Introduction to Text Mining

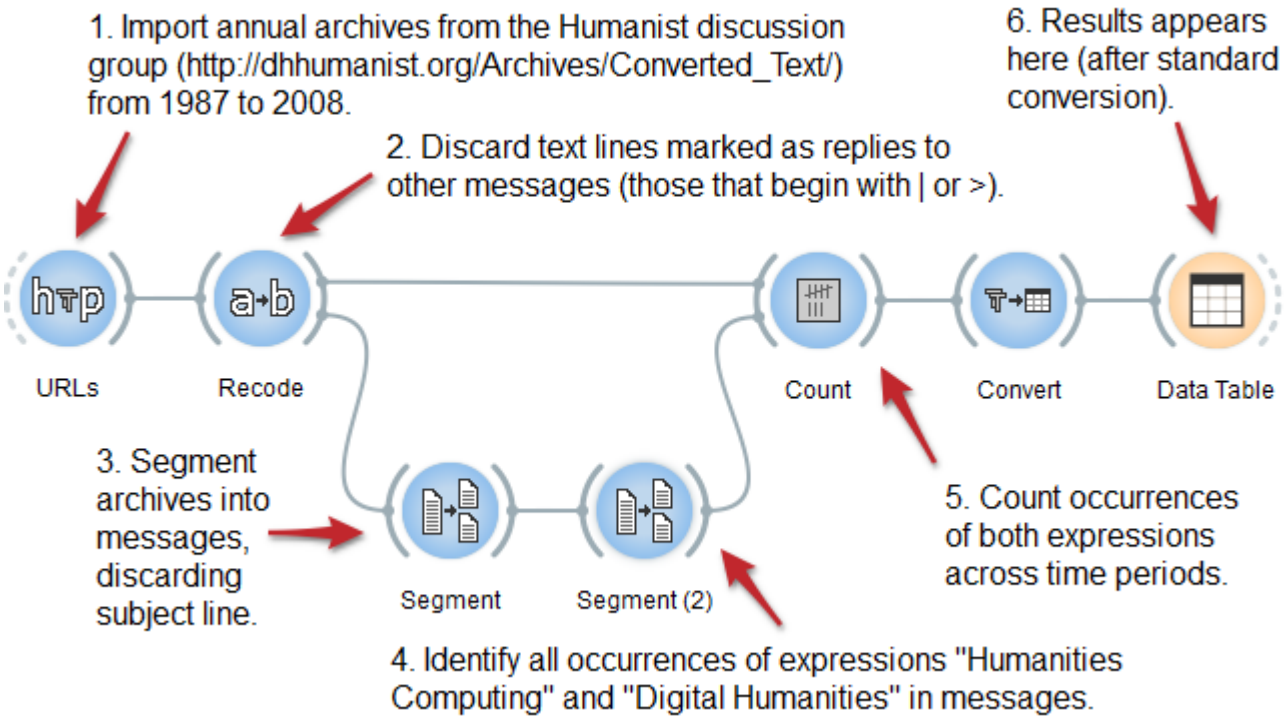
Aris Xanthos - University of Lausanne

Preliminary notes

- Presentation designed for a novice audience
- Text mining = text analysis = text analytics:
  - using computational and quantitative methods for extracting information from text data that can have anywhere between no structure and a fair amount of structure

