

Quantitative Text Analysis – Essex Summer School

Advanced representations of text. Preprocessing with unsupervised methods

dr. Martijn Schoonvelde

University of Groningen

Today's class

- Advanced representations of text
- Preprocessing text and unsupervised methods
 - Denny and Spirling (2018)
- Lab session
- Breakout groups research interests

Advanced representations of text

Yesterday, we talked about how to construct a simple **a bag of words** consisting on plain unigrams - this is in many ways the **default** for many QTA applications

But there are many more ways that we can add information to our bag of words. Grimmer, Roberts and Stewart (2022) encourage us to **rethink the default**

- Bi-grams, trigrams
- Collocations
- Part of speech tagging

Biden inaugural address

"So now, on this hallowed ground where just days ago violence sought to shake this Capitol's very foundation, we come together as one nation, under God, indivisible, to carry out the peaceful transfer of power as we have for more than two centuries." (Biden, 2021)



Biden inaugural address

```
> biden_tokens <- tokens(biden_sentence,
+                         remove_punct = TRUE) %>%
+                         tokens_tolower()
> as.character(biden_tokens)
[1] "so" "now" "on"  "this"  "hallowed" "ground" "where" "just"   "days"  "ago"
[11] "violence" "sought" "to"    "shake"  "this"    "capitol's" "very"   "foundation"
[21] "we" "come"  "together" "as"    "one"    "nation"  "under"  "god"    "indivisible"
[31] "to" "carry" "out"   "the"   "peaceful" "transfer" "of"     "power"  "as"    "we"
[41] "have" "for"   "more"  "than"  "two"    "centuries"
```

N-grams

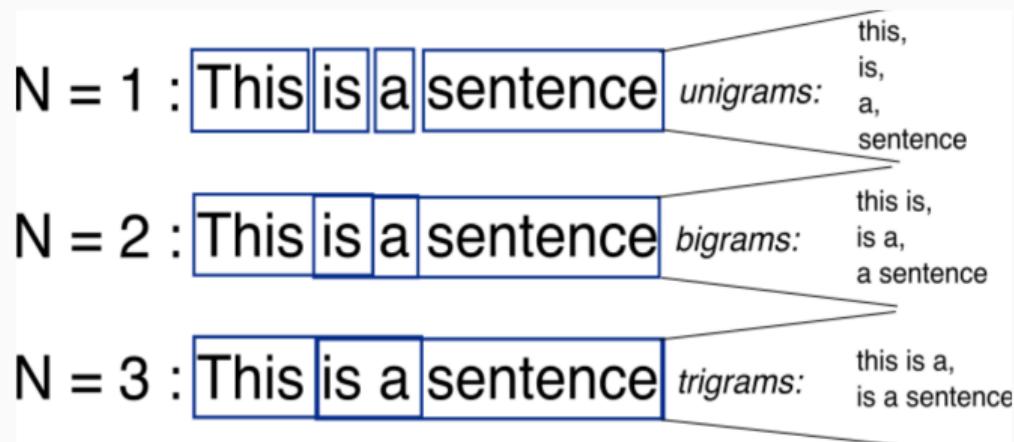


Image credit: <https://compsocialscience.github.io>

N-grams

```
> biden_tokens <- tokens(biden_sentence,
+                         remove_punct = TRUE) %>%
+   tokens_tolower() %>%
+   tokens_ngrams(2)
> as.character(biden_tokens)

"so_now" "now_on" "on_this" "this_hallowed" "hallowed_ground" "ground_where"
"where_just" "just_days" "days_ago" "ago_violence" "violence_sought" "sought_to"
"to_shake" "shake_this" "this_capitol's" "capitol's_very" "very.foundation"
"foundation_we" "we_come" "come_together" "together_as" "as_one"
etc..
```

Bigrams can be useful to account for, for example, **negations**. However, we'll have many more **unique types** in our dfm, and fewer examples to generalize from

Collocations

A special case of n-grams are collocations: pairs of words that occur **more frequently than expected** based on their underlying occurrence (Van Atteveldt *et al.* 2022)

