JM2050 – Natural Language Processing

Introduction to text analysis

September 12th, 2024

Prof.dr.ir. U. Kaymak

Recap

- Challenges with text data
  - Ambiguity
  - Variation
  - World knowledge
  - Context
- Major types of learning
  - Supervised
  - Unsupervised
  - Reinforcement learning
- Performance measurement
  - Metrics derived from confusion matrix
- Balance between model fit and generalization

[2]

www.jads.nl

Jheronimus Academy of Data Science

Outline

- Descriptive text analysis
- Pre-processing text
- Regular expressions

www.jads.nl

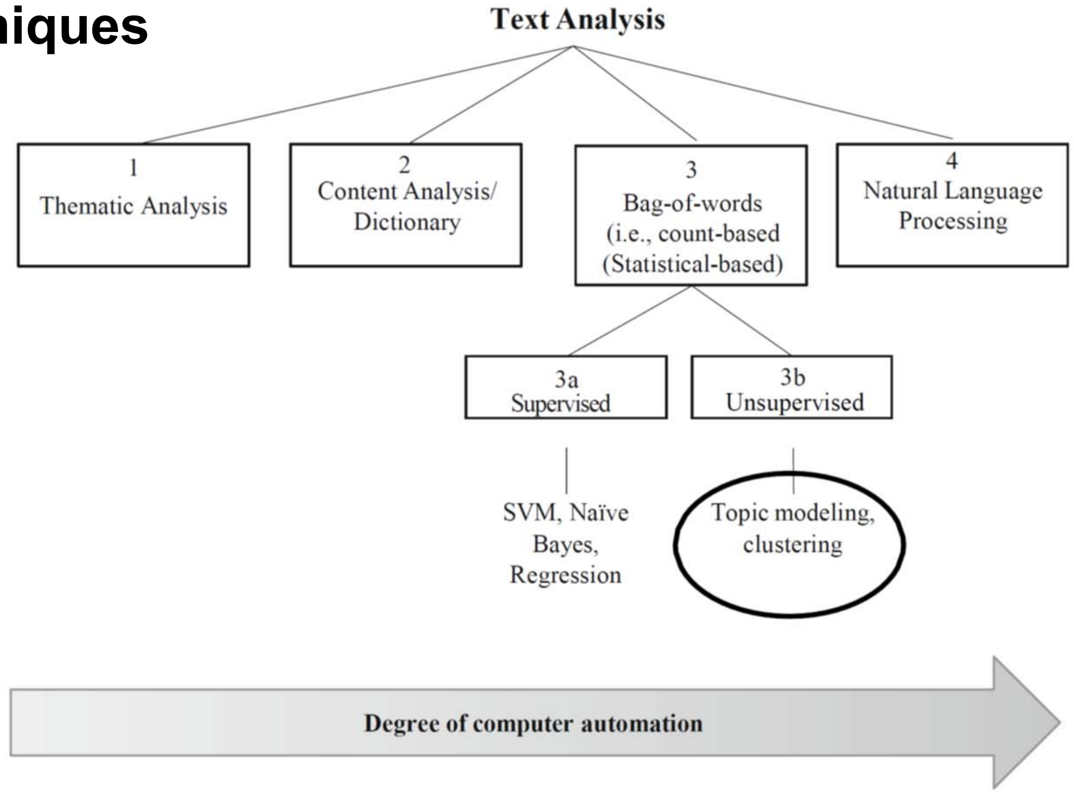
Jheronimus Academy of Data Science

Descriptive text analysis

Acknowledgement: slides adopted from K. Zervanou

Jheronimus Academy of Data Science

Text analysis techniques



[5]

(Banks et al., 2018)

www.jads.nl

Some common natural language processing tasks

Text classification	Information retrieval	Information extraction
spam filtering	recommender systems	Template-filling
topic modeling	search engine	named entity recognition (NER)
sentiment analysis	question answering	relationship extraction
	Summarization	ontology extraction

