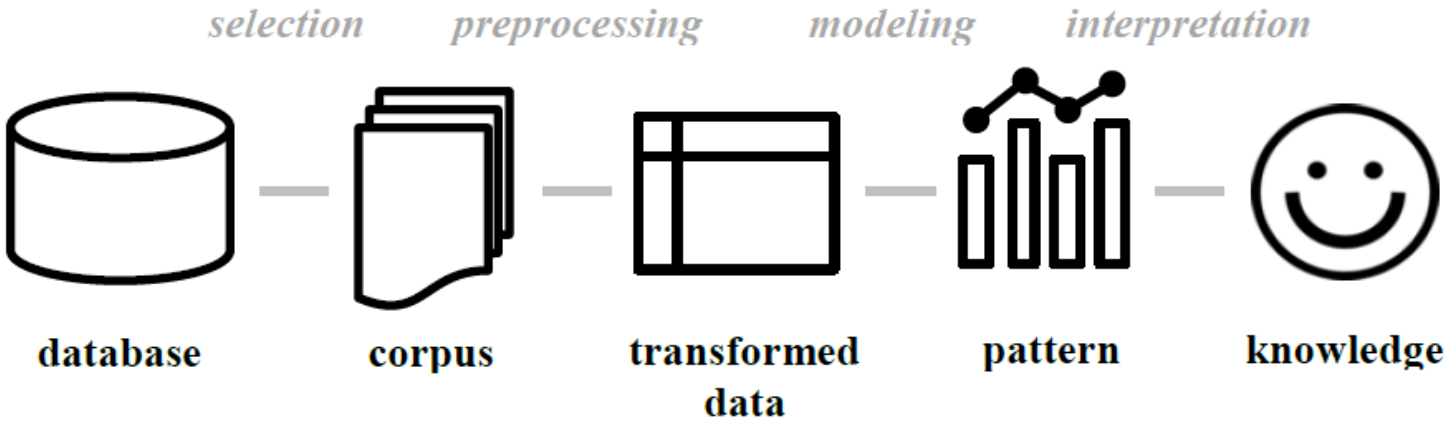


general multistep process of knowledge discovery

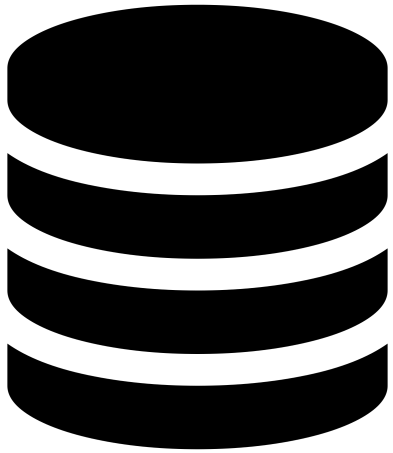
constructed out of a need for handling ‘extraction of useful information (knowledge) from rapidly growing volumes of digital data’

for each project you develop a pipeline within this workflow

accessing data select specific documents (target data/corpus) relevant to your research question from the database.

online databases and research libraries are excellent resources

- beware of legal issues, always ask your provider/supervisor
- historical sources can be a problem



we will focus on documents stored locally in a plain text format (Python and R are very accommodating though).

```
1 library(rvest)
2 walkingdead.l <- html("http://www.imdb.com/title/tt1520211/")
3 cast.v <- html_text(html_nodes(walkingdead.l,
4             "#titleCast .itemprop span"))
```

to prepare a document ('a string of characters') we need to split or *tokenize* it at the relevant level(s).

unstructured data are very noisy, to increase the signal we therefore remove irrelevant data through **preprocessing**.

use a range of text normalization techniques to preprocess the data:

- casefolding
- removal of non-alphanumeric characters (punctuation, blanks)
- removal of numeral and stopwords
- reduction of inflectional forms through stemming and lemmatization
- synonym substitution

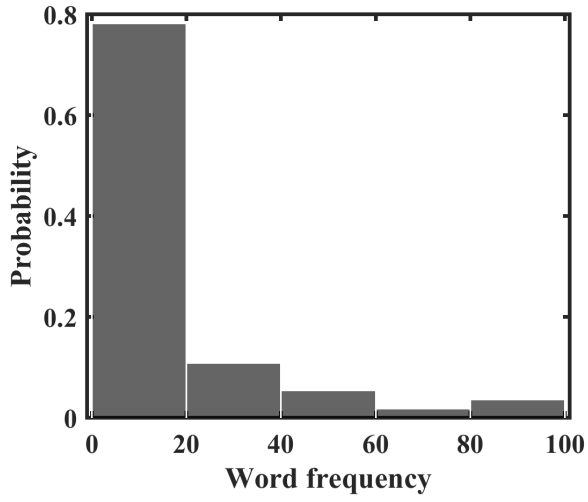
remember that one man's rubbish may be another's treasure

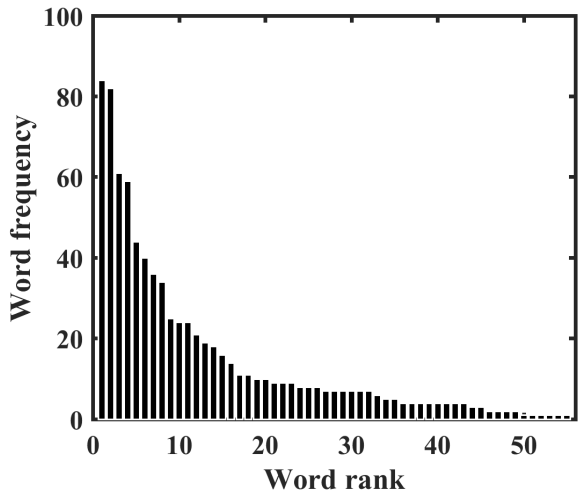
words are (one of) the basic units of meaning

most text analytics techniques rely on word frequencies, that is, we tokenize a text at the word level and count the number of tokens for each type.