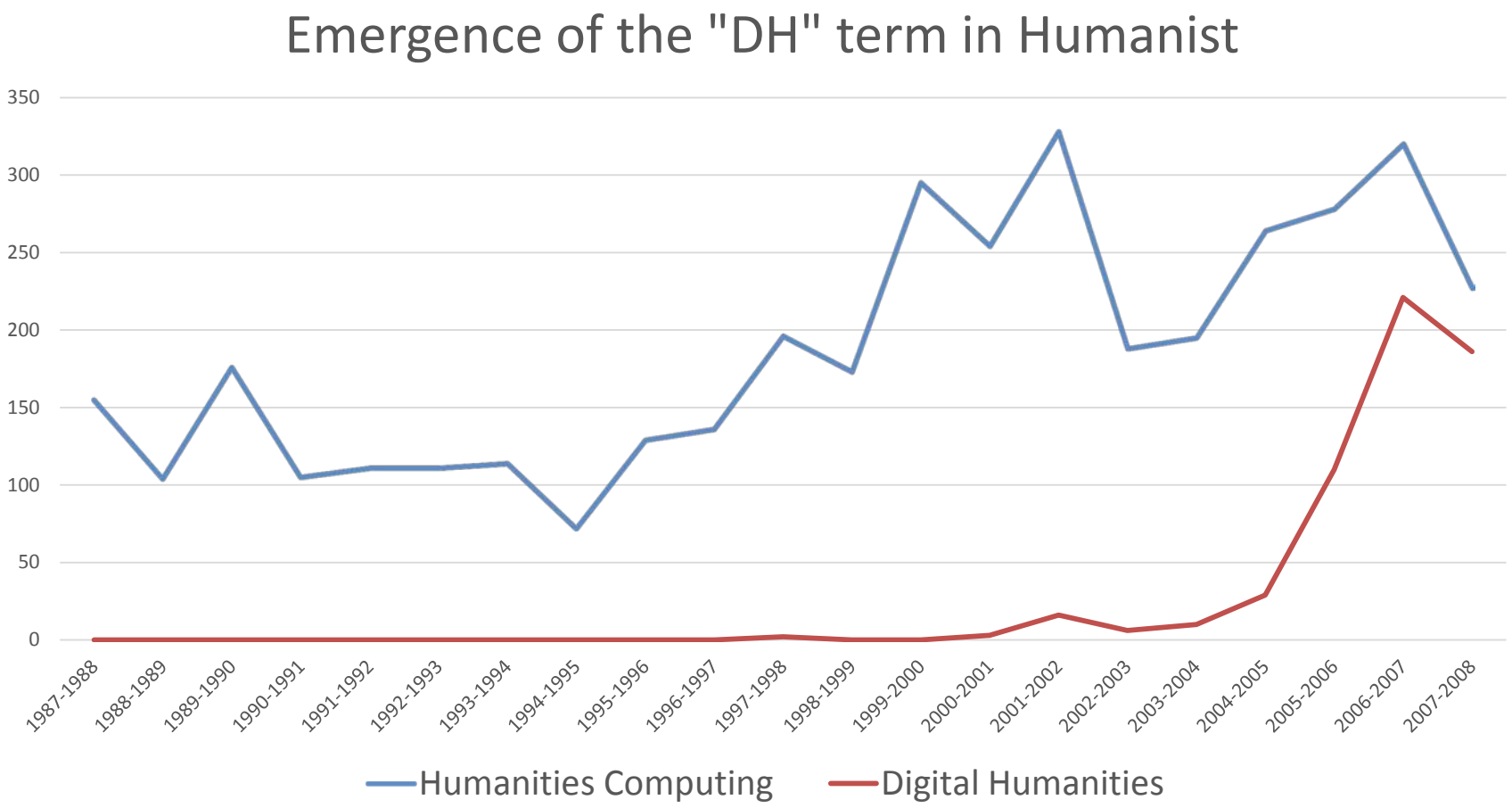
Whetting your appetite

- Mining Humanist (cf. Svensson 2009):



Source: Textable documentation

Whetting your appetite (2)



Overview

- Digital text data fundamentals
- Basic text analytics concepts
- Text annotation
- Sample text mining workflows

Introduction to Text Mining

DIGITAL TEXT DATA FUNDAMENTALS

Digital representation of data

- Anything's digital representation is a sequence of 0's and 1's (*bits*)
- Typically divided into *bytes* = sequences of 8 bits

#bits	#possible values
1	2 (*0* and *1*)
2	$2^2 = 4$ (*00*, *01*, *10* and *11*)
...	...
8 bits (1 byte)	$2^8 = 256$

Text encoding

- *Encoding* = convention about how to convert between binary and alphabetic text representation
- E.g.: *ASCII* code (*American Standard Code for Information Interchange*), developed in the 1960s
  - 128 characters (95 displayable)
  - unaccented letters, digits, symbols
  - coded with 7 bits ($128 = 2^7$) $\Rightarrow$ 1 spare bit per byte
  - 1000001 = *A*, 1000010 = *B*, ...
  - only English (and Latin) could be represented properly

Extended ASCII

- From the 1980s onwards, the 8th bit was used to define 128 extra characters: the *high* or *extended ASCII*, as opposed to the *low* or *standard ASCII*
- *Code page* = convention about how to convert between binary and alphabetic representation of high ASCII
- *ISO-8859* = set of 16 standard code pages, e.g. 11110001 corresponds to:
  - *ñ* in ISO-8859-1 (aka *Latin-1*)
  - *ρ* in ISO-8859-7
  - …
- But also *CP858*, *Windows-1252*, *Mac OS Roman*, ...

Unicode

- The development of the internet in the 1990s, issues due to conversion between code pages became more frequent.
- Solution = *Unicode*:
  - create an exhaustive directory of characters (more than 128'000 in 2017)
  - assign a unique numeric identifier (*code point*) to each character
  - several conventions for representing code points as binary strings: *UTF-16* (*big endian* ou *low endian*), *UTF-8*, ...

