Basic Text Processing

Tokenizing

Sentence splitting

Word/Text Normalisation

Task description

- Word and text normalization gets the text and the tokens as input

  It`s owa, Anakin. I have the high ground!

- And has the task to „normalize“ it

  It is over, Anakin. I have the high ground!

- But what does „normalize“ even mean?

Normalization

- Normalization is depending on the task, for this lecture we define it to be any operation that does indeed **modify the text**

- Which operations are there? (Listing probably not complete)
  1. Modifying individual tokens
     - Expanding abbreviations and acronyms (dept. → department)
     - Lemmatization (has → have)
     - Stemming (weakness → weak)
     - Correcting missspellings (missspelling → misspelling)
     - Anonymization (Beloved Cristiano Ronaldo → Beloved <FirstName><LastName>)
  2. Deleting tokens
     - Removing stop words (Me and my brother → Me brother)
  3. Adding tokens
     - Correcting syntax („You going to the cinema?“ → „Are you going to the cinema?“)

Normalization

- Acronym Expansion (AE)
  - What: Given any acronym (which is some sort of abbreviation) e.g. **CNN**, find the expanded string to that abbreviation
  - How:
    1. Get a list of acronyms (https://www.acronymfinder.com)

★★★★★	CNN	Cable News Network
★★★★★	CNN	Convolutional Neural Network (machine learning)
☆★★★★	CNN	Clinton News Network
☆★★★★	CNN	Cellular Neural Network (parallel computing paradigm)

Normalization

- Acronym Expansion (AE)
  - What: Given any acronym (which is some sort of abbreviation) e.g. **CNN**, find the expanded string to that abbreviation
  - How:
    1. Get a list of acronyms (https://www.acronymfinder.com)
    2. Disambiguate using the context of the Acronym („The process of **using** a **CNN** for classifying a single **pixel** “)
       - Either rule based
       - Or based on machine learning
  - Why:
    - Reduces vocabulary
    - No need to store acronyms as synonyms (useful for coreference resolution)

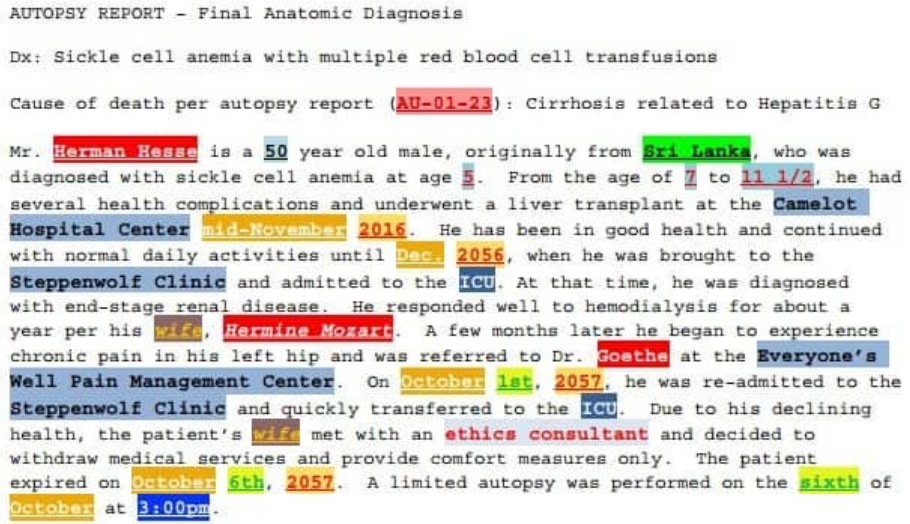
Normalization

- Correcting misspellings
  - What: Scan all tokens, and correct any misspelled ones
  - How:
    1. Get a dictionary of your language (https://www.duden.de/)
    2. Whenever you encounter a token, that is not in the dictionary:
       - Verify its character N-grams (e.g. „heihgt“ vs. „height“, the bigram „hg“ is very unlikely)
       - Calculate an **Edit-Distance** to all tokens in your dictionary
         - Symspell Algorithm
         - BK-Trees
    3. Suggest a change in the token
  - Why:
    - Reduces vocabulary (might have a huge impact!)

We will present this in a later lecture

Normalization

- Anonymization
  - What: Replace some information of a document by generics
  - How:
    - Usually done using an entity recognition approach (we will come to that later)
  - Why:
    - Some domains (e.g. medical domain) have severe data privacy regulations, using anonymized data helps to even publish some data sets (potentially!)



