Describing and comparing texts

Kenneth Benoit

Quants 3: Quantitative Text Analysis

Week 2: February 23, 2018

Day 2 Outline

- Problems to watch out for
- Getting to know your texts
- Key words in context
- Revisiting feature selection
- Feature weighting strategies
- Collocations
- Named entity recognition
- Readability and lexical diversity
- Assignment 2

Problems you are likely to encounter

- Problems with encoding
- Problems file formats
- Extraneous junk (page footers, numbers, titles, etc)
- misspelllings
- different normalizations (e.g. for Japanese)

Simple descriptive table about texts: Describe your data!

Speaker	Party	Tokens	Types
Brian Cowen	FF	5,842	1,466
Brian Lenihan	FF	7,737	1,644
Ciaran Cuffe	Green	1,141	421
John Gormley (Edited)	Green	919	361
John Gormley (Full)	Green	2,998	868
Eamon Ryan	Green	1,513	481
Richard Bruton	FG	4,043	947
Enda Kenny	FG	3,863	1,055
Kieran ODonnell	FG	2,054	609
Joan Burton	LAB	5,728	1,471
Eamon Gilmore	LAB	3,780	1,082
Michael Higgins	LAB	1,139	437
Ruairi Quinn	LAB	1,182	413
Arthur Morgan	SF	6,448	1,452
Caoimhghin O'Caolain	SF	3,629	1,035
All Texts		49,019	4,840
Min		919	361
Max		7,737	1,644
Median		3,704	991
Hapaxes with Gormley Edited		67	
Hapaxes with Gormley Full Speech		69	

Exploring Texts: Key Words in Context

KWIC *Key words in context* Refers to the most common format for concordance lines. A KWIC index is formed by sorting and aligning the words within an article title to allow each word (except the stop words) in titles to be searchable alphabetically in the index.

lime (14)

79[C.10]	4	/Which was builded of **lime** and sand;/Until they came to
247A.6	4	/That was well biggit with **lime** and stane.
303A.1	2	bower,/Well built wi **lime** and stane,/And Willie came
247A.9	2	/That was well biggit wi **lime** and stane,/Nor has he stoln
305A.2	1	a castell biggit with **lime** and stane,/O gin it stands not
305A.71	2	is my awin,/I biggit it wi **lime** and stane;/The Tinnies and
79[C.10]	6	/Which was builded with **lime** and stone.
305A.30	1	a prittie castell of **lime** and stone,/O gif it stands not
108.15	2	/W*hi*ch was made both of **lime** and stone,/Shee tooke him by
175A.33	2	castle then,/Was made of **lime** and stone;/The vttermost
178[H.2]	2	near by,/Well built with **lime** and stone;/There is a lady
178F.18	2	built with stone and **lime**!/But far mair pittie on Lady
178G.35	2	was biggit wi stane and **lime**!/But far mair pity o Lady
2D.16	1	big a cart o stane and **lime**,/Gar Robin Redbreast trail it

Another KWIC Example (Seale et al (2006)

Table 3
Example of Keyword in Context (KWIC) and associated word clusters display

Extracts from Keyword in Context (KWIC) list for the word 'scan'

An MRI **scan** then indicated it had spread slightly

Fortunately, the MRI **scan** didn't show any involvement of the lymph nodes

3 very worrying weeks later, a bone **scan** also showed up clear.

The bone **scan** is to check whether or not the cancer has spread to the bones.

The bone **scan** is done using a type of X-ray machine.