UTF-8

- Most widely used Unicode encoding
- Each code point is represented by at least 1 byte and at most 4 bytes
- The 128 characters of standard ASCII are represented by the same, single byte in UTF-8
- Texts encoded in standard ASCII can therefore be directly treated as if they were encoded in UTF-8

Best practices: encoding

- Know thy encoding!
- Fill in the metadata (in xml, html, ...)
- Write it down in file names (e.g. *macbeth_utf8.txt*)
- Use standard encodings (UTF-8, Latin-1, ...)

Best practices: files

- Use "raw text" formats (e.g. *.txt*, *.xml*, *.html*...)
  - avoid *.pdf*, *.docx*, *.rtf*, ...
- Name files carefully:
  - always use an extension (*.txt*, *.xml*, ...)
  - choose explicit names (~~*text.txt*~~ *joyce_ulysses.txt*)
  - use only lowercase unaccented letters (and digits)
  - use underscores (_), not spaces
  - use fixed-length numbers: *example01.txt*, *example02.txt*, ...
  - When using dates, use the *yyyy_mm_dd* format

Sources and acquisition

- Text in digital format (web pages, tweets, ebooks, ...)
  - preprocessing, recoding, normalization
- Printed text
  - scanning
  - optical character recognition (OCR)
  - correction
  - preprocessing, recoding, normalization
- Other sources: manuscripts, speech recordings, ...
  - keyboard transcription (but see Transkribus)

A few online repositories

- Project Gutenberg
- Internet Archive's eBooks and texts
- Oxford Text Archive
- Internet History Sourcebooks
- Perseus Digital Library
- …

Introduction to Text Mining

BASIC TEXT ANALYTICS CONCEPTS

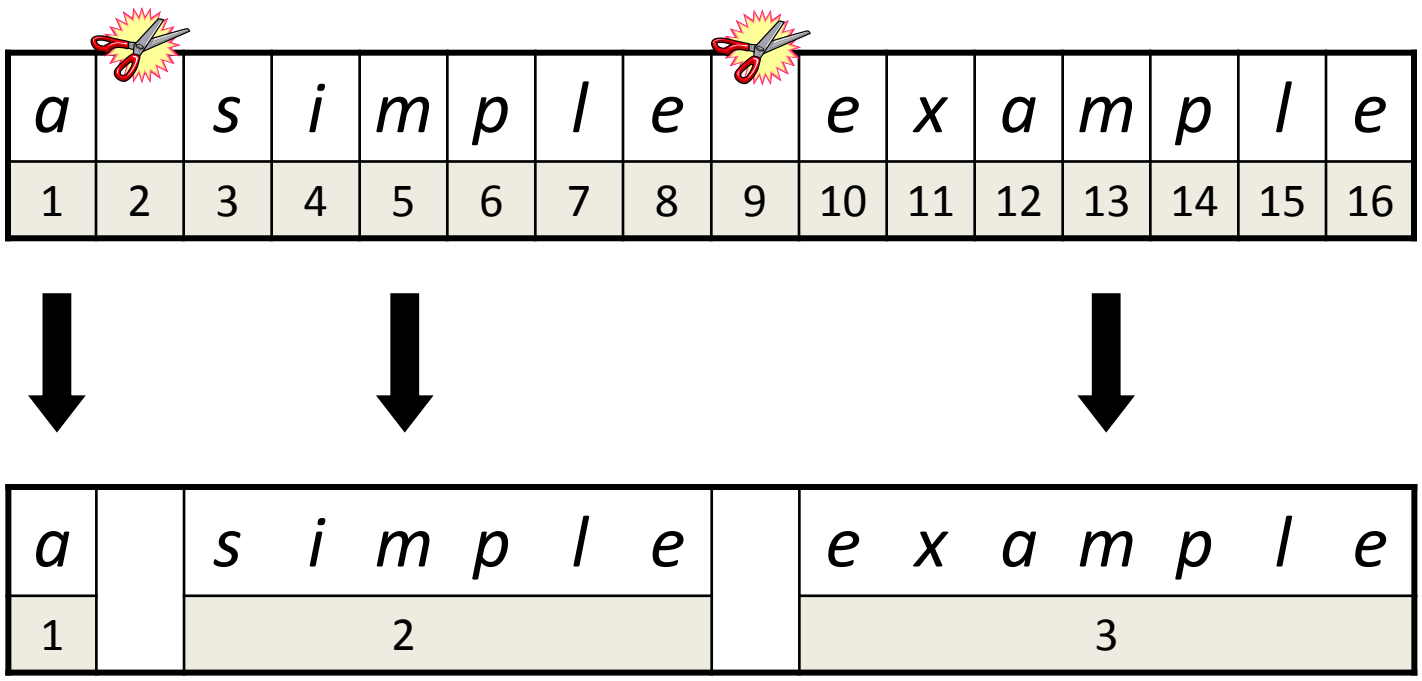
Types and tokens

a		*s*	*i*	*m*	*p*	*l*	*e*		*e*	*x*	*a*	*m*	*p*	*l*	*e*
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

