwords are useful!

- consider the Rockband "the The"
- or "a white house" vs "the white house"

The The

Band

thethe.com

Available on

YouTube

Spotify

Deezer

The The are an English post-punk band. They have been active in various forms since 1979, with singer/songwriter Matt Johnson being the only constant band member. Wikipedia

Members: Matt Johnson, Johnny Marr, Jools Holland, MORE

Genres: Post-punk, New wave, Alternative rock

Origin: London, United Kingdom, England, United Kingdom

Regular Expressions

Regular expressions – what are they?

Text documents

```
[+-]?(\d+(\.\d+)?
```

- are simply **sequences of characters:**

  "Each document is a sequence of characters, where each character is represented on a computer by an integer value. For instance the character 'a' is represented by the number 97, while 'b' is the number 98, and so on…."

Regular expressions

- are just **patterns** that allow us to **search** within text documents
- for **specific sequences of characters**

Why do we want to search with regular expressions?

1. so we can find out **whether pattern exists** in document
2. so we can **extract information** from document wherever pattern occurs

Regular expressions – simple examples

Simplest pattern is an **exact match**:

- the regular expression: '**abc**'
  - will match the sequence 'aaabcdddd'
  - but not the sequence 'aaabdddd', since the exact pattern doesn't appear in it

Next simplest pattern is a **choice** between two sequences:

- the regular expression: '**(abc|bdd)**'
  - will match both the sequence 'aaabcdddd'
  - and also the sequence 'aaabdddd'

Regular expressions – wildcards & square-brackets

An important pattern involves a **wildcard symbol** '.'

- it matches **any character** (except for the newline character)
- e.g. the regular expression with 2 consecutive dots: **'a..d'**
  - will match the sequence 'aaabcdddd'
  - but not the sequence 'aaabbcddd'

Another common pattern involves **square brackets** []

- it indicates a choice for a single character
- **[abc]** = (a|b|c) = any one of characters within the brackets
- **[a-z]** = (a|b|…|z) = any character in range a, b, …, z
- **[^abc]** = any characters except those that match [abc]

Regular expressions – special characters

Other special characters that can be used in regular expressions:

- all of them are prefixed with the backslash character '\'
- \n = newline character
- \t = tab character
- \s = any whitespace character
- \S = any non-whitespace character
- \d = [0-9] = any digit
- \w = [a-zA-Z0-9] = any 'word' character

Regular expressions – repetition

The real power of a regular expression comes from **repetition**

- the following patterns, when added to a regular expression, tell us how many times the previous character (or pattern) must be repeated:

  * = zero or more times

  \+ = one or more times

  ? = zero or one times

  {n} = exactly n times

  {n,m} = at least n, up to m times

- example: the regular expression **'ad*'**
  - would match sequence 'aaaaaddddddcc' ← **greedily** matches longest sub-sequence possible
  - and also the sequence 'aaaaacc' ← since character 'd' can appear **zero** times

More complicated example

Consider the regular expression:

[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}

Which of the following text sequences would it match?

- 'my email is Steve.Rogers@iamyourcaptain.com'
- '@Steve, that new shield you ordered has just arrived'
- 'send jamesbond007@hermajestyssecretservice.co.uk a mail & wait for a reply'
- 'see you in the bar at @7 for a vodka martini'
- 'I was up way too l@te last night watching old superhero films'

What is the pattern looking for?

Pros and cons of regular expressions

Regular expressions provide a powerful language for writing rules to extract content from text documents

Advantages of regular-expression based text extraction:

- Simplicity of approach
- Rules can me made quite precise, to reduce number of **false positives** (items that should not have been extracted)

Limitations of regular-expression based text extraction:

- extraction rules must (usually) be written by hand, which can be difficult/laborious
- **some false positives** are usually present
  - due to insufficiency of syntactic structure to identify them
  - e.g. extract productID *849302949* as phone number because it has same form
- often **many false negatives**
  - items that should have been extracted but weren't
  - due to fact that rule is not general enough
- **hard to integrate knowledge of context** around extracted entity
  - *Dear Mr **Chair**, I find it difficult to …*

Tutorials in Python

Learning by doing -- notebooks

Given time-constraints and student cohort

- this course is **practical by design**
- with less theory and more practical sessions

In the next session we will start using Jupyter notebooks

- if you don't have Jupyter, you can either:
  - install Anaconda: https://www.anaconda.com/products/individual
  - or make use of Google colab, a free online notebook environment: https://colab.research.google.com/notebooks/intro.ipynb

Image source: https://www.anaconda.com/

Image source: https://colab.research.google.com/notebooks/intro.ipynb

All coding for the course will be in Python

- Isn't that some kind of snake?
- Yes, but but it's also the most important language for NLP/Deep Learning
- if you haven't; used Python before, do the free online course "Introduction to Python": https://www.datacamp.com/courses/intro-to-python-for-data-science

Conclusions

Conclusions

Natural language is **pervasive**

- so techniques for processing it automatically are critical

Natural language processing is **hard**

- due to unbounded expressivity and ambiguity of natural language

Hand-written regular expressions

- provide a simple mechanism for data extraction from text documents