The results were terrific, CT **scan** and pelvic X-ray looked good

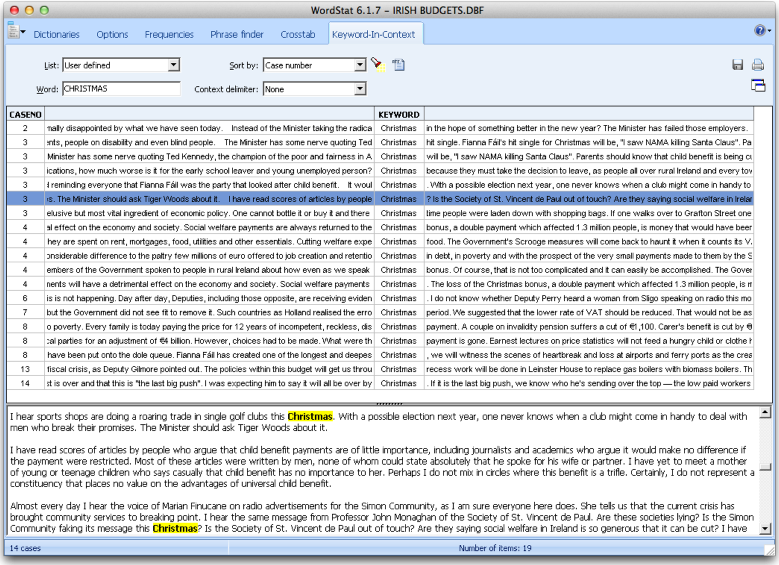
Your next step appears to be to await the result of the **scan** and I wish you well there.

I should go and have an MRI **scan** and a bone **scan**

Three-word clusters most frequently associated with keyword 'scan'

N	Cluster	Freq
1	A bone scan	28
2	Bone scan and	25
3	An MRI scan	18
4	My bone scan	15
5	The MRI scan	15
6	The bone scan	14
7	MRI scan and	12
8	And Mri scan	9
9	Scan and MRI	9

Another KWIC Example: Irish Budget Speeches



Irish Budget Speeches KIWC in quanteda

R Console

```
> data(iebudgets)
> iebudgets2010 <- subset(iebudgets, year==2010)
> kwic(iebudgets2010, "christmas", regex=TRUE)
                                                             preword       word                             postword
     [2010_BUDGET_02_Richard_Bruton_FG.txt, 628]        and to see out this  Christmas in the hope of something
       [2010_BUDGET_03_Joan_Burton_LAB.txt, 371]       to suggest titles for a Christmas  hit single. Fianna Fáil’s hit
       [2010_BUDGET_03_Joan_Burton_LAB.txt, 379]      Fianna Fáil’s hit single for Christmas  will be, “I saw NAMA
       [2010_BUDGET_03_Joan_Burton_LAB.txt, 922]      women will say goodbye after Christmas  because they must take the
      [2010_BUDGET_03_Joan_Burton_LAB.txt, 1518]       in single golf clubs this Christmas. With a possible election next
      [2010_BUDGET_03_Joan_Burton_LAB.txt, 1726] Community faking its message this Christmas? Is the Society of St.
      [2010_BUDGET_03_Joan_Burton_LAB.txt, 3159]       bags. In previous years at Christmas  time people were laden down
      [2010_BUDGET_04_Arthur_Morgan_SF.txt, 346]          €204 per week or the Christmas  bonus. Of course, that is
     [2010_BUDGET_04_Arthur_Morgan_SF.txt, 3239]   to social welfare payments this Christmas. The loss of the Christmas
     [2010_BUDGET_04_Arthur_Morgan_SF.txt, 3244]       Christmas. The loss of the Christmas  bonus, a double payment which
     [2010_BUDGET_04_Arthur_Morgan_SF.txt, 3272]     streets on Santa presents and Christmas  food. The Government’s Scrooge measures
     [2010_BUDGET_04_Arthur_Morgan_SF.txt, 5899]        their jobs, who face this Christmas  in debt, in poverty and
        [2010_BUDGET_06_Enda_Kenny_FG.txt, 2629] to implement the reduction before Christmas. I do not know whether
   [2010_BUDGET_07_Kieran_ODonnell_FG.txt, 1365]          from the change in the  Christmas period. We suggested that the
     [2010_BUDGET_08_Eamon_Gilmore_LAB.txt, 550]       cut of €641, including the Christmas  payment. A couple on invalidity
     [2010_BUDGET_08_Eamon_Gilmore_LAB.txt, 638]       are on social welfare, the Christmas  payment is gone. Earnest lectures
     [2010_BUDGET_08_Eamon_Gilmore_LAB.txt, 998]    of emigration. Once again this Christmas, we will witness the scenes
    [2010_BUDGET_13_Ciaran_Cuffe_Green.txt, 911]      noted recently that over the  Christmas recess work will be done
[2010_BUDGET_14_Caoimhghin_OCaolain_SF.txt, 148]            will all be over by Christmas. If it is the last
>
```

