

Digital Humanities & History

METODE 3 @ SDU/Dept. of History

github.com/kln-courses/hist-3

Kristoffer L Nielbo

knielbo@sdu.dk

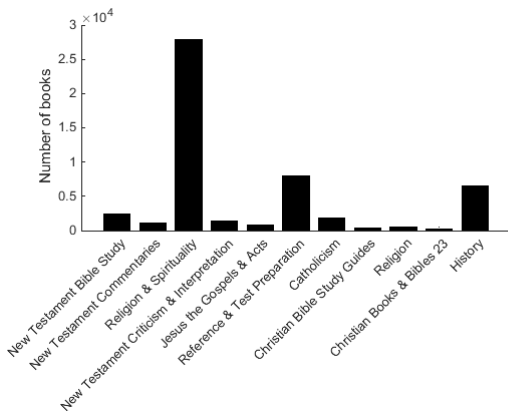
knielbo.github.io

Department of History | University of Southern Denmark



- domain knowledge in history, language, literature &c combined with microscopic and (predominantly) qualitative analysis of human cultural manifestations

Gospel of Marc (KJV) ~ 16500 words in 16 chp. on 11 p.

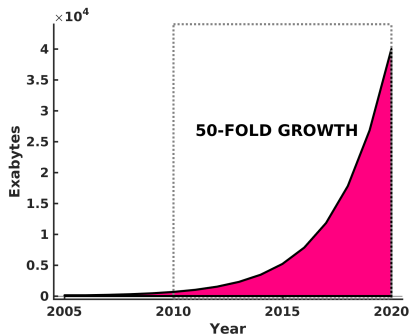


'from the dawn of civilization until 2003, humankind generated five exabytes of data. Now we produce **five exabytes every two days** ... and the pace is accelerating'

Eric Smith (Google)

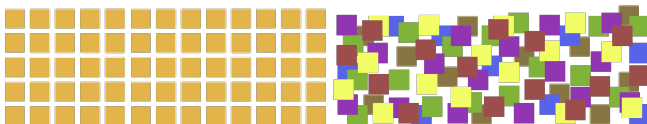
'increasingly, scientific breakthroughs will be powered by advanced computing capabilities that help researchers manipulate and explore massive datasets'

Jim Gray (Fourth Paradigm)



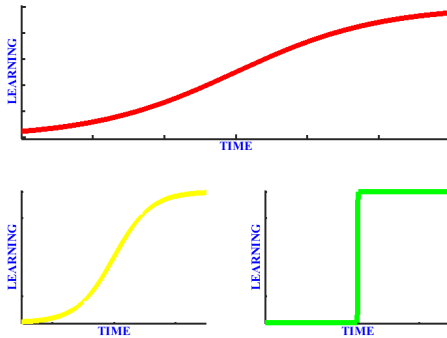
computational sciences are entering the exa-scale era
+
digital technologies are disruptive on a new scale

data \sim objects that are described over a set of (qualitative or quantitative) features



fundamental difference between **structured** data and **unstructured** data

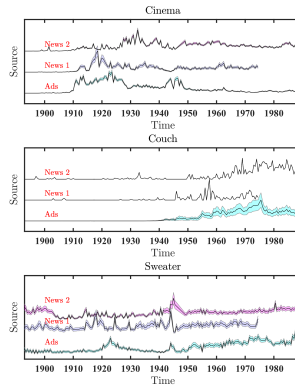
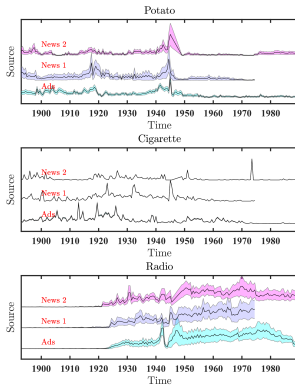
- word processing files, pdfs, emails, social media posts, digital images, video, and audio
- today $> 80\%$ of all data are unstructured
- increased demand for expertise from culture, media and linguistic domains



every knowledge-intensive industry have to “break” the learning curve

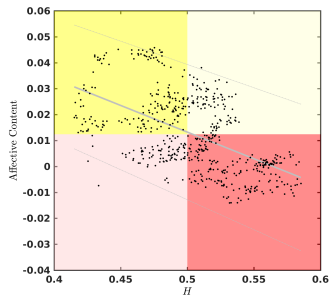
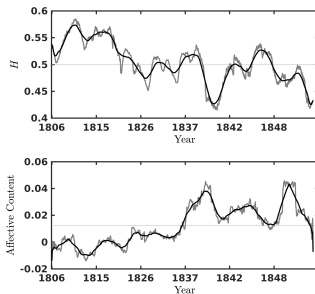


- domain knowledge in history, language, literature &c needs to scale, if we want to maintain our cultural knowledge base



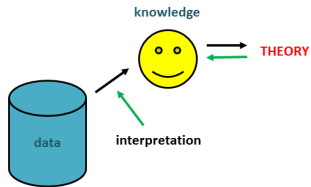
Digital history and media studies

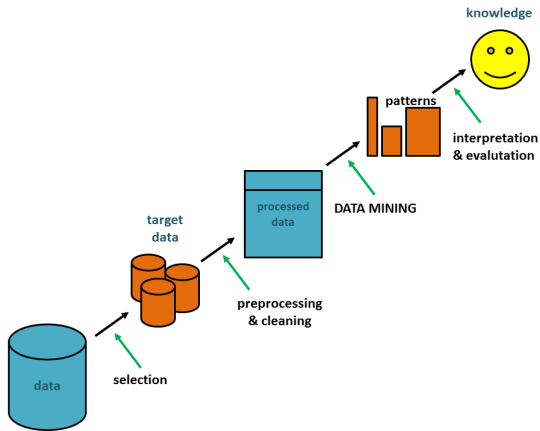
- prerequisite: humanistic domain experts that use content analysis
- source digitization (newspapers) og super computing change resolution and scale
- technologies create new standards for the domains involved
- share technology, but not data!