6

www.jads.nl

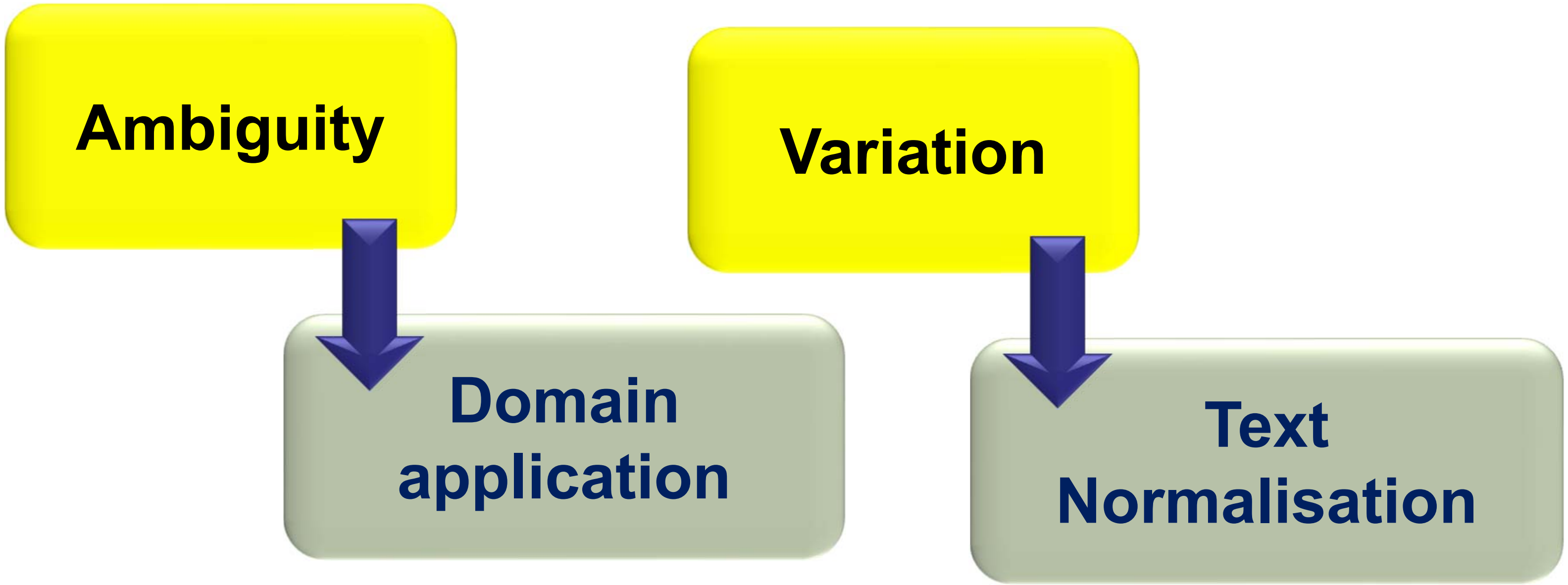
Basic terminology

- Text – a series of symbols/characters
- Token – a sequence of symbols (characters) that form a useful semantic unit for processing
- Document – a collection of tokens
- Corpus – a collection documents

[7]

www.jads.nl

Reduce ambiguity and variation: constraints & pre-processing



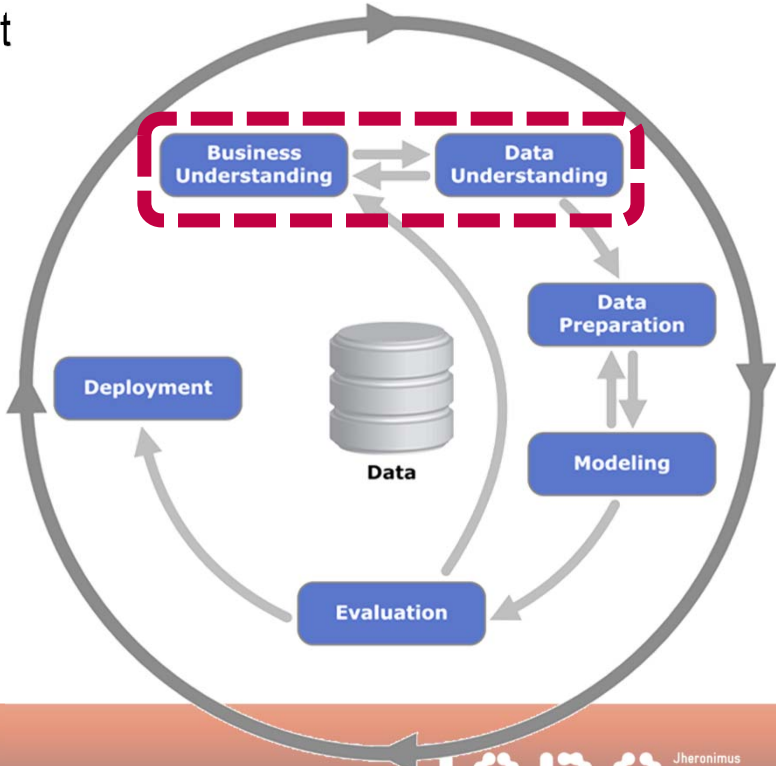
www.jads.nl

Jheronimus Academy of Data Science

Domain: *Text type or communication context*

- extra-linguistic/pragmatic document context

❖ *letters,*
❖ *tweets,*
❖ *chat,*
❖ *reports,*
❖ *news stories,*
❖ *scientific articles,…*