Defining Features

- words
- word stems or lemmas: this is a form of defining *equivalence classes* for word features
- word segments, especially for languages using compound words, such as German, e.g.
  Rindfleischetikettierungsberwachungsaufgabenbertragungsgesetz
  (the law concerning the delegation of duties for the supervision of cattle marking and the labelling of beef)
  Saunauntensitzer

Defining Features (cont.)

- "word" sequences, especially when inter-word delimiters (usually white space) are not commonly used, as in Chinese
  莎拉波娃现在居住在美国东南部的佛罗里达。今年 4 月
  9 日，莎拉波娃在美国第一大城市纽约度过了 1 8 岁生
  日。生日派对上，莎拉波娃露出了甜美的微笑。
- linguistic features, such as parts of speech
- (if qualitative coding is used) coded or annotated text segments
- linguistic features: parts of speech

Stemming words

Lemmatization refers to the algorithmic process of converting words to their lemma forms.

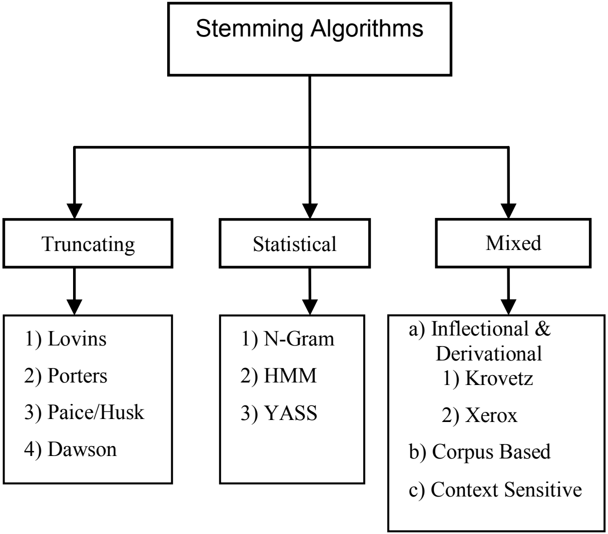
stemming the process for reducing inflected (or sometimes derived) words to their stem, base or root form. Different from *lemmatization* in that stemmers operate on single words without knowledge of the context.

both convert the morphological variants into stem or root terms

example: `produc` from `production`, `producer`, `produce`, `produces`, `produced`

Why? Reduce feature space by collapsing different words into a stem (e.g. “happier” and “happily” convey same meaning as “happy”)

Varieties of stemming algorithms



Issues with stemming approaches

- The most common is probably the Porter stemmer
- But this set of rules gets many stems wrong, e.g.
  - `policy` and `police` considered (wrongly) equivalent
  - `general` becomes `gener`, `iteration` becomes `iter`
- Other corpus-based, statistical, and mixed approaches designed to overcome these limitations
- Key for you is to be careful through inspection of morphological variants and their stemmed versions
- Sometimes not appropriate! e.g. Schofield and Minmo (2016) find that "stemmers produce no meaningful improvement in likelihood and coherence (of topic models) and in fact can degrade topic stability"

