TEXT MINING

L02. PREPROCESSING

SUZAN VERBERNE 2021

Universiteit Leiden

TODAY'S LECTURE

- Quiz about week 1
- Go from raw text to clean text
- Character encoding
- Edit distance (+ exercise)
- Regular expressions
- Tokenization and sentence splitting
- Lemmatization and stemming

QUIZ ABOUT WEEK 1

- What is Optical Character Recognition (OCR)?
  a. A technique for recognizing the correct character encoding
  b. A technique for converting handwritten text to digital text
  c. A technique for converting the image of a printed text to a digital text
  d. A technique for converting a formatted text to a plain text

QUIZ ABOUT WEEK 1

- If we use words as features in a text classification task, the resulting vectors are
  a. Low-dimensional and dense
  b. Low-dimensional and sparse
  c. High-dimensional and dense
  d. High-dimensional and sparse

QUIZ ABOUT WEEK 1

- What does the long-tail distribution for text data refer to?

a. When we add a document to a collection, the number of unique terms will increase

b. In a given document, most of the terms will have a frequency of zero

c. In a given collection, there are many terms with a low frequency and few terms with a high frequency

QUIZ ABOUT WEEK 1

- For which type of text processing task are capitalization and punctuation more useful?
  a. For sequence labelling
  b. For classification

QUIZ ABOUT WEEK 1

- Which evaluation metric would you prioritize for the task of identifying terrorist threats on Twitter, and why?
  a. Precision, because we want to be sure that we found all threats
  b. Recall, because we want to be sure that we found all threats
  c. Precision, because we don't want to accuse someone wrongly
  d. Recall, because we don't want to accuse someone wrongly

WHO FOUND A TEAM MATE?

GO FROM RAW TEXT TO CLEAN TEXT

Universiteit Leiden

Suzan Verberne 2021

GATHERING RAW TEXT

- Written text
  - Digitized (scanned) documents (We need OCR = optical character recognition)
  - Born-digital documents
    - text, html, pdf, MS Word documents