- The usefulness of collocations lies in the fact that they often have a special meaning (e.g., New York, political science) or can disambiguate a term (e.g., *sound* in a *sound proposal* or *loud sound* (Van Atteveldt *et al.* 2022)

Identification collocations can be based on pointwise mutual information, chi-square or other criteria

Collocations

```
> library(quanteda.textstats)
> data_corpus_inaugural %>%
+   tokens(remove_punct = TRUE) %>%
+   tokens_remove(stopwords("en"), padding = TRUE) %>%
+   textstat_collocations(min_count = 10) %>%
+   arrange(-lambda) %>%
+   head()
```

	collocation	count	count_nested	length	lambda	z
32	chief magistrate	10	0	2	10.463551	11.71123
28	vice president	18	0	2	9.495594	13.89318
15	four years	26	0	2	8.636985	18.08532
18	god bless	16	0	2	8.632746	17.06073
1	united states	158	0	2	7.904066	42.89094
3	fellow citizens	78	0	2	7.840842	34.65933

Part of speech (POS) tagging

In part-of-speech tagging or POS tagging, each word is enriched with information on its function in a sentence: verb, noun, adjective, adverb, pronoun, etc.

This can be useful as an input to our bag of words.

- If we are interested in sentiment we may want to include just **adjectives or adverbs**.
- POS tags may also help with disambiguation.

Overview of part-of-speech (POS) tags.

Part of speech	Example	UDPipe/Spacy Tag	Penn Treebank Tag
Noun	apple	NOUN	NN, NNS
Proper Name	Carlos	PROPN	NNP
Verb	write	VERB	VB, VBD, VBP, ..
Auxiliary verb	be, have	AUX	(same as verb)
Adjective	quick	ADJ	JJ, JJR, JJS
Adverb	quickly	ADV	RB
Pronoun	I, him	PRON	PRP
Adposition	of, in	ADP	IN
Determiner	the, a	DET	DT

Source: Van Atteveldt *et al.*
2022

Part of speech (POS) tagging

For example, kind is a positive emotion word, found in most **positive sentiment dictionaries**.
But in English, kind may mean:

- (noun) “a group of people or things having similar characteristics”
- (adjective) “having or showing a friendly, generous, and considerate nature”
- (adverb) “to some extent”

If we want to measure the sentiment of a text (i.e., how positive or negative is it) we **need to account** for this

Source example: keynote Ken Benoit at IC2S2, 2019

Part of speech (POS) tagging

POS tagging in R can be done with **UDPipe** library (Wijffels, 2022) which provides language-agnostic tokenization, tagging, lemmatization and dependency parsing of raw text.

```
> library(udpipe)
> udpipe(biden_sentence, "english") %>%
+   select(token_id:upos)
```

	token_id	token	lemma	upos
1	1	So	so	ADV
2	2	now	now	ADV
3	3	,	,	PUNCT
4	4	on	on	ADP
5	5	this	this	DET
6	6	hallowed	hallow	ADJ
7	7	ground	ground	NOU
etc.				

Part of speech (POS) tagging

The output of `udpipe()` is a **dataframe with one token per row** and each column a linguistic feature

- With some wrangling we can take this as an input for constructing a bag of words

```
> parsed_biden_sentence <- udpipe(biden_sentence, "english")
> names(parsed_biden_sentence)
[1] "doc_id"    "paragraph_id" "sentence_id" "sentence"   "start"    "end"     "term_id"
[8] "token_id"   "token"      "lemma"      "upos"       "xpos"     "feats"    "head_token_id"
[15] "dep_rel"    "deps"      "misc"
```

Preprocessing text: theory



Preprocessing text: practice?