Parts of speech

- the Penn "Treebank" is the standard scheme for tagging POS

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Parts of speech (cont.)

```
> library("spacyr")
> txt <- "Pierre Vinken, 61 years old, will join the board as a nonexecutive
          director Nov. 29.  Mr. Vinken is chairman of Elsevier N.V.,
          the Dutch publishing group."
> spacy_parse(txt)
doc_id sentence_id token_id        token       lemma    pos       entity
1   text1          1        1       Pierre      pierre PROPN     PERSON_B
2   text1          1        2       Vinken      vinken PROPN     PERSON_I
3   text1          1        3            ,           , PUNCT
4   text1          1        4           61          61   NUM       DATE_B
5   text1          1        5        years        year  NOUN       DATE_I
6   text1          1        6          old         old   ADJ       DATE_I
7   text1          1        7            ,           , PUNCT
8   text1          1        8         will        will  VERB
9   text1          1        9         join        join  VERB
10  text1          1       10          the         the   DET
11  text1          1       11        board       board  NOUN
12  text1          1       12           as          as   ADP
13  text1          1       13            a           a   DET
14  text1          1       14 nonexecutive nonexecutive   ADJ
15  text1          1       15   \n            \n        SPACE
16  text1          1       16     director    director  NOUN
17  text1          1       17         Nov.        nov. PROPN       DATE_B
18  text1          1       18           29          29   NUM       DATE_I
19  text1          1       19            .           . PUNCT
```

Parts of speech (cont.)

20	text1	1	20			SPACE	
21	text1	2	1	Mr.	mr.	PROPN	
22	text1	2	2	Vinken	vinken	PROPN	PERSON_B
23	text1	2	3	is	be	VERB	
24	text1	2	4	chairman	chairman	NOUN	
25	text1	2	5	of	of	ADP	
26	text1	2	6	Elsevier	elsevier	PROPN	ORG_B
27	text1	2	7	N.V.	n.v.	PROPN	ORG_I
28	text1	2	8	,	,	PUNCT	
29	text1	2	9	\n	\n	SPACE	WORK_OF_ART_B
30	text1	2	10	the	the	DET	WORK_OF_ART_I
31	text1	2	11	Dutch	dutch	ADJ	NORP_B
32	text1	2	12	publishing	publishing	NOUN	
33	text1	2	13	group	group	NOUN	
34	text1	2	14	.	.	PUNCT	

Stemming v. lemmas

```
> library("quanteda")
> tokens(txt) %>% tokens_wordstem()
tokens from 1 document.
text1 :
[1] "Pierr"     "Vinken"    ","         "61"        "year"      "old"        ","
[9] "join"      "the"       "board"     "as"        "a"         "nonexecut" "di
[17] "."         "29"        "."         "Mr"        "."         "Vinken"    "i
[25] "of"        "Elsevier"  "N.V"       "."         ","         "the"       "D
[33] "group"     "."

sp$lemma
[1] "pierre"       "vinken"       ","            "61"           "year"
[7] ","            "will"         "join"         "the"          "board"
[13] "a"            "nonexecutive" "\n           "  "director"     "nov."
[19] "."            " "            "mr."          "vinken"       "be"
[25] "of"           "elsevier"     "n.v."         ","            "\n           "
[31] "dutch"        "publishing"   "group"        "."
```

Weighting strategies for feature counting

term frequency
: Some approaches trim very low-frequency words. Rationale: get rid of rare words that expand the feature matrix but matter little to substantive analysis

document frequency
: Could eliminate words appearing in few documents

inverse document frequency
: Conversely, could weight words more that appear in the most documents

tf-idf
: a combination of term frequency and inverse document frequency, common method for feature weighting

Strategies for feature *weighting*: tf-idf

- $tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$ where $n_{i,j}$ is number of occurences of term t_i in document d_j, k is total number of terms in document d_j
- $idf_i = \ln \frac{|D|}{|\{d_j : t_i \in d_j\}|}$
  where
  - $|D|$ is the total number of documents in the set
  - $|\{d_j : t_i \in d_j\}|$ is the number of documents where the term t_i appears (i.e. $n_{i,j} \neq 0$)
- $tf\text{-}idf_i = tf_{i,j} \cdot idf_i$

Computation of tf-idf: Example

Example: We have 100 political party manifestos, each with 1000 words. The first document contains 16 instances of the word "environment"; 40 of the manifestos contain the word "environment".