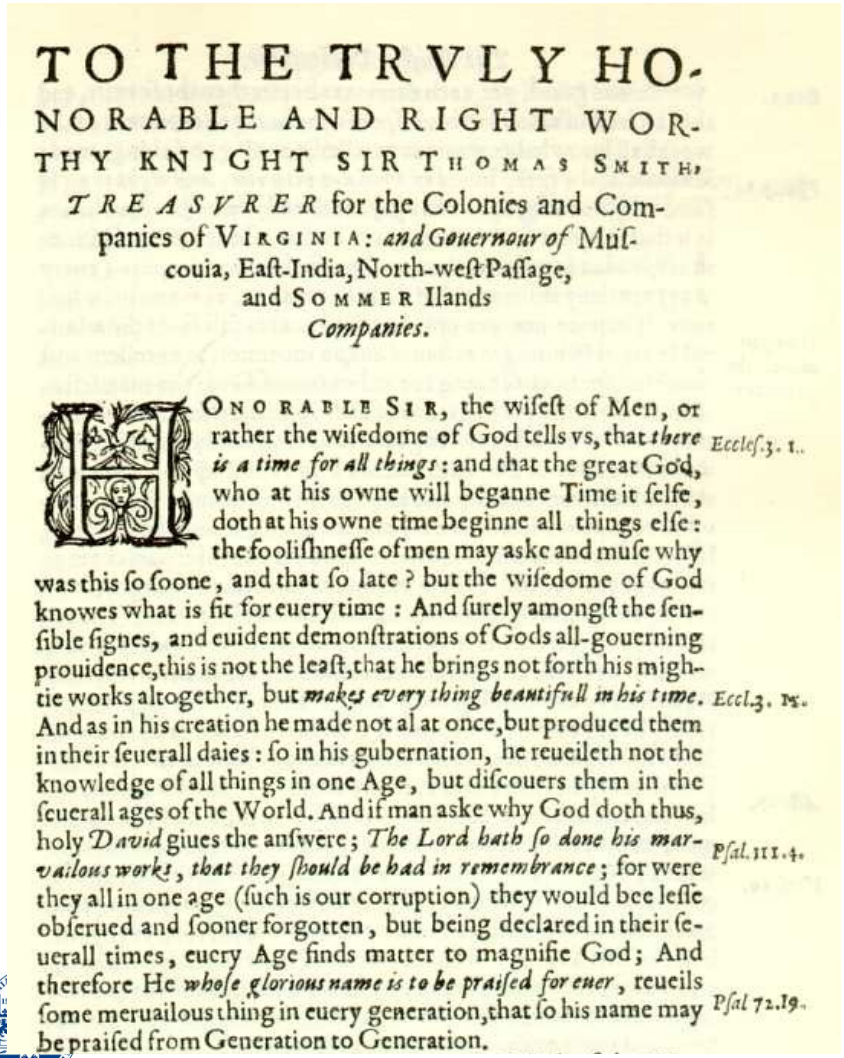
- All text needs clean-up of some kind

DIGITAL INPUT NOT CLEAN

- Scanned text and born-digital PDFs might contain:
  - photos, tables, graphics
  - layout or design information
  - disclaimers, copyright statements
  - headers, footers
  - column and page breaks
  - OCR errors
  - character encoding errors

- Semi-structured text: text with markup (HTML, XML, docx)

OCR EXAMPLE (HISTORICAL)



TO THE TRVLY HO-
NORABLE AND RIGHT WOR-
THY KNIGHT SIR THOMAS SMITH,
TREASVRER for the Colonies and Com-
panies of VIRGINIA: *and Gouernour of* Muſ-
couia, Eaſt-India, North-weſt Paſſage,
and SOMMER Ilands
Companies.

HONORABLE SIR, the wiſeſt of Men, or rather the wiſedome of God tells vs, that *there is a time for all things*: and that the great God, who at his owne will beganne Time it ſelfe, doth at his owne time beginne all things elſe: the fooliſhneſſe of men may aske and muſe why was this ſo ſoone, and that ſo late? but the wiſedome of God knowes what is fit for euery time: And ſurely amongſt the ſenſible ſignes, and euident demonſtrations of Gods all-gouerning prouidence, this is not the leaſt, that he brings not forth his mightie works altogether, but *makes every thing beautifull in his time.* And as in his creation he made not al at once, but produced them in their ſeuerall daies: ſo in his gubernation, he reueileth not the knowledge of all things in one Age, but diſcouers them in the ſeuerall ages of the World. And if man aske why God doth thus, holy *David* giues the anſwere; *The Lord hath ſo done his marvailous works, that they ſhould be had in remembrance*; for were they all in one age (ſuch is our corruption) they would bee leſſe obſerued and ſooner forgotten, but being declared in their ſeuerall times, euery Age finds matter to magnifie God; And therefore He *whoſe glorious name is to be praiſed for euer*, reueils ſome meruailous thing in euery generation, that ſo his name may be praiſed from Generation to Generation.

Eccleſ. 3. 1.

Eccl. 3. 11.

Pſal. 111. 4.

Pſal 72. 19.

TO'THE TRVLY HO.

NORABLE AND RIGHT Wok
THY KNIGHT SIR Tiiom^s SMITHY
YR E A S F R E R for the Colonies and Conj
panics of V i& G 1 N 1 A: aod Gouernsur of Mtif.
couia, Eaft-lildiaNorth-weltPaffagc,
and S 0 M M E 9 llands
COM
N 0 R A B L 11 S 1 It, the W121 Of Men, Ot
rather the wifedome of God tells vs, thatibere
~r a time for a# tkings: and that the great Gor4,
who at his owne will beganne Time it fcl&,
dothathisowne timebeginne all things elfe:
the-foolithneffe ofinen may askc and tnufc why
was this fo roone, and that fo late ? but the wifedome of God
knowes what is fit for euery time : And furely among(l the fen
fible fignes, and cuidcnt demonfirations of Gods all-gouerning
prouidence,this is not the leaft,that he brings not fbrth his migh
tie works altogether, but makgrvffY *thing boc#tifs.11 in Ur time. rs.*
And as in his creation he made not al at once,but produced them
in their feuerall daies : fo in his gubernation, hcreuedethnotthc
knowledge of all things in one Age, but difcouers them in the
fcuerall ages ofthe World. And if man aske why God doth thus,
holy Davidgiues the anfwerc; *The Lord both* ***fo done*** *Ur mar-*