- *Types* :
  - *a, s, i, m, p, l, e, x,* _ (space)
  - #types = *size* of alphabet (or lexicon), here 9
- *Tokens* (or *occurrences*) :
  - (1,*a*), (2,_), ..., (16,*e*)
  - #tokens = *length* of the text, here 16 (characters)

Segmentation

- Splitting a sequence to form new types and tokens:



Index and distribution

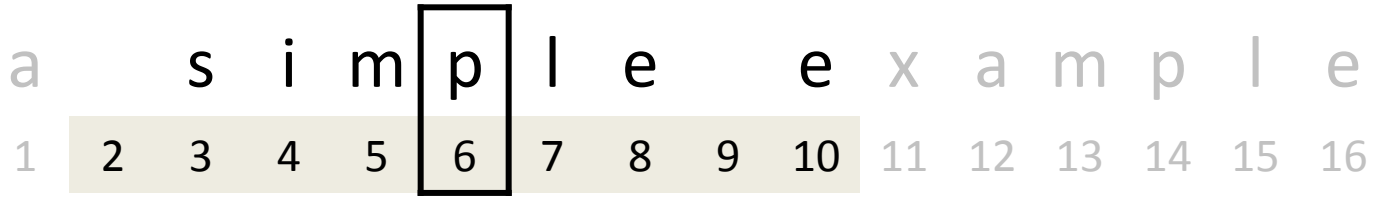
Type	Positions	Abs. freq	Rel. freq
a	1, 12	2	0.13
e	8, 10, 16	3	0.19
i	4	1	0.06
l	7, 15	2	0.13
m	5, 13	2	0.13
p	6, 14	2	0.13
s	3	1	0.06
x	11	1	0.06
_	2, 9	2	0.13
		16	1

A distribution in evening dress

- Most frequent words in inaugural speeches of US presidents (corpus compiled by Lucien Chapuisat):

Context of a token

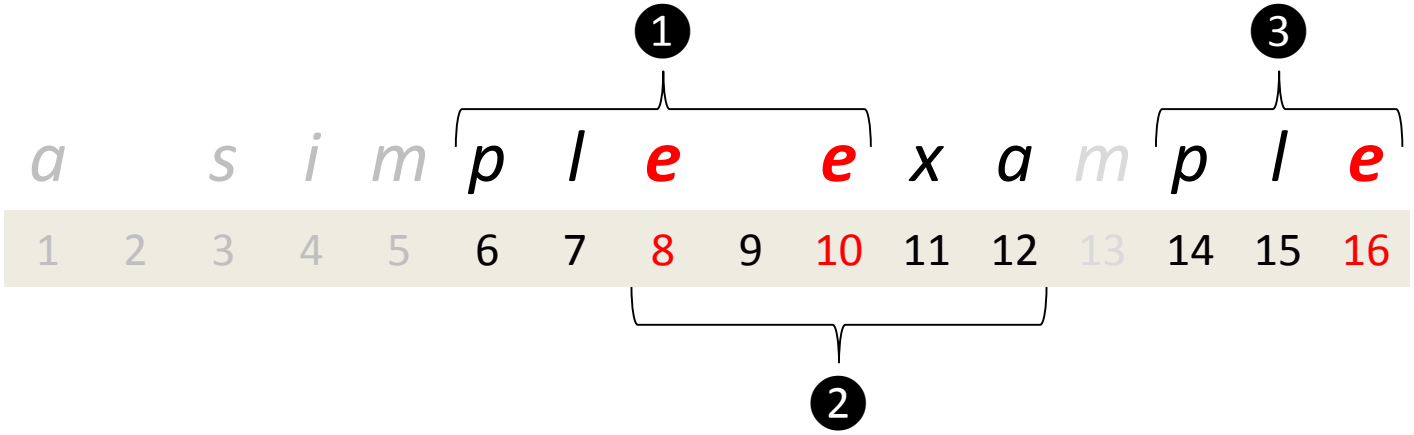
- *Window of n* adjacent tokens (e.g. *n* = 4):