- The *term frequency* is $16/1000 = 0.016$
- The *document frequency* is $100/40 = 2.5$, or $\ln(2.5) = 0.916$
- The *tf-idf* will then be $0.016 * 0.916 = 0.0147$
- If the word had only appeared in 15 of the 100 manifestos, then the *tf-idf* would be 0.0304 (three times higher).
- A high weight in tf-idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; hence the weights hence tend to filter out common terms

Other weighting schemes

- the SMART weighting scheme (Salton 1991, Salton et al): The first letter in each triplet specifies the term frequency component of the weighting, the second the document frequency component, and the third the form of normalization used (not shown). Example: *lnn* means log-weighted term frequency, no idf, no normalization

Term frequency		Document frequency	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$
a (augmented)	$0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N - \mathrm{df}_t}{\mathrm{df}_t}\}$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$		
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$		

- Note: Mostly used in information retrieval, although some use in machine learning

Selecting more than words: collocations

collocations bigrams, or trigrams e.g. *capital gains tax*

how to detect: pairs occuring more than by chance, by measures of χ^2 or *mutual information* measures

example:

Summary Judgment	Silver Rudolph	Sheila Foster
prima facie	COLLECTED WORKS	Strict Scrutiny
Jim Crow	waiting lists	Trail Transp
stare decisis	Academic Freedom	Van Alstyne
Church Missouri	General Bldg	Writings Fehrenbacher
Gerhard Casper	Goodwin Liu	boot camp
Juan Williams	Kurland Gerhard	dated April
LANDMARK BRIEFS	Lee Appearance	extracurricular activities
Lutheran Church	Missouri Synod	financial aid
Narrowly Tailored	Planned Parenthood	scored sections

Table 5: Bigrams detected using the mutual information measure.

Identifying collocations

- Does a given word occur next to another given word with a higher relative frequency than other words?
- If so, then it is a candidate for a collocation or "word bigram"
- We can detect these using χ^2 or likelihood ratio measures (Dunning paper)
- Implemented in quanteda as `collocations()`

Getting texts into `quanteda`

- text format issue
  - text files
  - zipped text files
  - spreadsheets/CSV
  - (pdfs)
  - (Twitter feed)
- encoding issue
- metadata and document variable management

Identifying collocations

- Does a given word occur next to another given word with a higher relative frequency than other words?
- If so, then it is a candidate for a collocation
- We can detect these using measures of association, such as a likelihood ratio, to detect word pairs that occur with greater than chance frequency, compared to an independence model
- The key is to distinguish "true collocations" from uninteresting word pairs/triplets/etc, such as "of the"
- Implemented in quanteda as `collocations`

Example

$C(w^1\ w^2)$	w^1	w^2
80871	of	the
58841	in	the
26430	to	the
21842	on	the
21839	for	the
18568	and	the
16121	that	the
15630	at	the
15494	to	be
13899	in	a
13689	of	a
13361	by	the
13183	with	the
12622	from	the
11428	New	York
10007	he	said
9775	as	a
9231	is	a
8753	has	been
8573	for	a

Table 5.1 Finding Collocations: Raw Frequency. $C(\cdot)$ is the frequency of something in the corpus.

(from Manning and Schütze, *FSNLP*, Ch 5)

Example

$C(w^1\ w^2)$	w^1	w^2
80871	of	the
58841	in	the
26430	to	the
21842	on	the
21839	for	the
18568	and	the
16121	that	the
15630	at	the
15494	to	be
13899	in	a
13689	of	a
13361	by	the
13183	with	the
12622	from	the
11428	New	York
10007	he	said
9775	as	a
9231	is	a
8753	has	been
8573	for	a

Table 5.1 Finding Collocations: Raw Frequency. $C(\cdot)$ is the frequency of something in the corpus.