HTML SOURCE

```
            site_slug: 'nunlv2'
        };
    </script>
    <!-- DPG Ad Framework -->
    <script async src="https://advertising-cdn.dpgmedia.cloud/web-advertising/11/8/1/advert-xandr.js" type="application/javascript"></script><script async data-advert-src="https://acdn.adnxs.com/ast/ast.js" data-advert-vendor
    <script id="gtm_datalayer">
        window.dataLayer = window.dataLayer || [];
        dataLayer.push({"site_name": "NU.nl", "site_location": "prod", "site_country": "NL", "page_zone": "muziek-artikel", "page_type": "article", "page_category": "entertainment", "page_subcategory": "muziek", "page_catego
        const pageViewId = window.crypto.getRandomValues(new Uint32Array(4)).join('');
        window.pageViewId = pageViewId;
        dataLayer.push({ pageViewId });
    </script>
    <!-- Google Tag Manager -->
    <script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
    new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
    j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
    'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
    })(window,document,'script','dataLayer','GTM-N2NWCN7');</script>
    <!-- End Google Tag Manager -->
<link rel="preconnect" href="https://myprivacy.dpgmedia.net">
<link rel="dns-prefetch" href="https://myprivacy.dpgmedia.net">
<link rel="preconnect" href="https://media.nu.nl">
<link rel="dns-prefetch" href="https://media.nu.nl">
<link rel="preload" href="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous" as="script" />
<link rel="preload" href="https://content.jwplatform.com/libraries/whqXCOFb.js" as="script" />
<link rel="preload" href="https://cdns.eu1.gigya.com/js/gigya.js?apiKey=3_7sfOtbY8gzx5ydlcXfJKWkcfkkzAnbMmbxVyRrSY57itzhCeED67Dv6CvQiYjY0X" as="script" />
<script type="text/javascript" src="/static/bundles/js/fastjs.f5a032d8.js" ></script>
        <script type="text/javascript">
            var NU = NU || {}; var static_url = '/static/'; var mediatool_base_url = 'media.nu.nl'; var blocks = {}; var JW_VIDEO_URL = 'https://content.jwplatform.com/manifests/{id}.m3u8'; var JW_LIVESTREAM_URL = 'https://c
        </script>
<script type="text/javascript" src="/static/CACHE/js/output.1f96a412198e.js"></script>
<script data-privacy-src="/static/vendor/vwo/vwo-v2.js" data-privacy-category="marketing"></script>
    <!-- Advertorial styling -->
    <!-- Advertorial styling -->
<script type="application/ld+json">
{
    "@context": "http://schema.org",
    "@type": "NewsArticle",
    "mainEntityOfPage": "https://www.nu.nl/muziek/6156772/meer-ruimte-voor-meerdaagse-evenementen-en-festivals.html",
    "headline": "Meer ruimte voor meerdaagse evenementen en festivals",
    "description": "Meerdaagse evenementen en festivals worden onder voorwaarden weer mogelijk, melden ingewijden maandag. De voorwaarden worden nog verder uitgewerkt. Daarvoor wordt ook gebruikgemaakt van de bevindingen van
    "url": "https://www.nu.nl/muziek/6156772/meer-ruimte-voor-meerdaagse-evenementen-en-festivals.html",
    "author": {
        "@type": "Organization",
        "name": "NU.nl",
        "sameAs" : [
            "https://www.facebook.com/nu.nl/",
            "https://www.linkedin.com/company/697052/",
            "https://twitter.com/NUnl",
            "https://plus.google.com/106943062990152739506"
        ]
    }
```


https://www.nu.nl/muziek/6156772/meer-ruimte-voor-meerdaagse-evenementen-en-festivals.html

CLEAN TEXT STORAGE

- Markup: meta-information in a text file that is clearly distinguishable from the textual content
  - In the case of XML and json, markup often provides useful information in text processing
  - We typically convert PDF to XML, using pdf-to-xml convertors
  - Also, benchmark data is often stored as XML

- Character Encoding: the way that a computer displays text in a way that humans can understand.