- Higher-level unit (word, sentence, document...) :

Concordance

- List of contexts of occurrences of a given type:



1	p	l	**e**		e
2	e		**e**	x	a
3	p	l	**e**		

Concordance (2)

- Example: US inaugural speeches

Americans—and my fellow citizens of the world community: I ask you to share with
ws of international relationship, and establish a world court for the disposition of such justiciabl
of Republics reflects the glory of new-world democracy, but in the new order of
the settled opinions of mankind; for the world did not then believe that the supreme
a weapon our adversaries in today's world do not have. It is a weapon
amidst the conflicting elements of a troubled world. During the contest of opinion through whi
the throes and convulsions of the ancient world, during the agonizing spasms of infuriated m
. Second, we will continue our programs for world economic recovery. This means, first of al
is foreign and what is domestic; the world economy, the world environment, the world A

- Concordance plot (same query):

Collocations

- Set of types that appear in the context of a given type
- Sorted based on decreasing strength of association with the queried item:
  - *mutual information*
  - *T-score*
- Parameters:
  - minimal frequency
  - context size

Collocations (2)

- Example: inaugural speeches, "world", MI, freq. ≥ 10

	Context = 1L, 1R
1	old
2	s
3	new
4	the
5	is
6	a
7	we
...	

	Context = 20L, 20R
1	recovery
2	peoples
3	throughout
4	peace
5	international
6	mankind
7	arms
...	

Document-term matrix

- Distribution of units in different contexts
- E.g., if units = letters and contexts = words:

	a	***s***	***i***	***m***	***p***	***l***	***e***	***x***	
a	1	-	-	-	-	-	-	-	*1*
simple	-	1	1	1	1	1	1	-	*6*
example	1	-	-	1	1	1	2	1	*7*
	2	*1*	*1*	*2*	*2*	*2*	*3*	*1*	*14*

- Typically, units = (groups of) words and contexts = (groups of) documents

Normalization

- Divide each frequency by the total frequency (here row sum)
- Makes it possible to compare frequencies in contexts of different size:

	a	***s***	***i***	***m***	***p***	***l***	***e***	***x***	
a	**1**	-	-	-	-	-	-	-	*1*
simple	-	0.17	0.17	0.17	0.17	0.17	0.17	-	*1*
example	**0.14**	-	-	0.14	0.14	0.14	0.28	0.14	*1*

Specificity analysis

- Study the degree of attraction or repulsion between units and contexts
- E.g. *independence quotients* are > 1 in case of attraction and < 1 in case of repulsion:

	a	***s***	***i***	***m***	***p***	***l***	***e***	***x***	
a	7	-	-	-	-	-	-	-	*1*
simple	-	2.33	2.33	1.17	1.17	1.17	0.78	-	*1*
example	1	-	-	1	1	1	1.33	2	*1*

Specificity analysis (2)

- Word-level example: inaugural speeches, nouns with freq ≥ 5 most typical of Democratic party

__unit__	Democratic	Republican	Unaffiliated
discipline	2.479	0.000	0.000
agitation	2.479	0.000	0.000
vitality	2.479	0.000	0.000
globe	2.203	0.190	0.000
agreement	2.169	0.214	0.000
solicitude	2.125	0.244	0.000
amount	2.125	0.244	0.000
bounty	2.125	0.244	0.000
terror	2.125	0.244	0.000
compromise	2.097	0.263	0.000
posterity	2.066	0.285	0.000
compact	2.066	0.285	0.000
philosophy	2.066	0.285	0.000

Cooccurrences

- Number of contexts in which units occur together (e.g. letters in words):

	a	*s*	*i*	*m*	*p*	*l*	*e*	*x*
a	2	0	0	1	1	1	1	1
s	0	1	1	1	1	1	1	0
i	0	1	1	1	1	1	1	0
m	1	1	1	2	2	2	2	1
p	1	1	1	2	2	2	2	1
l	1	1	1	2	2	2	2	1
e	1	1	1	2	2	2	2	1
x	1	0	0	1	1	1	1	1

- Often represented as a weighted undirected network

Introduction to Text Mining

TEXT ANNOTATION

Annotation defined

«Corpus annotation is the practice of adding *interpretative* (especially linguistic) information to an existing corpus of spoken and/or written language, by some kind of coding attached to, or interspersed with, the electronic *representation* of the language material itself. »

Geoffrey Leech (1993). Corpus Annotation Schemes. *Literary and Linguistic Computing, 8(4*): 275-281