www.jads.nl

Domain: *application domain*

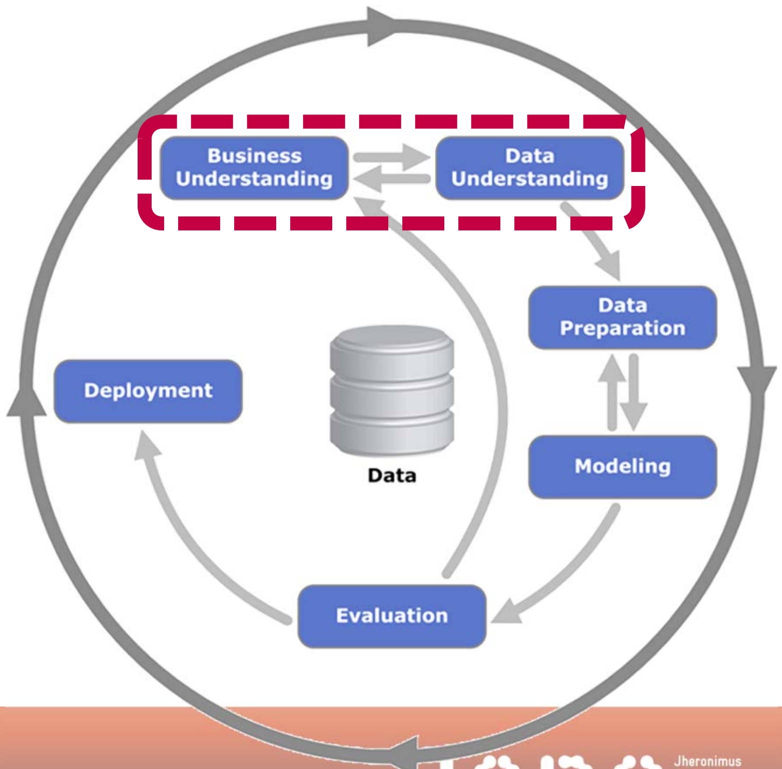
- application domain: area of application

❖ topics & content

❖ vocabulary use: terminology, jargon, general

❖ writing style: formal, informal, 😅 …

❖ language(s)



www.jads.nl

Domain: *Corpus characteristics*

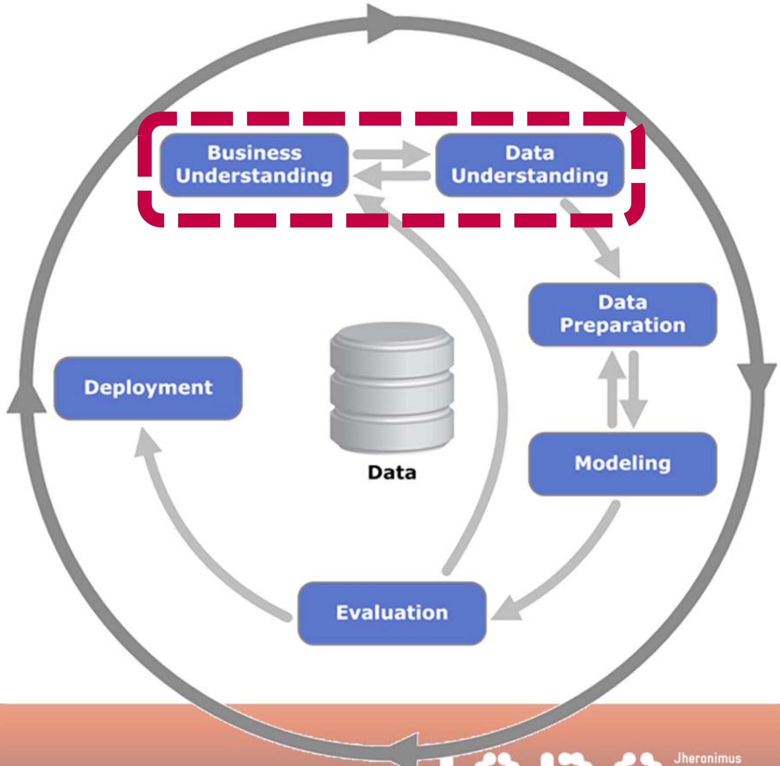
- ***Corpus:*** Document collection

❖ **text format:** annotations? Text, XML, HTML, …?

❖ **text encoding:** ASCII? UTF-8?

❖ **text unit(s)** of interest: documents? paragraphs? sentences? phrases?

❖ **text units length**



www.jads.nl

Jheronimus Academy of Data Science

Corpus characteristics: *Know your text data*

What You See Is **NOT** What You Get!