CHARACTER ENCODING

Universiteit Leiden

Suzan Verberne 2021

ASCII

- Character encoding: translates a string of 0s and 1s to a character
- ASCII is a 7-bit encoding based on the English alphabet

  1100001 a

  1100010 b

  1100011 c

- ASCII: American Standard Code for Information Interchange

HOW ABOUT THE REST OF THE WORLD?

- corpus linguistics (English)
- corpuslinguïstiek (Dutch)
- कोष भाषा विज्ञान (Hindi)
- اللسانيات الإحصا (Arabic)
- 語料庫語言學 (Chinese)
- בלשנות קורפוס (Hebrew)
- การ ศึกษา ภาษาศาสตร์(Thai)
- Цорпус Лингуистицс (Serbian)

UNICODE

- Universal standard for all writing systems (>100,000 characters)
  - Independent of platform, software, vendor
- Interpretation of the character is done by the implementation (e.g. UTF-8) in the software (e.g. editor, printer or web browser) that determines the actual rendering (size, shape, font, style)
- For maximum compatibility (forward and backward) we encode text in UTF-8 when we read and write them

READ AND WRITE UTF-8 IN PYTHON 3

```
with open(filename,'r',encoding='utf-8') as raw:
    text = raw.read()
```

```
with open(filename,'w',encoding='utf-8') as clean:
    clean.write(text)
```

https://docs.python.org/3/library/functions.html#open

DATA CLEANING IN PRACTICE

- Digitalization, data conversion & cleaning are the first steps in the text mining process. These steps are:
  - necessary
  - time consuming
  - error prone
  - often complicated
- When building a text mining pipeline for new (raw) data, data collection and cleaning is long and tedious step that is not to be underestimated!

EDIT DISTANCE

(SECTION 2.5 IN J&M)

Universiteit Leiden

Suzan Verberne 2021

WHY EDIT DISTANCE

For measuring string similarity:

- Spelling correction/normalization (think about normalizing the content of doctor's notes, user-generated content, or search queries)
  - E.g. 'graffe' – what word was meant?
  - 'giraffe' differs by only one letter ␣ most likely
  - 'grail' or 'graf' differ in more letters
- In bio-informatics: Align two sequences of nucleotides
  - AGGCTATCACCTGACCTCCAGGCCGATGCCC
  - TAGCTATCACGACCGCGGTCGATTTGCCCGAC

MINIMAL EDIT DISTANCE

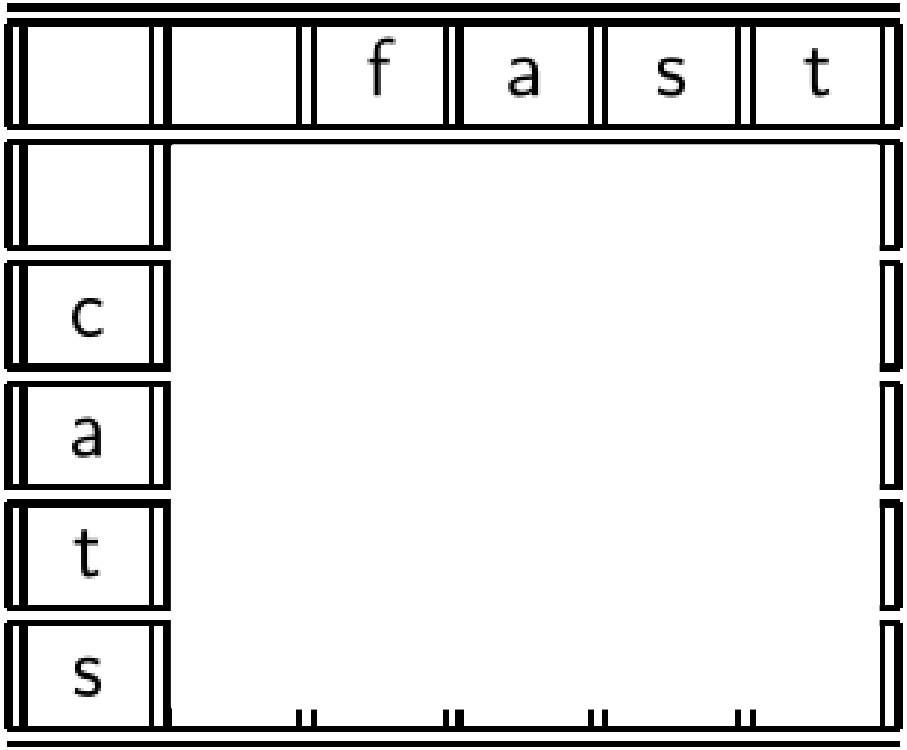
- the minimum edit distance between two strings is defined as the minimum number of editing operations (insertion, deletion, substitution) needed to transform one string into another
- Levenshtein distance: insertion, deletion and substitution all have a cost of 1.
  - dog-do: 1
  - cat-cart: 1
  - cat-cut: 1
  - cat-act: ?
  - cat-act: 2