(from Manning and Schütze, *FSNLP*, Ch 5)

Contingency tables for bigrams

Tabulate every token against every other token as pairs, and compute for each token:

	token2	$\neg$token2	Totals
token1	n_{11}	n_{12}	n_{1p}
$\neg$token1	n_{21}	n_{22}	n_{1p}
Totals	n_{p1}	n_{p2}	n_{pp}

Contingency tables for trigrams

		token3	¬token3	Totals
token1	token2	n_{111}	n_{112}	n_{11p}
token1	¬token2	n_{121}	n_{122}	n_{12p}
¬token1	token2	n_{211}	n_{212}	n_{21p}
¬token1	¬token2	n_{221}	n_{222}	n_{22p}
	Totals	n_{pp1}	n_{pp2}	n_{ppp}

computing the “independence” model

- bigrams

$$Pr(\text{token1}, \text{token2}) = Pr(\text{token1})Pr(\text{token2})$$

- trigrams

$$Pr(\text{t1}, \text{t2}, \text{t3}) = Pr(\text{t1})Pr(\text{t2})Pr(\text{t3})$$
$$Pr(\text{t1}, \text{t2}, \text{t3}) = Pr(\text{t1}, \text{t2})Pr(\text{t3})$$
$$Pr(\text{t1}, \text{t2}, \text{t3}) = Pr(\text{t1})Pr(\text{t2})Pr(\text{t3})$$
$$Pr(\text{t1}, \text{t2}, \text{t3}) = Pr(\text{t1}, \text{t3})Pr(\text{t2})$$

more independence models

- for 4-grams, there are 14 independence models
- generally: the number equals the *Bell number* less one, where the Bell number B_n can be computed recursively as:

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

- but most of these are of limited relevance in collocation mining, as they subsume elements of earlier collocations

statistical association measures

where m_{ij} represents the cell frequency expected according to independence:

G^2 likelihood ratio statistic, computed as:

$$2 * \sum_i \sum_j (n_{ij} * log \frac{n_{ij}}{m_{ij}}) \qquad (1)$$

χ^2 Pearson's χ^2 statistic, computed as:

$$\sum_i \sum_j \frac{(n_{ij} - m_{ij})^2}{m_{ij}} \qquad (2)$$

statistical association measures (cont.)

pmi point-wise mutual information score, computed as $\log n_{11}/m_{11}$

dice the Dice coefficient, computed as

$$\frac{n_{11}}{n_{1.} + n_{.1}} \tag{3}$$

Augmenting collocation detection with additional information

- Use parts of speech information

Tag Pattern	Example
A N	*linear function*
N N	*regression coefficients*
A A N	*Gaussian random variable*
A N N	*cumulative distribution function*
N A N	*mean squared error*
N N N	*class probability function*
N P N	*degrees of freedom*

Table 5.2 Part of speech tag patterns for collocation filtering. These patterns were used by Justeson and Katz to identify likely collocations among frequently occurring word sequences.

- other (machine prediction) tools

Named Entity recognition

```
> sp <- spacy_parse(txt, tag = TRUE)
> entity_consolidate(sp)
   doc_id sentence_id token_id          token          lemma    pos    tag e
1   text1           1        1  Pierre_Vinken  pierre_vinken ENTITY ENTITY
2   text1           1        2              ,              ,  PUNCT      ,
3   text1           1        3   61_years_old    61_year_old ENTITY ENTITY
4   text1           1        4              ,              ,  PUNCT      ,
5   text1           1        5           will           will   VERB     MD
6   text1           1        6           join           join   VERB     VB
7   text1           1        7            the            the    DET     DT
8   text1           1        8          board          board   NOUN     NN
9   text1           1        9             as             as    ADP     IN
10  text1           1       10              a              a    DET     DT
11  text1           1       11   nonexecutive   nonexecutive    ADJ     JJ
12  text1           1       12     \n             \n           SPACE     SP
13  text1           1       13       director       director   NOUN     NN
14  text1           1       14        Nov._29        nov._29 ENTITY ENTITY
15  text1           1       15              .              .  PUNCT      .
```

Quantities for comparing texts

Length in characters, words, lines, sentences, paragraphs, pages, sections, chapters, etc.