Corpus characteristics: *Know your text data*

Documents optimized for visual presentation or machine readability…

What is your text document format?	XML, HTML, text, pdf, MS word document,…
What is your information structure?	Plain text, XML, tabular data, combined/distributed information
What is your text encoding?	UTF-8, UTF-16, ISO 8859, …

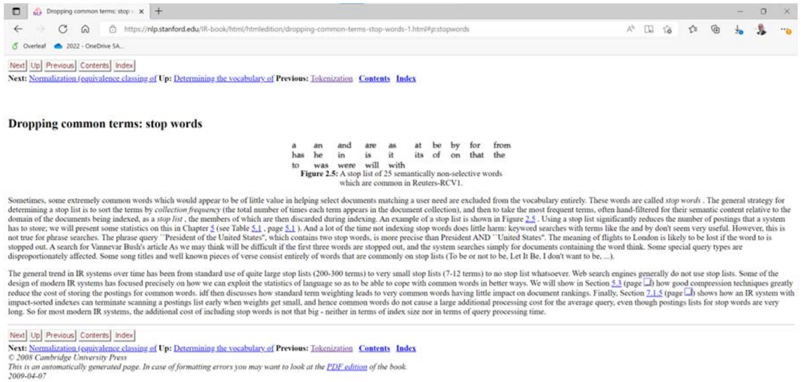
What You See Is **NOT** What You Get!

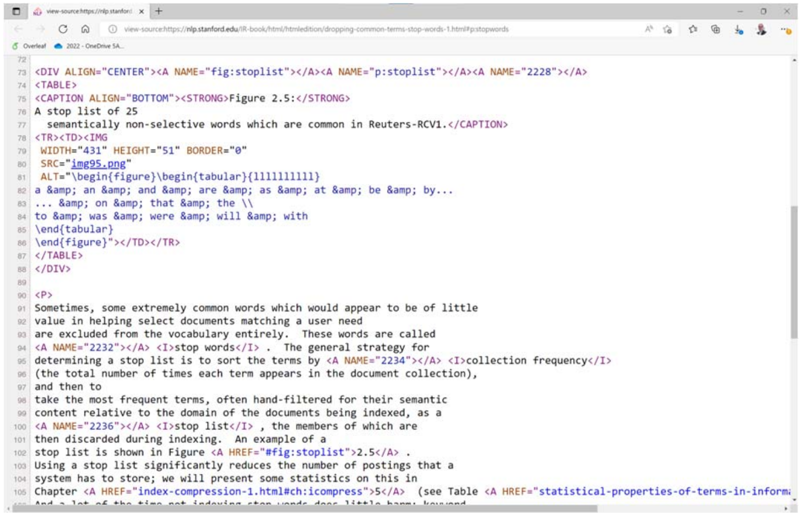
[13]

www.jads.nl

Jheronimus Academy of Data Science

Example: processing HTML text





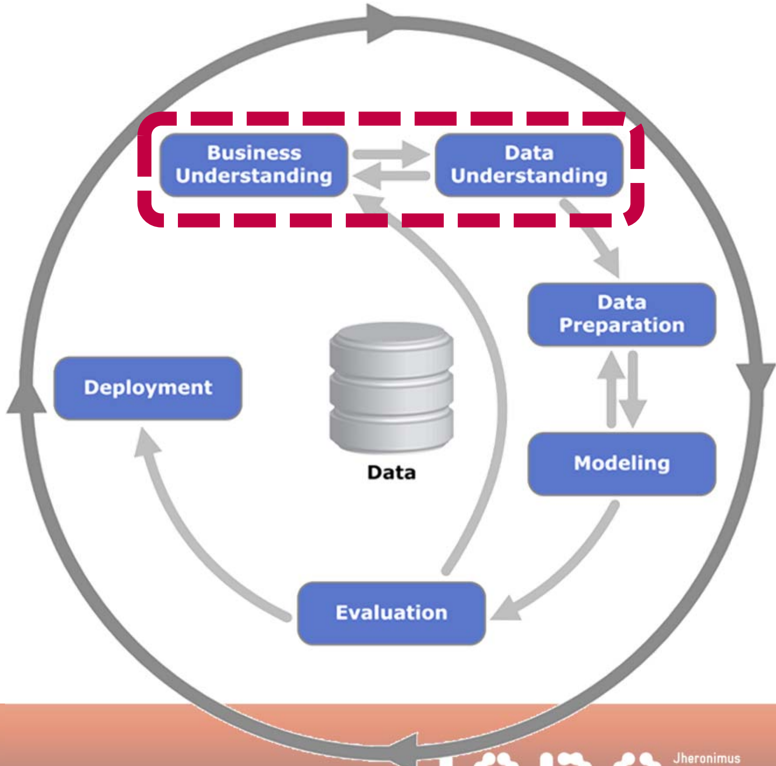
[14]

www.jads.nl

Domain: *Corpus characteristics*

Corpus: Document collection

- **vocabulary** richness/variation
  (i.e. unique vs. total "words" number)
- document **structure**,
  e.g. CS articles, wikipedia, etc.
- corpus **homogeneity**,
  e.g. wikipedia, news



www.jads.nl

Jheronimus Academy of Data Science

Corpus data understanding: *Descriptive statistics*

- How many documents?
- How many “words”?
- Which “words” occur very frequently?
- How much lexical variation do your texts have?
  - type / token ratio: unique words vs. total words
- Average sentence length?
- Average document length?