COMPUTING MINIMAL EDIT DISTANCE

- "The space of all possible edits is enormous, so we can't search naively.
- However, lots of distinct edit paths will end up in the same state (string), so rather than recomputing all those paths, we could just remember the shortest path to a state each time we saw it.
- We can do this by using dynamic programming.
- Dynamic programming is the name for a class of algorithms that apply a table-driven method to solve problems by combining solutions to sub-problems."

(J&M, section 2.5.1)

LEVENSHTEIN DISTANCE: ALGORITHM



- Operations: insert (cost 1), delete (cost 1), substitute (cost 1), copy (cost 0)

LEVENSHTEIN DISTANCE: ALGORITHM

- Initialization
  D(i,0) = i
  D(0,j) = j

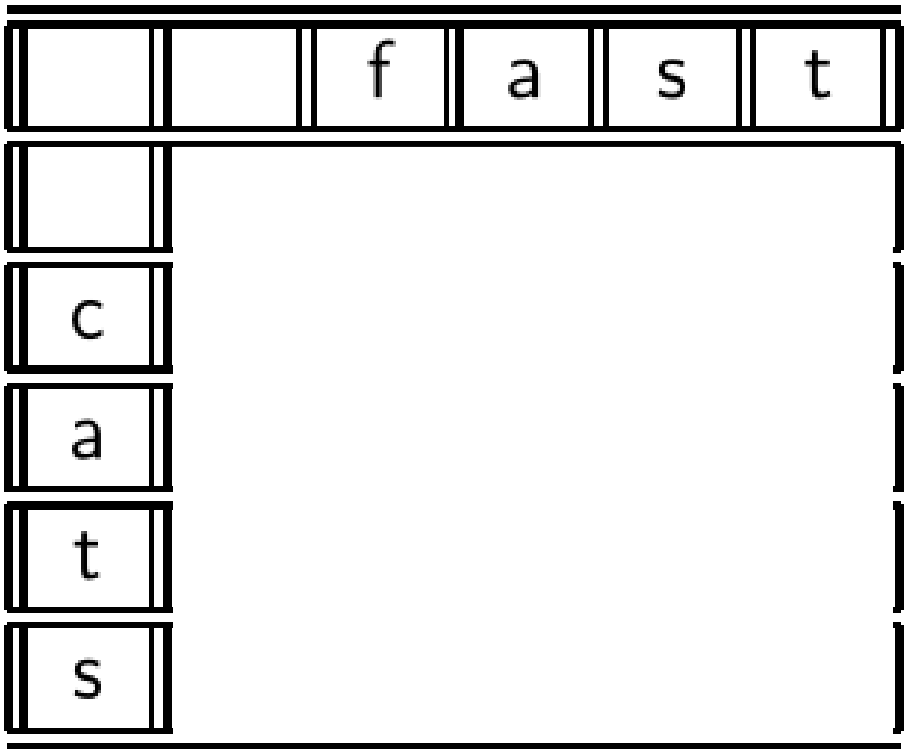
- Recurrence Relation:
  For each i = 1...M
    For each j = 1...N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{cases} \end{cases}$$

