Digital Humanities & Historical Research

@ SDU/Dept. of History

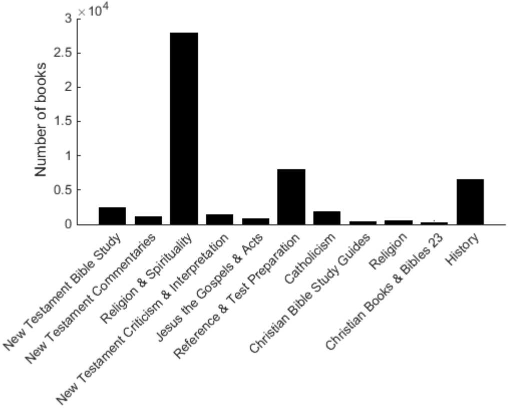
github.com/kln-courses/hist-3

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.

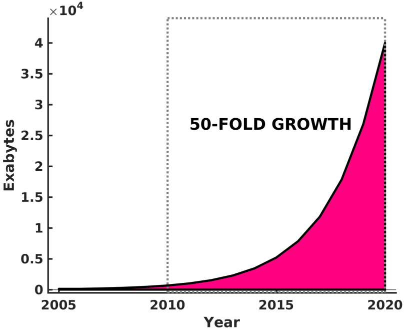


‘from the dawn of civilization until 2003, humankind generated five exabytes of data. Now we produce five exabytes every two days ... and the pace is accelerating’

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)

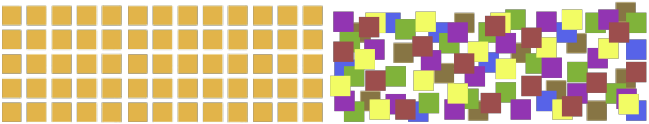


computational sciences are entering the exa-scale era

+

digital technologies are disruptive on a new scale

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

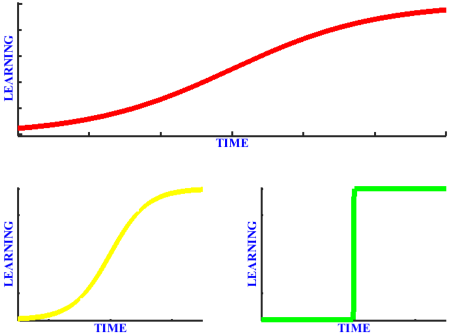


fundamental difference between **structured** data and **unstructured** data

- word processing files, pdfs, emails, social media posts, digital images, video, and audio

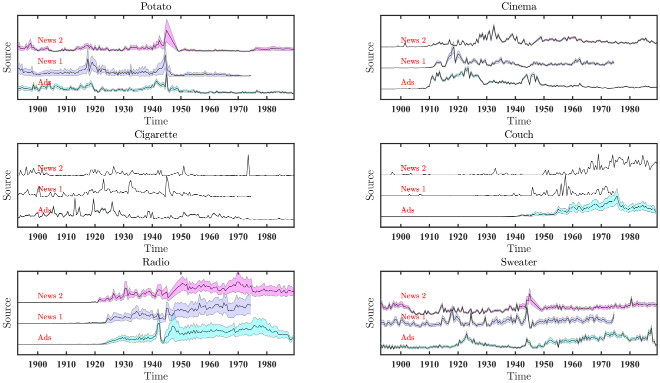
- today $> 80\%$ of all data are unstructured

- increased demand for expertise from culture, media and linguistic domains



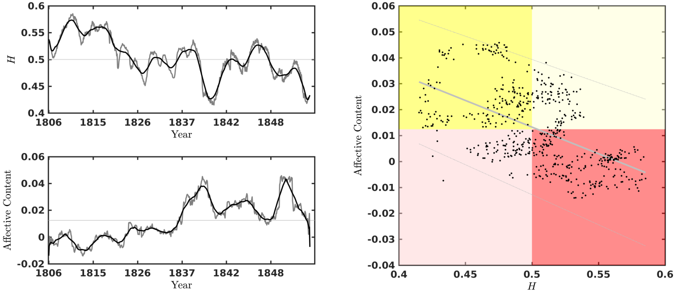
every knowledge-intensive industry have to “break” the learning curve