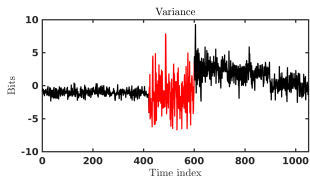
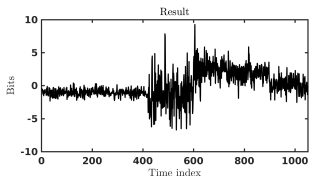
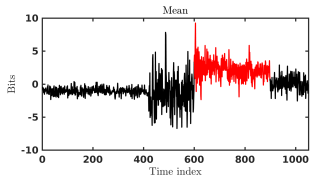
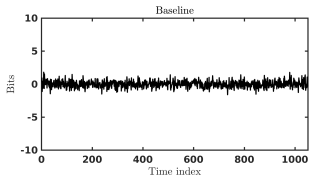


Computational literary history

- prerequisite: humanistic domain experts that study writers and literary periods
- high quality digitization of writers, annotation and NLP changes perspective and scale
- technologies that are creating new standards
- sharing of technology and data







Perspective: “Predictive History” ~ Culture Analytics

- given enough data, we can use past knowledge to predict future trends
- linked archives, news databases, social media ...
- knowledge of these technologies become imperative for critical use and assessment
- BUT we need free data access and mobility → OPEN SCIENCE

```
1 if questions:
2     try:
3         answer()
4     except RuntimeError:
5         pass
6 else:
7     print "break"
```

INTERVENTION|from the console

GUI → CLI

- novice-friendly visual approach to computer interaction w. a fast learning curve **ERROR**
- expert-friendly text-based approach to computer interaction w. ++freedom **VALID**
- **CONFLICT** break the learning curve through training intensive, non-intuitive, and specialized tools

- a `shell` is a program whose primary purpose is to read commands and run other programs
- the `shell`'s main advantages are its high action-to-keystroke ratio, its support for automating repetitive tasks, and its capacity to access networked machines
- the `shell`'s main disadvantages are its primarily textual nature and how cryptic its commands and operation can be

```
root@kali:~/Documents#  
root@kali:~/Documents# head genesis  
The first Book of Moses, called Genesis  
... (1:1) In the beginning God created the heaven and the earth. (1:2)  
And the earth was without form, and void, and darkness: [and] upon the  
face of the deep. And the Spirit of God moved upon the face of the  
waters.  
... (1:3) And God said, Let there be light: and there was light. (1:4)  
And God saw the light, that [it was] good: and God divided the light
```

`PS1='$ '` sets prompt string in console to

1

```
$
```

prompt indicates that the shell is waiting for input

1 \$ whoami

2 kln

3 \$

user ID or who the shell thinks you are

`whoami`

- 1 finds a program called `whoami`
- 2 runs that program
- 3 displays that program's output
- 4 displays a new prompt to tell us that it's ready for more commands

unknown command

```
1 $ somecommand
2 somecommand: command not found
3 $
```

- the shell runs other programs, so it does not work if the program does not exist

print working directory - current default directory

```
1 $ pwd
2 /home/kln
3
4 $ a=$(pwd)
5 $ echo "current wd is: $a"
6 current wd is: /home/kln
```

the path to the **home** directory varies between operating systems:

- [linux] /home/yourname
- [mac] /Users/yourname
- [windows] C:\Users\yourname

tokenization - unigrams

```
1 $ tr -sc "A-Za-z" "\n" < 2017-Trump.txt
```

sort in alphabetic order

```
1 $ tr -sc "A-Za-z" "\n" < genesis | sort
```

uniq - lexicon of document

```
1 $ tr -sc "A-Za-z" "\n" < 2017-Trump.txt | sort | uniq  
2 $ tr -sc "A-Za-z" "\n" < 2017-Trump.txt | sort | uniq -c  
3 $ tr -sc "A-Za-z" "\n" < 2017-Trump.txt | sort | uniq -c > lexicon.txt  
4 ...
```

tired of cryptic commands and operations from the command line?

luckily we have:



```
1 >>> import gensim, nltk, polyglot, spacy
2 >>> from adl.util import thefunctionthattrulesthemall
3 >>> thefunctionthattrulesthemall("yourfile.dat")
```

and:



```
1 > libs b<- c("mallet", "tidyverse", "tm", "syuzhet")  
2 > lapply(libs, require, character.only = TRUE)  
3 > thefunctionthattrulesthemall("yourfile.dat")
```

```
1 if questions:
2     try:
3         answer()
4     except RuntimeError:
5         pass
6 else:
7     print "thank you"
```