- Termination:
  D(N,M) is distance

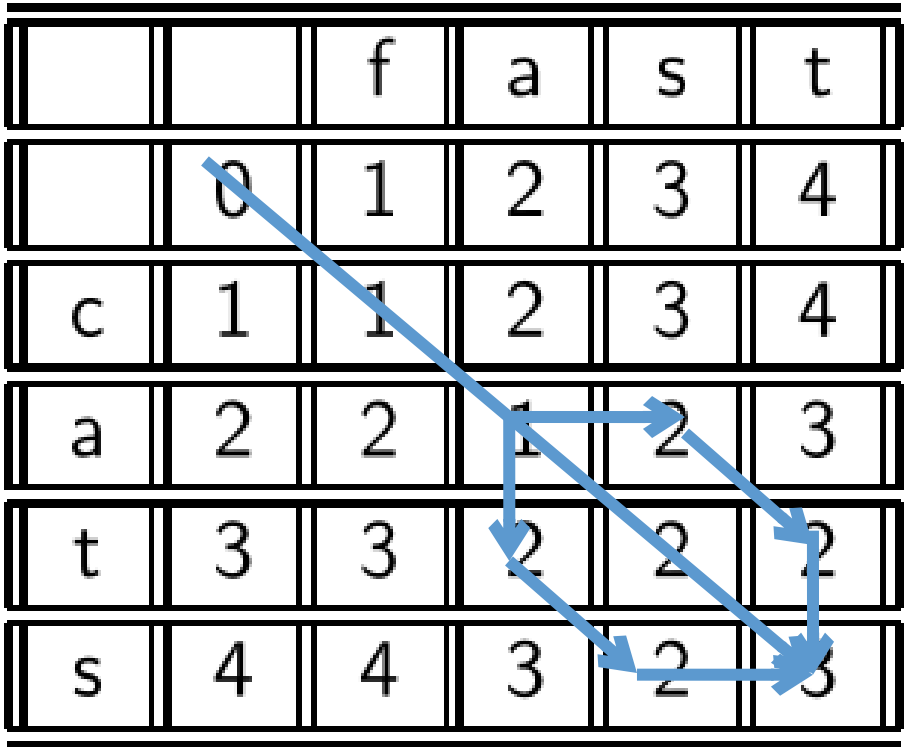
X = string 1
Y = string 2
i = index in X
j = index in Y
D(i,j) = value of cell (i,j)

LEVENSHTEIN DISTANCE: ALGORITHM



- Operations: insert (cost 1), delete (cost 1), substitute (cost 1), copy (cost 0)

LEVENSHTEIN DISTANCE: ALGORITHM



		f	a	s	t
	0	1	2	3	4
c	1	1	2	3	4
a	2	2	1	2	3
t	3	3	2	2	2
s	4	4	3	2	3

- Operations: insert (cost 1), delete (cost 1), substitute (cost 1), copy (cost 0)

LEVENSHTEIN DISTANCE: ALGORITHM

- Initialization
  D(i,0) = i
  D(0,j) = j

- Recurrence Relation:
  For each i = 1...M
    For each j = 1...N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{cases} \end{cases}$$

- Termination:
  D(N,M) is distance

X = string 1
Y = string 2
i = index in X
j = index in Y
D(i,j) = value of cell (i,j)

LEVENSHTEIN DISTANCE: ALGORITHM

cost	operation	input	output
1	substitute	c	f
0	(copy)	a	a
1	substitute	t	s
1	substitute	s	t
Total: 3			

cost	operation	input	output
1	substitute	c	f
0	(copy)	a	a
1	deletion	t	*
0	(copy)	s	s
1	insertion	*	t
Total: 3			

LEVENSHTEIN DISTANCE: EXERCISE

		a	n		a	c	t
	0	1	2	3	4	5	6
a	1						
	2						
c	3						
a	4						
t	5						

LEVENSHTEIN DISTANCE: EXERCISE

		a	n		a	c	t
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
	2	1	1	1	2	3	4
c	3	2	2	2	2	2	3
a	4	3	3	3	2	3	3
t	5	4	4	4	3	3	3

LEVENSHTEIN DISTANCE: EXERCISE

cost	operation	input	output
0	(copy)	a	a
1	insert	*	n
0	(copy)		
1	substitute	c	a
1	substitute	a	c
0	(copy)	t	t