Example: POS-tagging

- Sample output of the Treetagger program:

```
When        WRB     when
it          PP      it
was         VBD     be
first       RB      first
perceived   VVN     perceive
,           ,       ,
in          IN      in
…
```

Example: emotion annotation

- Sample EmotionML markup, with two categorized and quantified emotion annotations:

```
<emotion>
  <category name="Disgust" value="0.82"/>
    ‘Come, there’s no use in crying like that!’
</emotion>
  said Alice to herself rather sharply;
<emotion>
  <category name="Anger" value="0.57"/>
    ‘I advise you to leave off this minute!’
</emotion>
```

Why and how to annotate

- Motivations:
  - facilitate data analysis
  - enable "recycling"
  - make interpretation explicit
- Methods
  - manual
  - semi-automatic
  - automatic

Annotation formats

- Ad hoc vs standard formats
- XML (eXtensible Markup Language) :
  - text data markup standard
  - can be adapted to the needs of different users
  - well integrated (in browsers, programming languages, ...)
  - covers a number of technologies (XSLT, XPath, ...)

XML: basic principles

- Tags:

```
<text>a simple example</text>
```

- Strictly hierarchical structure:

```
<text>
  <word>a</word>
  <word>simple</word>
  <word>example</word>
</text>
```

- Attribute notation *(key="value"):*

```
<word lemma="example" type="NN">example</word>
```

XML : complete document

- Begin with XML declaration
- Use 1 and only 1 root element (here **<text>**)

```
<?xml version="1.0" encoding="utf-8"?>
<text>
  <word type="DT">a</word>
  <word type="JJ">simple</word>
  <word type="NN">example</word>
  followed by an unannotated string...
</text>
```

XML : well-formed vs. valid

- *Well-formedness* of an XML document
  - conformance to basic principles
- *Validity* of a well-formed XML document
  - relative to an external reference (DTD, schema, ...)
  - proper usage of predefined tags and attributes
  - E.g.: Text Encoding Initiative