www.jads.nl

Corpus data understanding

- Descriptive statistics are not enough
- Explore: read some documents yourself, **look for patterns**

Domain considerations

- **Data size,** *big? small?*
- **Private and sensitive data,** *e.g. military, police, healthcare, banking*
- **Ethical issues** *e.g. fake news promotion, user exploitation, surveillance*

- **data storage**
- **processing memory limitations**
- **type of processing**
- **available tools & resources**
- **ethical & legal constraints**

[18]

www.jads.nl

Jheronimus Academy of Data Science

Pre-processing text

Acknowledgement: slides adopted from K. Zervanou

[19]

Jheronimus Academy of Data Science

Pre-processing

- Document filtering
- OCR
- Document structuring
- Text cleaning
- Tokenisation
- Text normalization
- Stop-word removal

[20]

Document filtering

- Select relevant documents, *e.g.*
  - retrieve tweets about coronavirus using #coronavirus tag
  - retrieve wikipedia pages about TV series

www.jads.nl

Jheronimus Academy of Data Science

Optical Character Recognition (OCR)

CONVERT SCANNED TEXT IMAGES INTO TEXT

MAY INTRODUCE A LOT OF ERRORS

[22]

www.jads.nl

OCR

- Mixing **headlines** with plain text
- Advertisements
- Image captions

Subscribe and get your weekday paper for 50% less

THE TIMES

Emerge in style

Essential fashion for facing the outside world

INSIDE TIMES2

The 10 best interviews in TV history

Andrew Billen

Mass testing blitz as PM plans easing of lockdown

Millions to receive kits amid national campaign

I'm a hostage, says Dubai ruler's daughter

IN THE NEWS

China 'critical threat'

Stark loses in court

Free-speech payouts

Iran test for Biden

Bitcoin hits $50,000

Root apologises

[23]

www.jads.nl

Jheronimus Academy of Data Science

Document structuring

- Identify & select document sections,
  e.g. Abstract, Title, Conclusion


'I fight for children day in, day out'
Sport

The Guardian

5 January 2021
£2.20 | €2.70
From £1.75 for subscribers

PM imposes new lockdown and warns worst yet to come

- Schools in England closed, with exams unlikely to go ahead

www.jads.nl

Jheronimus Academy of Data Science

Text cleaning

Remove non-relevant:

✓metadata (e.g. author, edit history, etc.)

✓mark-up (e.g. HTML, javascript code etc.)

✓headers, redundant spaces

✓intervening page numbers, footnotes

✓tables

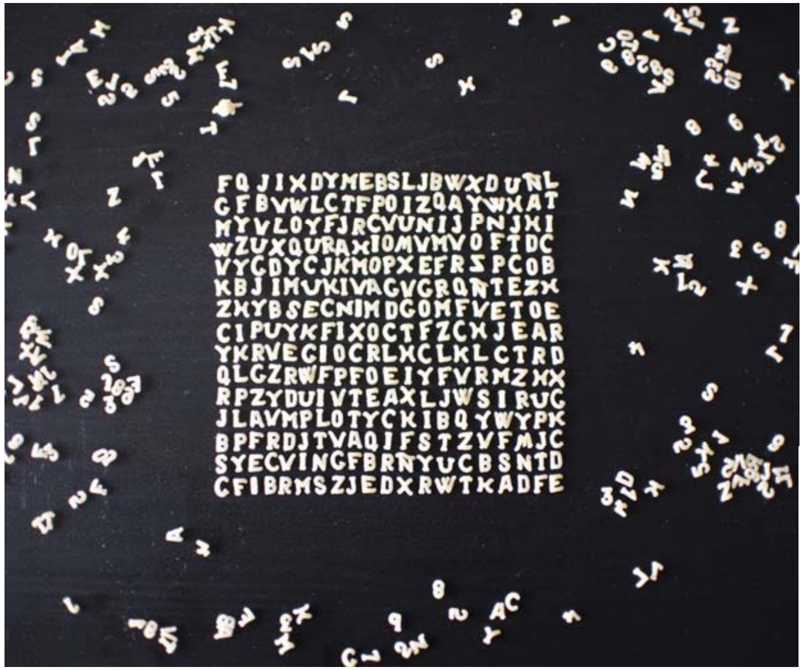
✓duplicate documents

✓noise / non-word characters

[25]

Text cleaning: “corrections”

- hyphenated words
- spelling correction
- OCR error correction
- convert irregular language
  e.g. abbreviations
- character encoding
- anonymise



[26]

www.jads.nl

Jheronimus Academy of Data Science

Tokenisation

- split text into tokens ("words") – based on spaces & punctuation

From 1997 to 2011, the number of adults aged 18 years or older with diagnosed diabetes who reported taking diabetes medication increased for those taking either insulin, pills, or both.