REGULAR EXPRESSIONS

SECTION 2.1 IN J&M

Universiteit Leiden

Suzan Verberne 2021

WHY REGULAR EXPRESSIONS

- For finding patterns, e.g.
  - “[0-9][0-9][0-9][0-9] [A-Z][A-Z]”
  - “https?://\S+”
  - “\S+@\S+”

WHY REGULAR EXPRESSIONS

- Match/count patterns in a file/string
- Or extract the matched pattern

- Tool for making and debugging regular expressions: https://regex101.com/
- Verbose regular expressions: https://www.geeksforgeeks.org/verbose-in-python-regex/

EXAMPLE (FROM THE BOOK)

- Suppose we want to write a regular expression to find all occurrences of the English article *the*
- A simple (but incorrect) pattern might be: `/the/`
- Why is this incorrect?

EXAMPLE (FROM THE BOOK)

- Suppose we want to write a regular expression to find all occurrences of the English article *the*
- A simple (but incorrect) pattern might be: `/the/`
- Why is this incorrect?
  - We want to find both the and The
  - But not: their, apothecary, etc.
- Instead, if we would really want to cover all occurrences, and no non-relevant occurrences of the string, we need:

```
/(^|[^a-zA-Z])[tT]he([^a-zA-Z]|$)/
```

EXAMPLE (ALTERNATIVES)

```
from nltk.tokenize import word_tokenize
tokens = word_tokenize(raw_text)
print(tokens.count('the'))
```

```
import spacy
spacy_nlp = spacy.load("en_core_web_sm")
spacy_doc = spacy_nlp(raw_text.lower())
print(sum(token.text == 'the' for token in spacy_doc))
```

TOKENIZATION AND SENTENCE SPLITTING

SECTION 2.2-2.4 & 2.4.5 IN J&M

Universiteit Leiden

Suzan Verberne 2021

DEFINITIONS

- Token – An instance of a word or term occurring in a document
- Term – A token when used as feature (or in an index), generally in normalized form (e.g. lowercased)

- Token count is the number of words (running) in a collection/document; this includes duplicates
- Vocabulary size is the number of unique terms; the feature size when we use words as features

TOKENIZATION

- Tokenization: Split text in tokens
- (1) remove punctuation; (2) split on whitespaces characters
- Question: how would you want to tokenize these strings?
  1. Hewlett-Packard
  2. State-of-the-art
  3. aren't
  4. C++
  5. cheap San Francisco-Los Angeles fares
  6. 20/03/97
  7. 071 527 7043

TOKENIZATION WITH NLTK

```
from nltk.tokenize import word_tokenize
tokens = word_tokenize(raw_text, language= 'english')
```

https://www.nltk.org/api/nltk.tokenize.html

TOKENIZATION WITH SPACY

- https://spacy.io/usage/linguistic-features#how-tokenizer-works

- Customization for your task possible

Editable Code spaCy v3.0 · Python 3 · via Binder

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
for token in doc:
    print(token.text)
```

RUN

```
Apple
is
looking
at
buying
U.K.
startup
for
$
1
billion
```

STOP WORDS

- Stop words: extremely common words that don't carry any content
- Examples: *a, an, and, are, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, will, with*
- Stop word elimination commonly used in retrieval and classification
- But… For which cases is this problematic?

STOP WORDS

- Stop words: extremely common words that don't carry any content
- Examples: *a, an, and, are, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, will, with*
- Stop word elimination commonly used in retrieval and classification
- But… For which cases is this problematic?
  - "to be or not to be"
  - You might need stop words for multi-word terms, e.g. "King of Denmark", "Marks and Spencer"

STOP WORDS

- Advantages of stop word removal in classification and retrieval:
  - reduces dimensionality (number of features)
  - removes the noise of highly frequent words (better model)
- Disadvantages:
  - in some cases stop words do carry important information
  - stop words are needed for phrases

- Rule of thumb: remove stop words for large data sets and long documents; keep them in small data sets and short documents

SENTENCE SPLITTING

- Many cases can be solved with a relatively simple approach:
  - Sentences end with . ? ! followed by whitespace
- Challenges:
  - Abbreviations (‘Mrs. Doe’)
  - Names/initials that include punctuation marks (‘S. Verberne’)
  - Sentences without punctuation markers (headers/titles)
  - Line endings inside sentences (PDF conversion)

SENTENCE SPLITTING WITH NLTK