AUTOPSY REPORT – Final Anatomic Diagnosis

Dx: Sickle cell anemia with multiple red blood cell transfusions

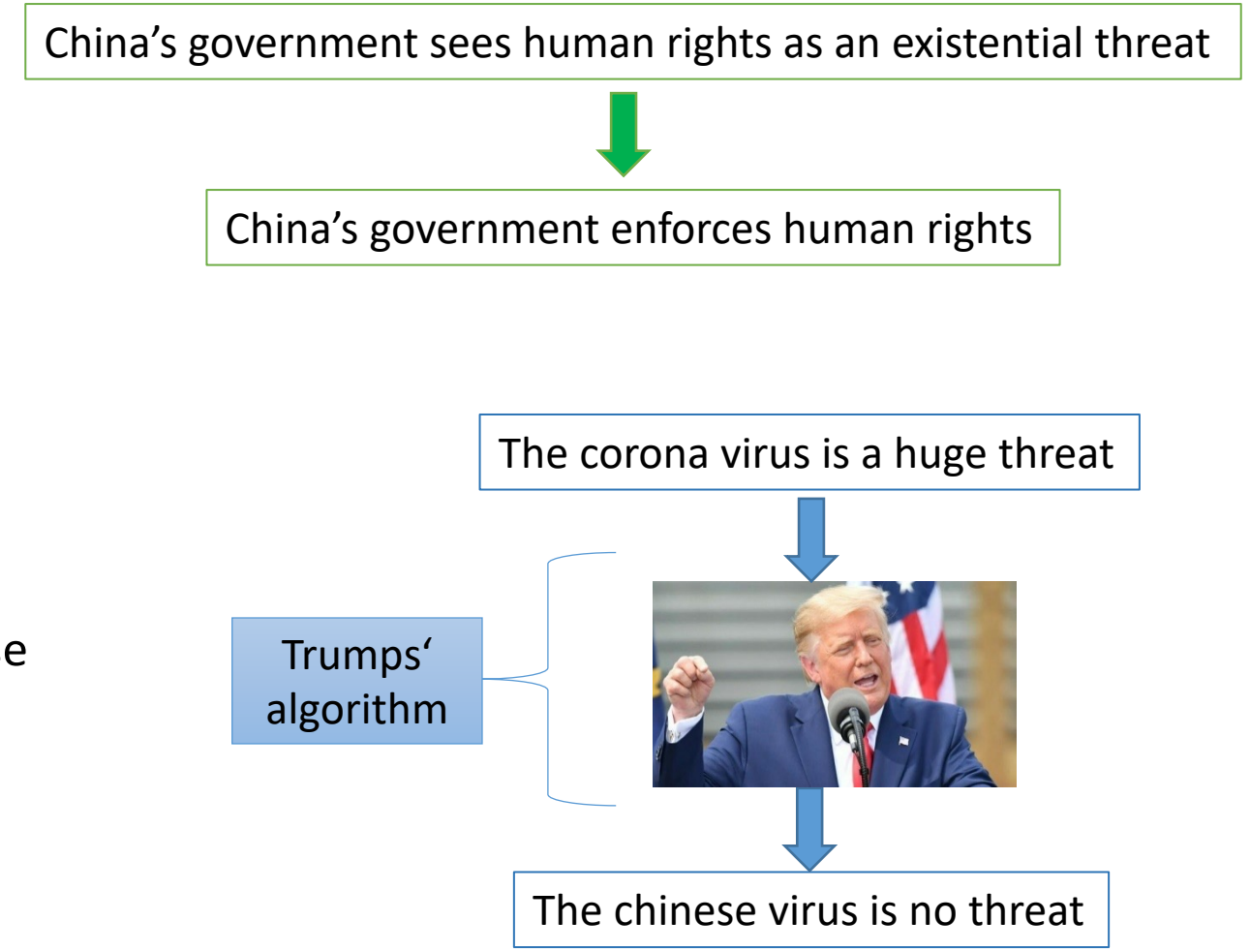
Cause of death per autopsy report (AU-01-23): Cirrhosis related to Hepatitis G

Mr. Herman Hesse is a 50 year old male, originally from Sri Lanka, who was diagnosed with sickle cell anemia at age 5. From the age of 7 to 11 1/2, he had several health complications and underwent a liver transplant at the Camelot Hospital Center mid-November 2016. He has been in good health and continued with normal daily activities until Dec. 2056, when he was brought to the Steppenwolf Clinic and admitted to the ICU. At that time, he was diagnosed with end-stage renal disease. He responded well to hemodialysis for about a year per his wife, Hermine Mozart. A few months later he began to experience chronic pain in his left hip and was referred to Dr. Goethe at the Everyone's Well Pain Management Center. On October 1st, 2057, he was re-admitted to the Steppenwolf Clinic and quickly transferred to the ICU. Due to his declining health, the patient's wife met with an ethics consultant and decided to withdraw medical services and provide comfort measures only. The patient expired on October 6th, 2057. A limited autopsy was performed on the sixth of October at 3:00pm.