Introduction to Text Mining

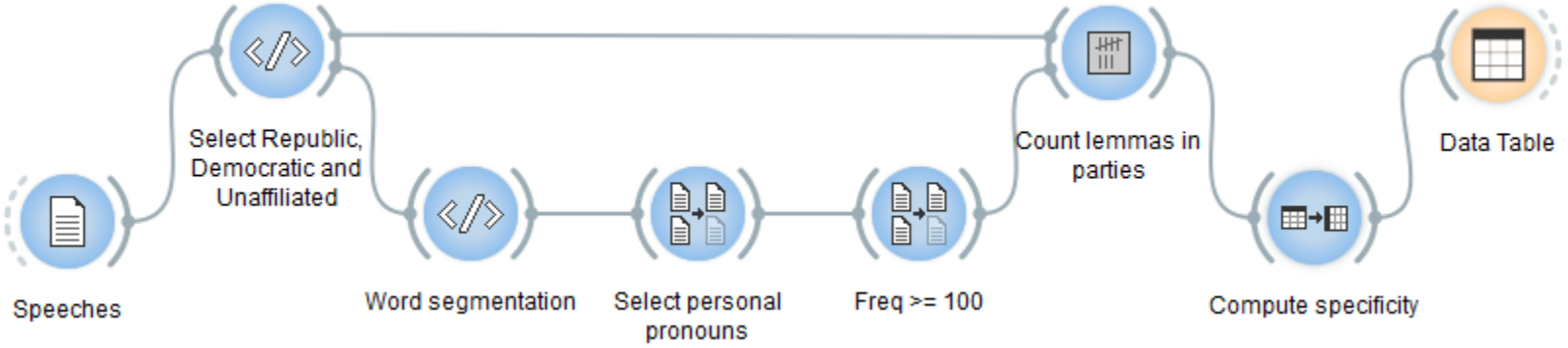
SAMPLE TEXT MINING WORKFLOWS

Data

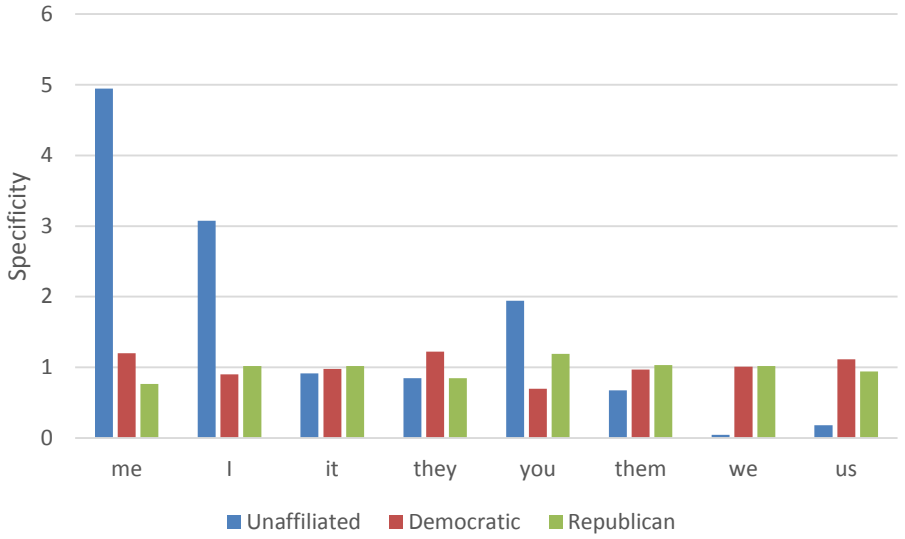
- 59 US inaugural speeches, ~149'000 mots
- 25 Republican, 22 democratic, 12 other
- POS-tagged, lemmatized, converted to XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<corpus>
<sp date='1789' name='G.Washington' party='Unaffiliated'
num='first'>
<w lemma="[unknown]" type="NNS">Fellow-Citizens</w>
<w lemma="of" type="IN">of</w>
<w lemma="the" type="DT">the</w>
<w lemma="Senate" type="NP">Senate</w>
…
```

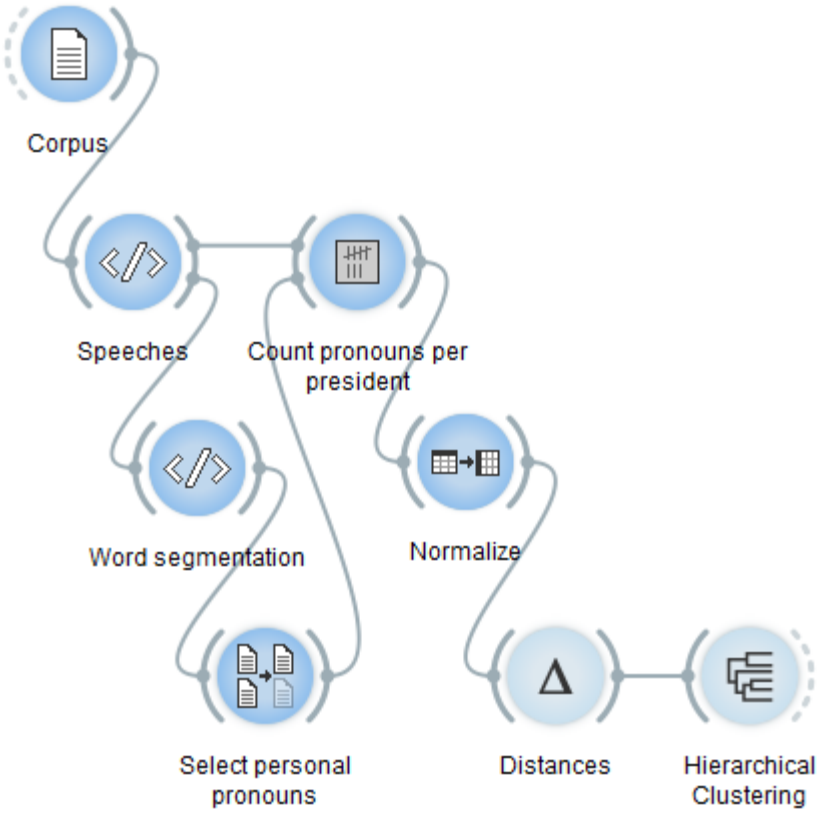
Personal pronoun specificity

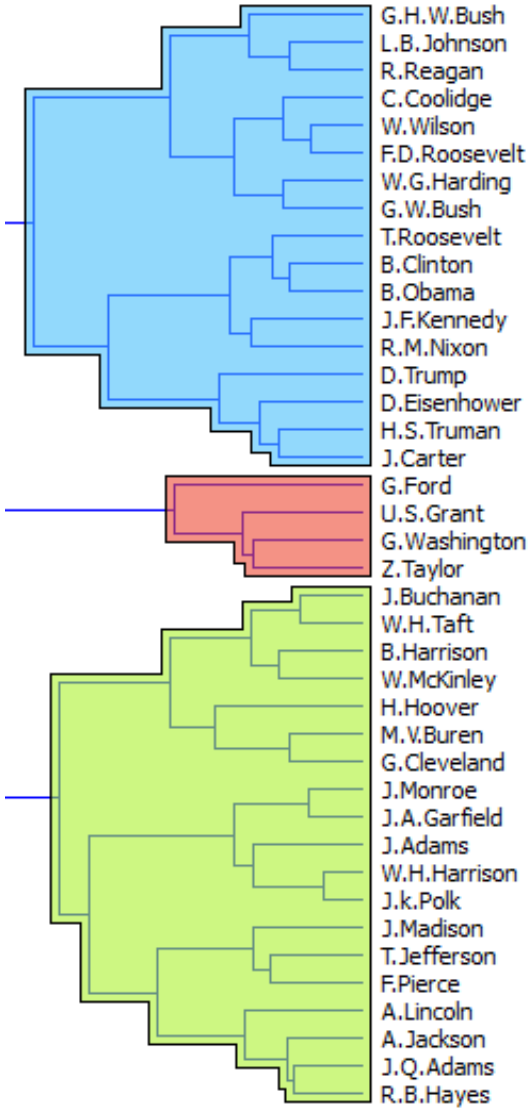