- Aim of preprocessing is to make “inputs to a given analysis less complex in a way that does not adversely affect the interpretability or substantive conclusions of the subsequent model” (Denny and Spirling 2018)
 - Simpler data, not too much information loss
- Standard in NLP and information retrieval, but does it make sense for political scientists as well?
 - In particular, when going from **supervised to unsupervised methods**
 - Goal is different: not classification but detecting latent structure
 - Supervised methods give us a clear benchmark, unsupervised methods do not

- “For just seven possible (binary) preprocessing steps, there would be $2^7 = 128$ possible models to run and analyze”
- Possibility of ‘heading down “forking paths of inference” (Gelman and Loken 2014)
 - End result may crucially depend on arbitrary steps
- Denny and Spirling check this possibility for multiple datasets, and provide an R package that implements solutions
- Argument: preprocessing requires substantive knowledge + statistical check

Preprocessing steps

1. Punctuation
2. Numbers
3. Lowercasing
4. Stemming
5. W Stopword removal
6. 3 n-gram inclusion
7. Infrequently Used Terms

NB: not all of these decisions are really binary; many possible stop word lists for example

Application 1: Wordfish on UK manifestos

- Wordfish model (Slapin and Proksch 2008) Labour and Conservative manifestos over 4 elections (1983 - 1997)
 - Estimates latent ideological position for each text
- Prior belief of order of manifestos:

$\text{Lab}_{1983} < \text{Lab}_{1987} < \text{Lab}_{1992} < \text{Lab}_{1997} < \text{Con}_{1992} < \text{Con}_{1997} < \text{Con}_{1987} < \text{Con}_{1983}$.

Wordfish positions of UK manifestos

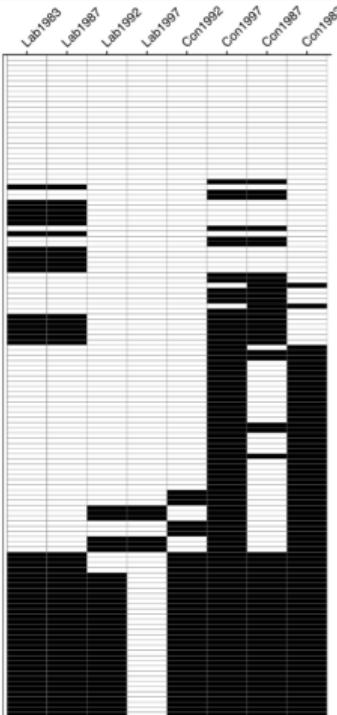


Figure 1. Wordfish results for the 128 different preprocessing possibilities. Each row of the plot represents a different specification. A white bar implies that the manifesto for that year is in the correct place as regards our priors. A black bar implies it was misplaced.

NB: twelve different orderings

Application 2: Topic models of Congress press releases

- LDA topic models (Blei, Ng & Jordan 2003) on congressional press releases
 - Unsupervised method to learn about themes
- Main assumption: words in a text are generated from a set of topics