Source: https://windowsreport.com/data-anonymization-software/

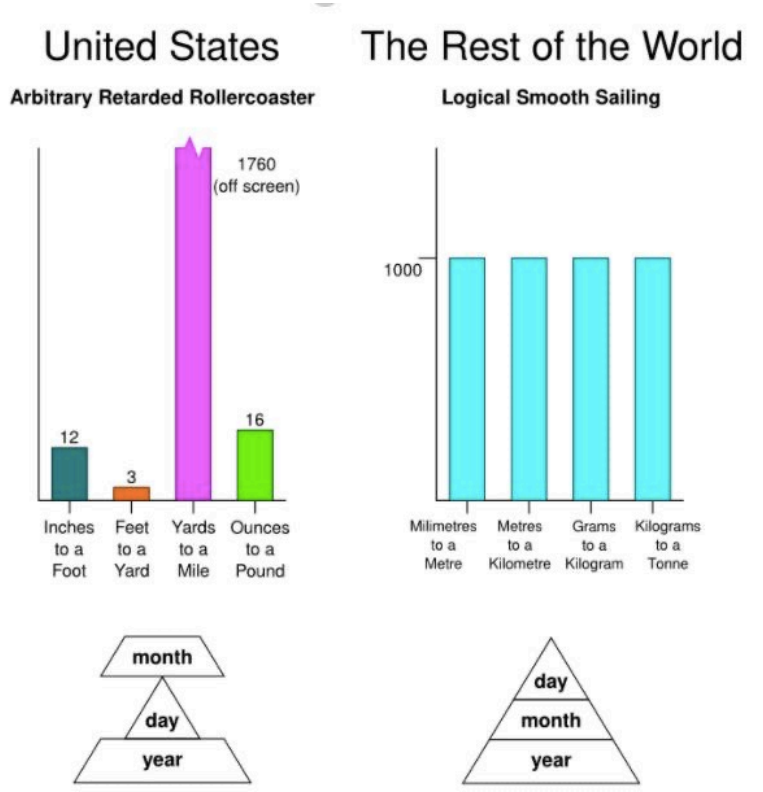
Normalization

- Paraphrasing
  - What: Replace some tokens/phrases using more appropriate expressions
  - How:
    - Tricky: Currently Deep Learning and its Seq2Seq approaches are very promising
    - Classical approaches might use a combination of patterns and parse trees
  - Why:
    - Censoring texts
    - Improving style



Normalization

- Normalizing numbers, dates, units, …
  - What: Regularize some snippets
  - How:
    - Regular Expressions (and some custom code) should be capable of doing that!
  - Why:
    - Removing noise and easing up processing of later engines



https://www.reddit.com/r/Metric/comments/cd2m47/the_imperial_system/

Normalization

- Removing „**stop words**“
  - What: Remove the most common words of a language
  - How:
    - Get a list of the most common words of your language and remove them
  - Why:
    - Some algorithms work much better without them
      - E.g. you do not want 2 texts to be similar because both contain „and“

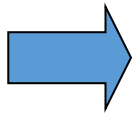
A	It
About	Its
Again	Itself
All	Just
Almost	km
Also	Made
Although	Mainly
Always	Make
An	May
And	mg
Another	Might
Any	ml
Are	mm
As	Most
At	Mostly

https://www.researchgate.net/figure/Example-of-stop-words_tbl1_262182428

Normalization

- **Stemming**
- Reduce terms to their stems in information retrieval
- Stemming is crude cutting of affixes
  - language dependent
  - e.g., automate(s), automatic, automation all reduced to automat

for example compressed and compression are both accepted as equivalent to compress.



for exampl compress and compress ar both accept as equival to compress

Stemming techniques

- **Look-up table:**
  - Problem: building the table
  - Production technique: generate all word variants from the basic words with rules, e.g. "run -> "running", "runs", "*runned", "runly*“
- **Suffix-stripping algorithms**
  - rules for reducing input word to root form
    - e.g. if the word ends in 'ed', 'ing', 'ly’, remove the 'ed', 'ing', 'ly',
  - infix-stripping algorithms, e.g. indefinitely -> definite?
  - reduction to real words? happier -> happ or happy?
  - additional rules necessary
- **Stochastic algorithms**
  - learn from annotated text
  - take the context into account to resolve ambiguity

Porter's algorithm - The most common English stemmer

- Splits word in vowel and consonant sequences [C]$(VC)^m$[V]
- Uses rule groups; from each group, **only one rule** is applied
- Available for many languages (e.g. for English)