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



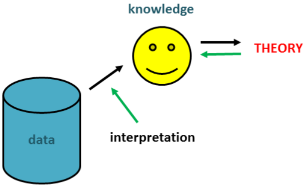
Digital history and media studies

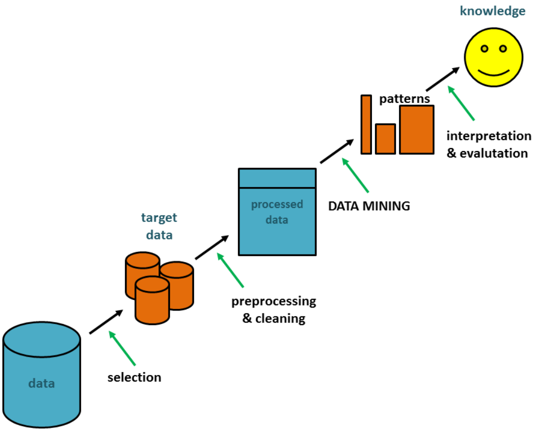
– prerequisite: humanistic domain experts that use content analysis
– source digitization (newspapers) og super computing change resolution and scale
– technologies create new standards for the domains involved
– share technology, but not data!

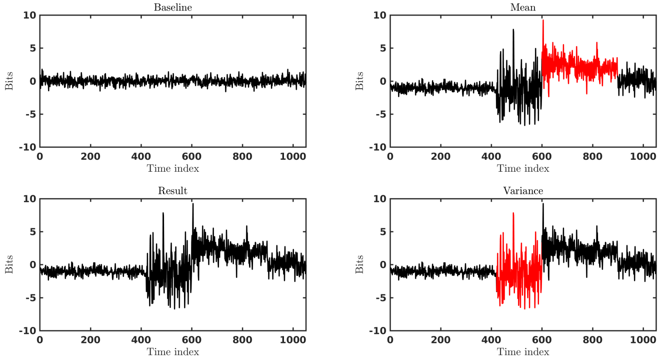


Computational literary history

– 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 tehcnology and data


knowledge
THEORY
data
interpretation


knowledge
patterns
interpretation
& evalutation
processed
data
DATA MINING
target
data
preprocessing
& cleaning
data
selection



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

```
if questions:
    try:
        answer()
    except RunTimeError:
        pass
    else:
        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

Pannang Curry 1-2-3!

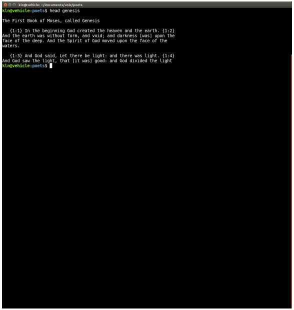
1. Open your right refrigerator door and remove ingredients from the following locations: Door shelf 2. Spot 1; Crisper drawer 1, Spot 3; Crisper drawer 1, Spot 5.
2. Open your third kitchen drawer and remove the utensils labeled “1”, “3”, “4”, “9”, and “12”.
3. Use your arms to apply utensil 1 to ingredients 1-3. Place ingredients inside utensil 3.

*Note: This recipe uses ShaKL the Shared Kitchen Layout. To use ShaKL, you’ll need to have installed ShaKL shelving, cabinetry, and utensils throughout your kitchen and pantry and have basic understanding of ShaKL managers.To learn more, read *Up and Running with ShaKL* (O’Billy Press, 2015). Want to improve ShaKL? Consider contributing to our team.

- 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



```
kln@vehicle:poets$ 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 [was] 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
kln@vehicle:poets$ 
```

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

```
$
```

`prompt` indicates that the `shell` is waiting for input

```
$ whoami
kln
$
```

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`

```
$ somecommand
somecommand: command not found
$
```

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

`print working directory` - current default directory

```
$ pwd
home/kln

$ a=$(pwd)
$ echo "current wd is: $a"
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

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

`sort` in alphabetic order

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

`uniq` - lexicon of document

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

tired of cryptic commands and operations from the command line?

luckily we have:

python

```
>>> import gensim, nltk, polyglot, spacy
>>> from adl.util import thefunctionthatrulesthemall
>>> thefunctionthatrulesthemall("yourfile.dat")
```

and:

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

```
if questions:
    try:
        answer()
    except RunTimeError:
        pass
    else:
        print "thank you"
```