Topic models

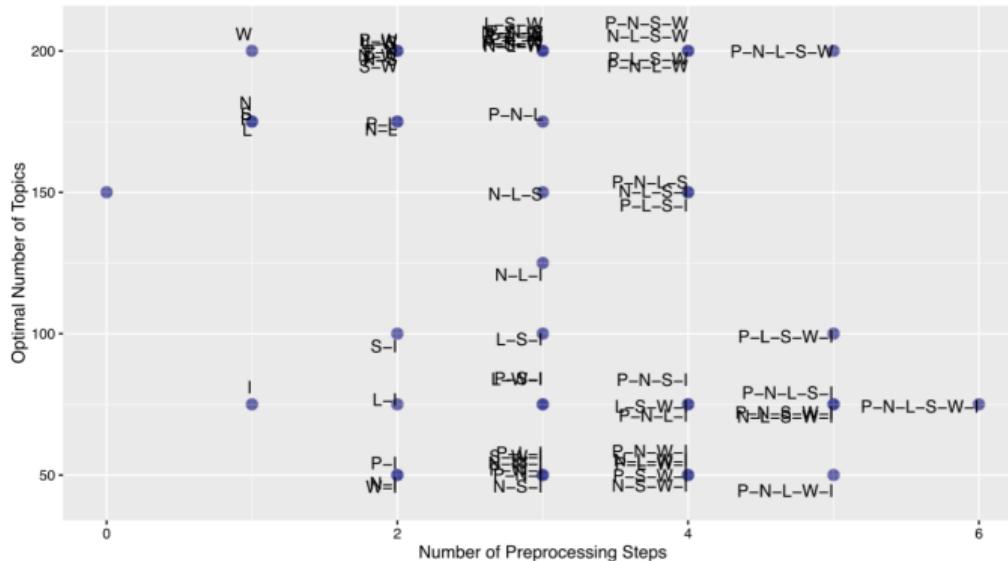


Figure 2. Plot depicting the optimal number of topics (as selected via perplexity) for each of 64 preprocessing specifications not including trigrams. On the x-axis is the number of preprocessing steps, and the y-axis is the number of topics. Each point is labeled according to its specification.

NB: “optimal” number of topics varies with preprocessing steps

Topic models

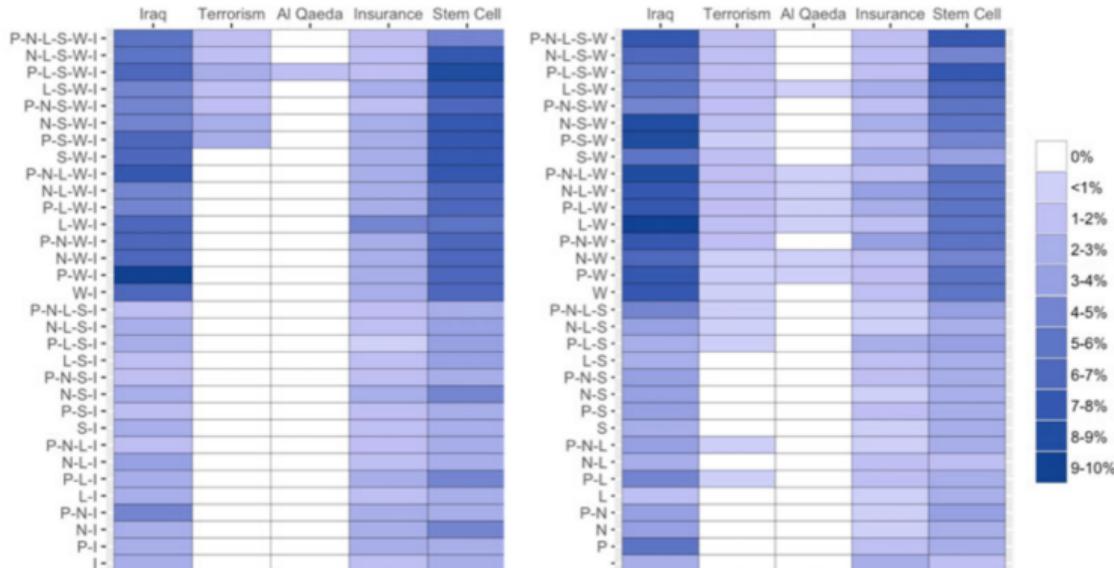


Figure 3. Plots depicting the percentage of topic top-20 terms which contain the stem of each of five keywords, for each of 64 preprocessing steps (thus excluding those which include trigrams). The number of topics for specifications fit to each of the 64 DFM were determined through tenfold cross-validation, minimizing the model perplexity.

NB: topical content of documents varies with preprocessing steps

Pretext scores

- Use theory to guide your preprocessing
- Check sensitivity of outcomes to preprocessing steps using the **preText** library
 - http://www.mjdenny.com/getting_started_with_preText.html
 - Summarized by a **Pretext** score (between 0 and 1)
 - If Pretext scores do not vary with preprocessing, then results are robust
 - If Pretext scores do vary with preprocessing, evaluate outcomes against different preprocessing steps and strength of prior beliefs

Conclusion

- Pre-processing is not a neutral step in QTA
- Denny and Spirling (2018) find preprocessing steps matter a lot for our interpretation of unsupervised models
- But further thinking is needed on how and why these preprocessing steps matter