unit	Unaffiliated	Democratic	Republican
me	4.944	1.199	0.765
I	3.074	0.900	1.021
you	1.945	0.697	1.191
it	0.913	0.977	1.018
they	0.847	1.222	0.847
them	0.676	0.968	1.030
us	0.179	1.114	0.939
we	0.045	1.009	1.016

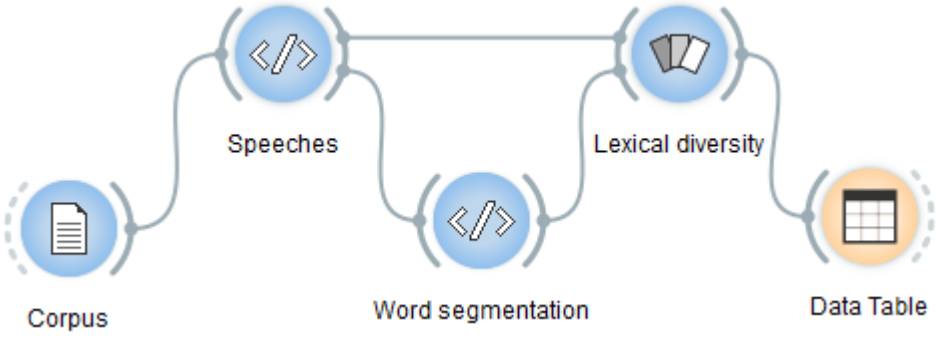


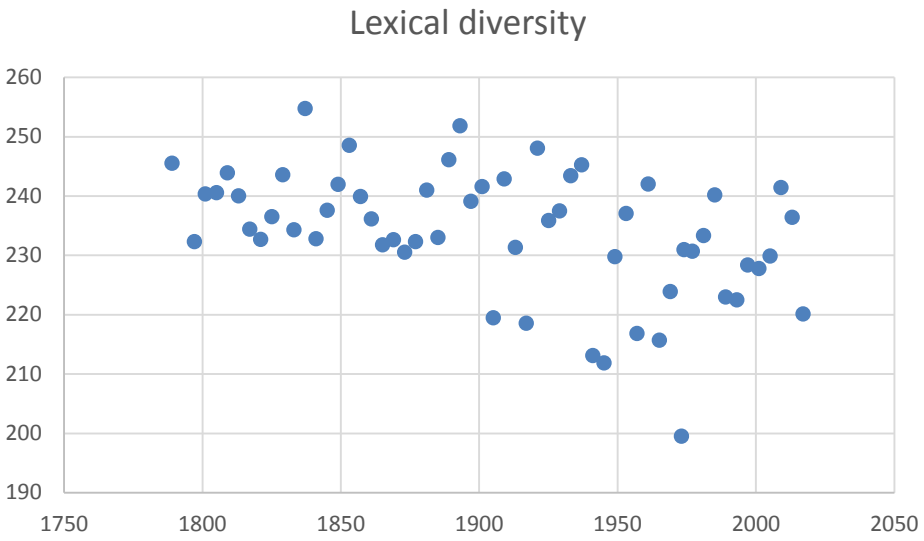
Clustering based on pronoun use





Evolution of lexical diversity





To be continued...

- Workshop today from 16 to 17:30 for learning how to do such things by yourselves using visual programming (including a crash course on regular expressions)
- Make sure to install the required open source software beforehand:
  1. Orange 3.4
  2. Textable