Readability statistics Use a combination of syllables and sentence length to indicate "readability" in terms of complexity

Vocabulary diversity (At its simplest) involves measuring a *type-to-token ratio* (TTR) where unique words are types and the total words are tokens

Word (relative) frequency counts or proportions of words

Theme (relative) frequency counts or proportions of (coded) themes

Lexical Diversity

- Basic measure is the TTR: Type-to-Token ratio
- Problem: This is very sensitive to overall document length, as shorter texts may exhibit fewer word repetitions
- Special problem: length may relate to the introdution of additional subjects, which will also increase richness

Lexical Diversity: Alternatives to TTRs

TTR $\frac{\text{total types}}{\text{total tokens}}$

Guiraud $\frac{\text{total types}}{\sqrt{\text{total tokens}}}$

D (Malvern et al 2004) Randomly sample a fixed number of tokens and count those

MTLD the mean length of sequential word strings in a text that maintain a given TTR value (McCarthy and Jarvis, 2010) – fixes the TTR at 0.72 and counts the length of the text required to achieve it

Vocabulary diversity and corpus length

- In natural language text, the rate at which new types appear is very high at first, but diminishes with added tokens

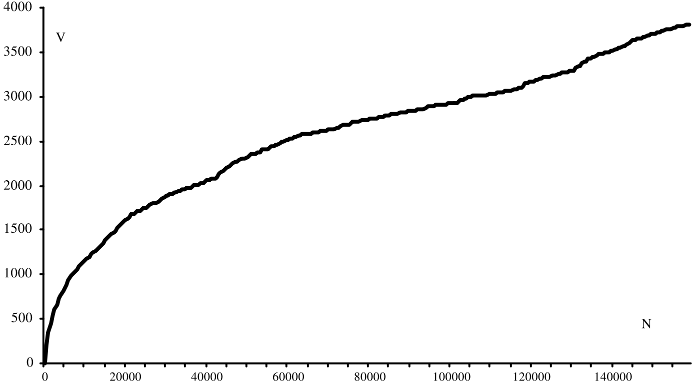


Fig. 1. Chart of vocabulary growth in the tragedies of Racine (chronological order, 500 token intervals).

Vocabulary Diversity Example

- Variations use automated segmentation – here approximately 500 words in a corpus of serialized, concatenated weekly addresses by de Gaulle (from Labbé et. al. 2004)
- While most were written, during the period of December 1965 these were more spontaneous press conferences

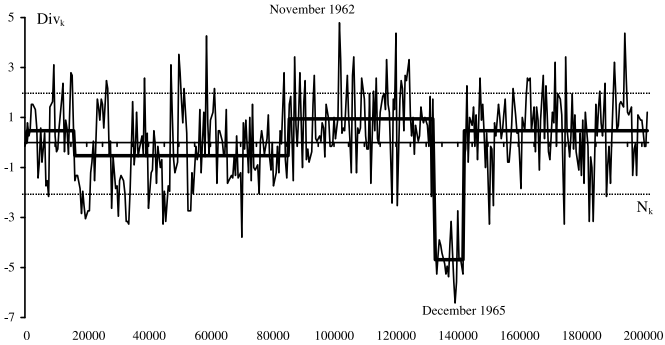


Fig. 8. Evolution of vocabulary diversity in General de Gaulle's broadcast speeches (June 1958–April 1969).

Complexity and Readability

- Use a combination of syllables and sentence length to indicate "readability" in terms of complexity
- Common in educational research, but could also be used to describe textual complexity
- Most use some sort of sample
- No natural scale, so most are calibrated in terms of some interpretable metric
- Implemented in **quanteda** as `textstat_readability()`

Flesch-Kincaid readability index

- F-K is a modification of the original Flesch Reading Ease Index:

  $$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

  Interpretation: 0-30: university level; 60-70: understandable by 13-15 year olds; and 90-100 easily understood by an 11-year old student.

- Flesch-Kincaid rescales to the US educational grade levels (1–12):

  $$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$$

Gunning fog index

- Measures the readability in terms of the years of formal education required for a person to easily understand the text on first reading
- Usually taken on a sample of around 100 words, not omitting any sentences or words
- Formula:

$$0.4\left[\left(\frac{\text{total words}}{\text{total sentences}}\right)+100\left(\frac{\text{complex words}}{\text{total words}}\right)\right]$$

  where complex words are defined as those having three or more syllables, not including proper nouns (for example, Ljubljana), familiar jargon or compound words, or counting common suffixes such as -es, -ed, or -ing as a syllable