```
from nltk.tokenize import sent_tokenize
sentences = sent_tokenize(document, language='english')
```

https://www.nltk.org/api/nltk.tokenize.html

SENTENCE SPLITTING WITH SPACY

- https://spacy.io/usage/linguistic-features#sbd

Editable Code spaCy v3.0 · Python 3 · via Binder

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("This is a sentence. This is another sentence.")
assert doc.has_annotation("SENT_START")
for sent in doc.sents:
    print(sent.text)
```

RUN

```
This is a sentence.
This is another sentence.
```

LEMMATIZATION AND STEMMING

(SECTION 2.4.4 IN J&M)

Universiteit Leiden

Suzan Verberne 2021

BASIC WORD FORMS

- We might want to normalize specific word forms to the same term:
  - For example, it can be useful to have the term *bicycle* for each occurrence of either *bicycle* or *bicycles*

BASIC WORD FORMS

- We might want to normalize specific word forms to the same term:
  - For example, it can be useful to have the term *bicycle* for each occurrence of either *bicycle* or *bicycles*
- Advantages:
  - reduces the number of features
  - generalizes better, especially for small datasets

BASIC WORD FORMS

- We might want to normalize specific word forms to the same term:
  - For example, it can be useful to have the term *bicycle* for each occurrence of either *bicycle* or *bicycles*
- Advantages:
  - reduces the number of features
  - generalizes better, especially for small datasets
- Two types of basic word forms:
  - Lemma
  - Stem

LEMMA

- Lemma: dictionary form of a word
- For example:
  - Verbs: infinitive
    - ‘think’ for ‘thinks’, ‘thinking’, ‘thought’
  - Nouns: singular form
    - ‘mouse’ for ‘mice’
    - ‘computer’ for ‘computers’

STEM

- Stem: the portion of a word that:
  - is common to a set of (inflected) forms when all affixes are removed
  - is not further analyzable into meaningful elements, being morphologically simple

- The stem is the part of the word that ‘never’ changes even when morphologically inflected. It is not necessarily an existing word:
  - ‘comput' for forms ‘computer’, ‘computing’, ‘computers’, ‘compute’

STEM VS LEMMA

- ‘produced’
  - what is the lemma?
  - what is the stem?

STEM VS LEMMA

- ‘produced’
  - what is the lemma?
  - what is the stem?
- Answers:
  - the lemma is ‘produce’
  - the stem is ‘produc’

(think of ‘producing’ as one of the morphological forms of the verb)

- We almost always prefer lemmas over stems. Stemming can be effective for very small collections.

EXAMPLE

- Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- Lemmatizer: Such an analysis can reveal feature that be not easily visible from the variation in the individual gene and can lead to a picture of expression that be more biologically transparent and accessible to interpretation
- Stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

CONCLUSIONS

SUZAN VERBERNE 2021

PYTHON PACKAGES AND TOOLS

- Sklearn (http://scikit-learn.org) has built-in functionality for:
  - Tokenization
    - https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
  - Stop word removal (with option to supply your own list)
- NLTK (http://www.nltk.org/) and Spacy (https://spacy.io/) have functionality for:
  - Sentence splitting
  - Lemmatization and stemming
  - and additional pre-processing steps

HOMEWORK

- Read Jurafsky & Martin chapter 2: “Regular Expressions, Text Normalization, Edit Distance” (Brightspace)
- Complete this week’s exercise: preprocessing with spacy. https://course.spacy.io/en/chapter1 **(note: only chapter 1)**
- Contact address for the teaching assistants: tmcourse@liacs.leidenuniv.nl

AFTER THIS LECTURE…

- you know what issues to take into account when converting raw text to clean plain text
- you can explain the the difference between ASCII and Unicode; and why Unicode exists
- you can use regular expressions in Python and you know what the possibilities and challenges of regular expressions are
- you can describe the challenges of tokenization and sentence splitting
- you can explain the considerations for stop word removal
- you can define the difference between stemming and lemmatization
- you can compute the minimal edit distance between two strings using the matrix algorithm