Token
From
1997
to
2011
,
the
number

Diabetes Patients Medication Status
Source: Centers for Disease Control and Prevention (CDC)
https://www.cdc.gov/

www.jads.nl

Jheronimus Academy of Data Science

Tokenisation: *issues*

- **multi-word tokens?**
  - New York, stock exchange
- **what about punctuation?**
  - E.U., EU
  - COVID-19, Murphy’s law
  - $4.4 billion,18.5°C, 31/03/2020…

- **Assumption: words separated by non-letters**
- Not always true **but** practical

[28]

www.jads.nl

Jheronimus Academy of Data Science

Text normalisation

If needed reduce vocabulary variation by

- ✓ removing numbers

  (but what if you need to find dates & amounts?)
- ✓ removing punctuation & special characters (e.g. @#, -, *, …)

  (but what if you need to identify sentiment?)
- ✓ convert into lower case

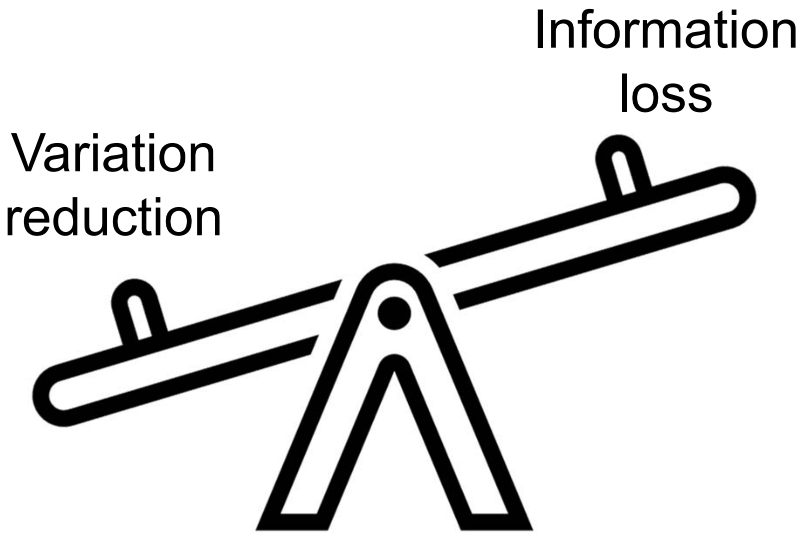
  (but what if you need to find names of people & products?)
- ✓ lemmatization or stemming

  (but what if you unnecessarily increase ambiguity?)

www.jads.nl

Jheronimus Academy of Data Science

Text normalisation



Some points to consider

- Does my corpus have a lot of variation?
  - What is the ratio of unique tokens vs. my total token number?
- Is it likely that I lose information that I need?
- How is modelling affected by the tokens I remove or normalize?
- Do I remove important text context?

30

Stemming

Assumption: a fixed number of characters ending a token are suffixes

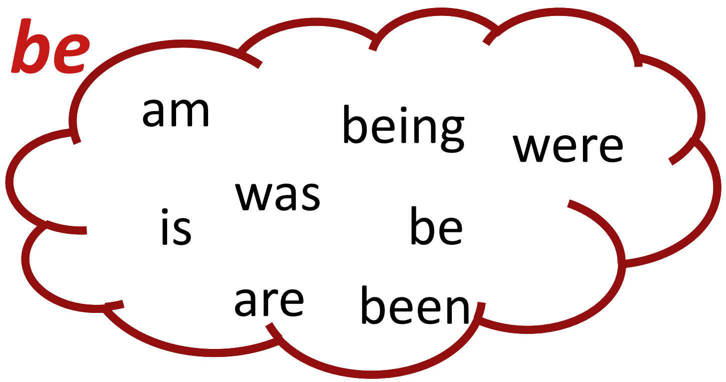
Token	Stem
worker	work
working	work
worked	work

Token	Stem
are	ar
requirement	requir
aged	ag
afterwards	afterward

www.jads.nl

Lemmatisation

- convert word to dictionary lemmas
- requires dictionary & part-of-speech
- result linguistically correct



www.jads.nl

Jheronimus Academy of Data Science

Removal of (stop) words

- Where is the information required?
- **If needed,** filter out "non-informational" text
  - function words (e.g. could, will, be, and, both, in…)?
  - Stop words? (all very common words in general language)
  - all verbs?
  - all words except nouns & adjectives?