I am Daniel	'I' 'am' 'Daniel' 'I'
I am Sam	'am' 'Sam' 'Sam'
Sam I am	'I' 'am' 'That'
That Sam-I-am	'Sam' 'I' 'am'
That Sam-I-am!	'That' 'Sam' 'I'
I do not like	'am' 'I' 'do' 'not'
that Sam-I-am	'like' 'that' 'Sam'
...	'I' 'am' ...

a	1	59	0.073
am	1	16	0.02
and	1	24	0.03
anyhwhere	1	1	0.001
anywhere	1	7	0.009
...			
you	1	34	0.042
total	55	804	1.0





most words are infrequent, but a few words are very frequent

'i' 'not' 'them' 'a' 'like' 'in' 'do' 'you' 'would'

model a text as a distribution over words. Some words are more likely than other.

often times we are looking at the mid-range (not too likely and not too unlikely).

binary term frequency: $f_{t,d} = 0, 1$

raw term frequency: $f_{t,d} = N(t, D)$

normalized term frequency [1] $f_{t,d} = \frac{N(t, D)}{N(D)}$

IDF weighted term frequency [2]: $tfidf(t, d, D) = f_{t,d} \cdot log \frac{\mid D \mid}{\{d \in D : t \in d\}}$

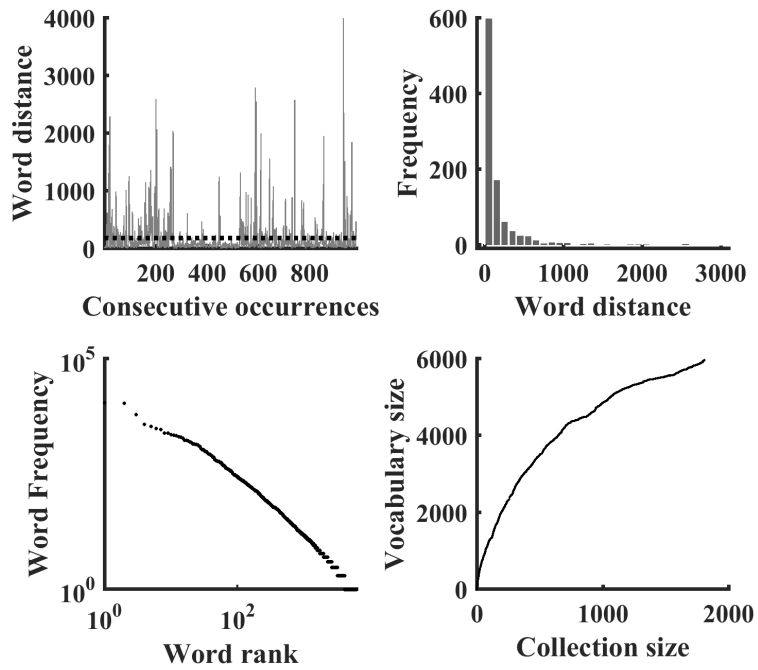
[1] prevents bias towards longer documents

[2] removes non-informative words

words occur in bursts - if word occurs it is likely to occur again in close proximity

a word's frequency is inversely proportional to its rank

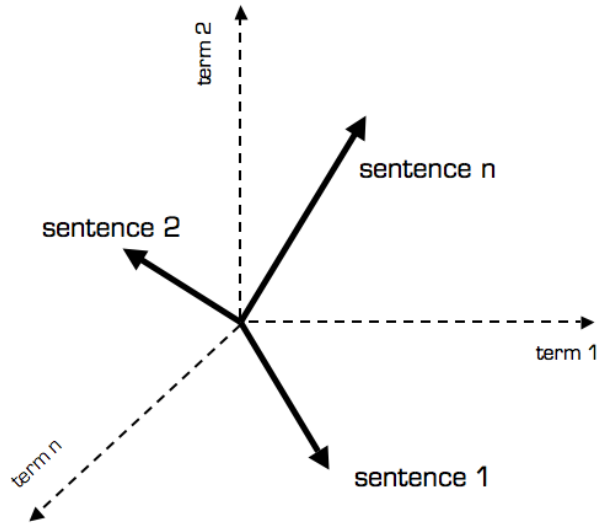
the vocabulary increases as a function of the number of texts, but the increase diminishes as more texts are included



any corpus (i.e., a collection of *n* documents) can be represented in the vector space model by a document-term matrix

a vector space model is a basic modeling mechanism for a word- or document-space (whether we look at rows or columns)

- a document vector with only one word is collinear to the vocabulary word axis
- a document vector that does not contain a specific word is orthogonal/perpendicular to the word axis
- two documents are identical if they contain the same words in a different order



Document space	t_1	t_2	t_3	...	t_n ← Term vector space
D_1	a_{11}	a_{12}	a_{13}	...	a_{1n}
D_2	a_{21}	a_{22}	a_{23}	...	a_{2n}
D_3	a_{31}	a_{32}	a_{33}	...	a_{3n}
...					
D_m	a_{m1}	a_{m2}	a_{m3}	...	a_{mn}
Q	b_1	b_2	b_3	...	b_n