Step 1a

sses → ss caresses → caress
ies → i ponies → poni
ss → ss caress → caress
s → ø cats → cat

Step 1b

(*v*)ing → ø walking → walk
sing → sing
(*v*)ed → ø plastered → plaster

…

Step 2 (for long stems)

ational→ ate relational→ relate
izer→ ize digitizer → digitize
ator→ ate operator → operate

…

Step 3 (for longer stems)

al → ø revival → reviv
able → ø adjustable → adjust
ate → ø activate → activ

…

Normalization

- **Lemmatization**
  - What: Reduce a token to its basic form
  - How:
    - Usually handled with a huge dictionary (e.g. the dictionary of the TreeTagger has 3.5 million entries)
    - For remaining cases - use a Finite state machine
    - Sequence2Sequence neural models
  - Why:
    - Greatly reduces vocabulary size, while retaining real words, used in:
      - Information Retrieval
      - Information Extraction

(e.g., went → go, bought → buy, is → be)

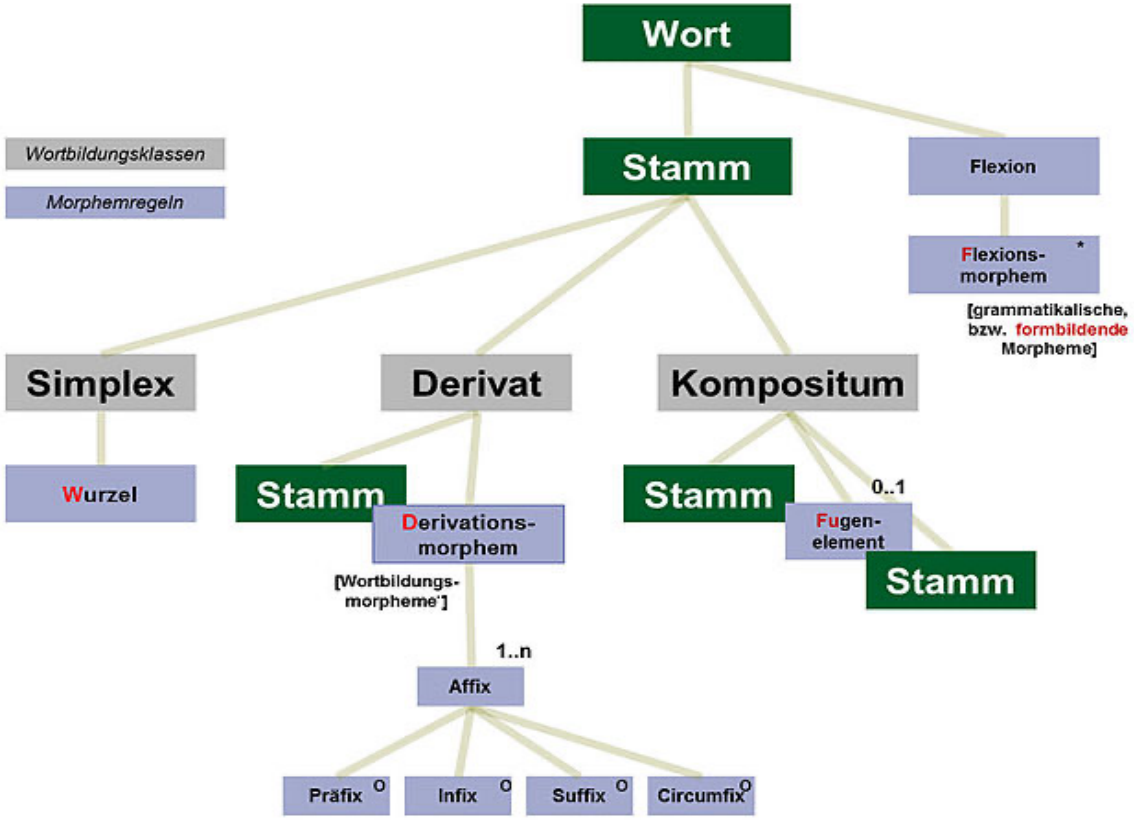
How many words?

Church and Gale (1990): $|V| > O(N^{½})$

- N = number of tokens
- V = vocabulary = set of types
  - |V| is the size of the vocabulary

	Tokens = N	Types = |V|
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13 million

How complicated can German words be?



Quelle: Von Ollio - Eigenes Werk, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=33441208

Normalization: General Applications

1. "Document Retrieval"
   - Find all documents that deal with "computer"
2. Information Extraction
   - Extract all diagnoses that are "mild"
3. Useful features for machine learning
   - If you know "computer" you would also recognize "computers"
     ➔ Reduces the amount of required training data
4. Pre-processing for:
   - Topic modelling
   - Author detection
   - …