www.jads.nl

Jheronimus Academy of Data Science

List of common stop words in English

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					

Figure 2.5: A stop list of 25 semantically non-selective words which are common in Reuters-RCV1.

https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html#p:stopwords

NLTK stop words

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

[34]

www.jads.nl

Jheronimus Academy of Data Science

Recap

Reduce ambiguity and variation

- Ambiguity → domain application
- Variation → text normalization

Text normalization

- removing numbers
- removing punctuation & special characters
- convert into lower case
- lemmatization or stemming

[35]

www.jads.nl

Jheronimus Academy of Data Science

Recap

Pre-processing text

- Document filtering
- OCR
- Document structuring
- Text cleaning
- Tokenization
- Text normalization
- Stop word removal

[36]

www.jads.nl

JADS Jheronimus Academy of Data Science

Rule-based pre-processing of text: regular expressions

Regular expression material based on slides from Jurafsky et al. (2020)

Jheronimus Academy of Data Science

[37]

Regular expressions

- A formal (regular) language for specifying text strings
- How can we search for any of these?
  - woodchuck
  - woodchucks
  - Woodchuck
  - Woodchucks

[38]

www.jads.nl

Regular Expressions: Disjunctions

- Letters inside square brackets []

Pattern	Matches
[wW]oodchuck	Woodchuck, woodchuck
[1234567890]	Any digit

- Ranges [A-Z]

Pattern	Matches	
[A-Z]	An upper case letter	Drenched Blossoms
[a-z]	A lower case letter	my beans were impatient
[0-9]	A single digit	Chapter 1: Down the Rabbit Hole

www.jads.nl

Regular Expressions: Negation in Disjunction

- Negations `[^Ss]`
  - Carat means negation only when first in []

Pattern	Matches	
`[^A-Z]`	Not an upper case letter	`Oyfn pripetchik`
`[^Ss]`	Neither 'S' nor 's'	`I have no exquisite reason"`
`[^e^]`	Neither e nor ^	`Look here`
`a\^b`	The pattern a carat b	`Look up a^b now`

www.jads.nl

Regular Expressions: More Disjunction

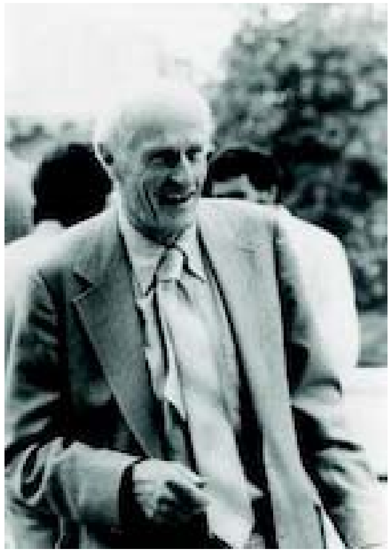
- Woodchuck is another name for groundhog!
- The pipe | for disjunction

Pattern	Matches
groundhog|woodchuck	groundhog
groundhog|Woodchuck	Woodchuck
a|b|c	= [abc]
[gG]roundhog|[Ww]oodchuck	Woodchuck

www.jads.nl

Regular Expressions: ? *+.

Pattern	Matches	
colou?r	Optional previous char	color colour
oo*h!	0 or more of previous char	oh! ooh! oooh! ooooh!
o+h!	1 or more of previous char	oh! ooh! oooh! ooooh!
o{2}h!	Precisely 2 times previous char	ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun begun beg3n



Stephen C. Kleene

www.jads.nl

Regular Expressions: Anchors ^ $

Pattern	Matches
`^[A-Z]`	Palo Alto
`^[^A-Za-z]`	1 "Hello"
`\.$`	The end.
`.$`	The end? The end!

- ^ : starts with
- $: ends with

[43]

www.jads.nl

Jheronimus Academy of Data Science

Example

- Find me all instances of the word “the” in a text.

  `the`

  Misses capitalized examples

  `[tT]he`

  Incorrectly returns `other` or `theology`

  `[^a-zA-Z][tT]he[^a-zA-Z]`

  `[^a-zA-Z]?[tT]he[^a-zA-Z]`

www.jads.nl

Errors

- Note that we fixed two kinds of errors:

1. Matching strings that we should not have matched (there, then, other)

   False positives (Type I errors)

2. Not matching things that we should have matched (The)

   False negatives (Type II errors)

www.jads.nl

Errors cont.

- In NLP we are always dealing with these kinds of errors.
- Reducing the error rate for an application often involves two antagonistic efforts:
  - Increasing accuracy or precision (minimizing false positives)
  - Increasing coverage or recall (minimizing false negatives)

www.jads.nl

Power of regular expressions

- Regular expressions play a surprisingly large role
  - Sophisticated sequences of regular expressions are often the first model for any text processing task

- For hard tasks, we use machine learning classifiers
  - But regular expressions are still used for pre-processing, or as features in the classifiers
  - Can be very useful in capturing generalizations

Splits and substitutions

- txt = “The rain is another matter in the theology in Spain.”
- Split at white spaces: `re.split(“\s”,txt)`
```
['The', 'rain', 'is', 'another', 'matter', 'in', 'the',
'theology', 'in', 'Spain.']
```
- Split when string does not contain word characters: `re.split(“\W”,txt)`
```
['The', 'rain', 'is', 'another', 'matter', 'in', 'the',
'theology', 'in', 'Spain', '']
```
- Substitute a pattern with a string
```
x = re.sub("\sthe\s"," a ", t)
The rain is another matter in a theology in Spain.
```

www.jads.nl

Capture Groups

- Say we want to put angles around all numbers:

  the 35 boxes → *the <35> boxes*

- Use parens () to "capture" a pattern into a numbered register (1, 2, 3...)

- Use \1 to refer to the contents of the register

```
re.sub("([0-9]+)", "<\\1>", txt)
```

www.jads.nl

Jheronimus Academy of Data Science

Capture groups: multiple registers

```
the (.*)er they (.*), the \\1er we \\2
```

Matches

the faster they ran, the faster we ran

But not

the faster they ran, the faster we ate

But suppose we don't want to capture?

Parentheses have a double function: grouping terms, and capturing

Non-capturing groups: add a ?: after paren:

`(?:some|a few) (people|cats) like some \\1`

matches

`some cats like some cats`

but not

`some cats like some some`

www.jads.nl

Lookahead assertions

- `(?= pattern)` is true if pattern matches, but is **zero-width; doesn't advance character pointer**
- `(?! pattern)` true if a pattern does not match
- How to match, at the beginning of a line, any single word that doesn't start with "Volcano":

`^(?!Volcano)[A-Za-z]+`

```
t = "Some patterns do not look like a volcano."
print(re.search("^(?!Volcano)[A-Za-z]+", t))
<re.Match object; span=(0, 4), match='Some'>
```

Example with repeating numbers

- text = ‘Some number 7785 and another number 34 with a digit 9.’
- pattern = ‘[0-9]{2}’
  print(re.findall(pattern,text))
  ['77', '85', '34’]

  Find two consecutive digits, continue after match
- pattern = ‘[0-9]{3}’
  print(re.findall(pattern,text))
  [‘778’]

  Find three consecutive digits, continue after match
- pattern = ‘[0-9]{2,3}’
  print(re.findall(pattern,text))
  [‘778', '34’]

  Find two to three consecutive digits, continue after match

Greedy search: try to find the longest match

[53]

www.jads.nl

Jheronimus Academy of Data Science

Example with repeating numbers and look ahead assertion

- text = ‘Some number 7785 and another number 34 with a digit 9.’
- pattern = ‘(?=([0-9]{2}))’
  print(re.findall(pattern,text))
  ['77', ‘78’, '85', '34’]

  Find two consecutive digits, continue with next character
- pattern = ‘(?=([0-9]{3}))’
  print(re.findall(pattern,text))
  [‘778’, ‘785’]

  Find three consecutive digits, continue with next character
- pattern = ‘(?=([0-9]{2,3}))’
  print(re.findall(pattern,text))
  [‘778’, ‘785’, ‘85’, '34’]

  Find two to three consecutive digits, continue with next character

Greedy search: try to find the longest match

Exercise

- text = ‘Some number 7785 and another number 34 with a digit 9.’
- pattern = "\\b[0-9]{2}\\b“

- Determine what the Python command

  print(re.findall(pattern, text))

  will return
- What does \\b do in the pattern? (search on the Internet)

[55]

Simple rule-based system: ELIZA

- Early NLP system that imitated a Rogerian psychotherapist
  - Joseph Weizenbaum, 1966.

- Uses pattern matching to match, e.g.,:
  - `"I need X"`

and translates them into, e.g.

  - `"What would it mean to you if you got X?`

Simple Application: ELIZA

Men are all alike.
IN WHAT WAY

They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED

www.jads.nl

Jheronimus Academy of Data Science

How ELIZA works

- s/.* I'M (depressed|sad) .*/I AM SORRY TO HEAR YOU ARE \1./
- s/.* I AM (depressed|sad) .*/WHY DO YOU THINK YOU ARE \1?/
- s/.* all .*/IN WHAT WAY?/
- s/.* always .*/CAN YOU THINK OF A SPECIFIC EXAMPLE?/

www.jads.nl

Jheronimus Academy of Data Science