university of groningen

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

Advanced representations of text

Yesterday, we talked about how to construct a simple a bag of words consisting of 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
- Word embeddings

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

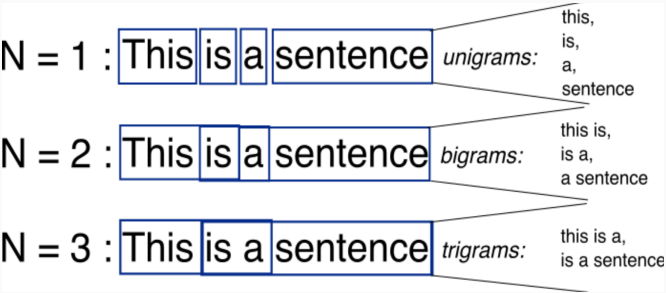

Image credit: `https://compsocialscience.github.io`

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

Pointwise Mutual Information (PMI)

- PMI is a measure of association that quantifies the discrepancy between the probability of two events occurring together and the probabilities of the events occurring separately.

The PMI of two events x and y is defined as:

$$\text{PMI}(x; y) = \log \frac{p(x, y)}{p(x)p(y)}$$

where $p(x, y)$ is the joint probability of x and y, and $p(x)$ and $p(y)$ are the probabilities of x and y independently.

- PMI measures the extent to which knowing one of these variables tells us about the other. A high PMI value indicates a strong association between x and y, while a PMI close to zero suggests no association.

Calculating PMI from a text sample

We use a small example:

"The quick brown fox jumps over the lazy dog. The dog sleeps in the sun."

We will calculate the PMI for the collocation: **"lazy dog"**

Source: Bayu Setiyawan on dribbble.com

Calculating PMI from a text sample

Step 1: Word and Bigram Count

- Total words: 15
- Total bigrams: 14

Frequencies:

- lazy: $1 \rightarrow P(\text{lazy}) = \frac{1}{15}$
- dog: $2 \rightarrow P(\text{dog}) = \frac{2}{15}$
- lazy dog: $1 \rightarrow P(\text{lazy, dog}) = \frac{1}{14}$

Step 2 : Calculate PMI

$$\text{PMI(lazy, dog)} = \log_2 \left(\frac{1/14}{(1/15) \cdot (2/15)} \right)$$

$$= \log_2 \left(\frac{1/14}{2/225} \right) = \log_2 \left(\frac{225}{28} \right) \approx \log_2(8.04) \approx \boxed{3.01}$$

Step 3: Interpretation

- PMI ≈ 3.01
- "lazy" and "dog" co-occur more than expected if they were independent
- Suggests a meaningful collocation (0 = no relationship; < 0 negative relationship; > 3 strong assosiation

- **But**: PMI is sensitive to low-frequency events:
  - Rare words that co-occur even once can produce inflated PMI.
  - That's why people often use PMI with frequency thresholds or the normalized PMI (NPMI).

Source: Bayu Setiyawan on dribbble.com

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 grammatical 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"
```

Word Embeddings

- Text as data research in social science has traditionally used the bag of words model.
- However, this is evolving rapidly. Word embeddings, a different representation of text, are increasingly popular
  - Embeddings represent words as real-valued vectors of numbers.
  - This technique leverages the co-occurrence of words within texts, using dimension reduction methods to construct a language representation.
  - The length of these vectors "corresponds to the nature and complexity of the multidimensional space in which we are seeking to 'embed' the word" (Rodriguez & Spirling, 2022).

Word embeddings

Two core ideas (Van Atteveldt *et al.* 2022):

1. Meaning of a word can be expressed using a relatively small embedding vector, generally consisting of around 300 numbers which can be interpreted as dimensions of meaning.
2. These embedding vectors can be derived by scanning the context of each word in millions and millions of documents.

Embeddings can then be used as features for further analysis. A model fit on embedding vectors gets a "head start" since the vectors for words like "great" and "fantastic" will already be relatively close to each other, while in a DTM they are treated independently.

- Meaning is baked into these embeddings

Intuition Behind Word Embeddings

You shall know a word by the company it keeps (Firth, 1957)

Distributional hypothesis – meaning of a word can be extracted by looking, over many texts, by the words that occur around it

- **Contextual meaning:** Words that appear in similar contexts have similar vectors.
- **Semantic relationships:** Reflects semantic relationships between words (e.g., king - man + woman = queen).

This may have exciting substantive implications for us as social science researchers:

- 'if the distance between "immigrants" and "hard-working" is smaller for liberals than for conservatives, we learn something about their relative worldviews' (Rodriguez & Spirling, 2022)

Word Embeddings vs. Bag of Words

Bag of Words (BoW):

- **Definition:** Represents text as an unordered collection of words.
- **Features:** Simple, based on word frequency.
- **Limitations:**
  - Ignores context.
  - High dimensionality / sparse vectors.
- **Comparison:**
  - "dog" , ... , ... $->$ [1, 0, 0, ...]
  - ... ,"bark", ... $->$ [0,1, 0, ...]
  - ... , ... ,"run" $->$ [0, 0, 1, ...]

Word Embeddings:

- **Definition:** Dense vector representations in continuous space.
- **Features:** Context-aware, captures semantics.
- **Advantages:**
  - Captures relationships and context
  - Low dimensionality / dense vectors.
- **Comparison:**
  - "dog" $->$ [0.23, -0.15, 0.67, ...]
  - "bark" $->$ [0.26, -0.20, 0.30, ...]
  - "run" $->$ [0.20, -0.10, 0.8, ...]

Word Embeddings vs. Bag of Words

Bag of Words	Word embeddings
One-hot encoding	Vector in a semantic space
DxN	*NxV*
No context	Estimated from context
Meaning exogenous	Meaning learned
Input to a model	Output from a model

D = number of documents
N = number of words
V = number of embedding dimensions

Preprocessing text: theory

Preprocessing text: practice?

Preprocessing in Analysis - Denny and Spirling 2018

The aim of preprocessing is to simplify the inputs to an analysis in a way that does not adversely affect the interpretability or substantive conclusions of the subsequent model (Denny and Spirling, 2018).

- Focus on maintaining a balance between simplicity and minimal information loss.

While preprocessing is standard in Natural Language Processing (NLP) and information retrieval, its suitability for political scientists, especially when transitioning from supervised to unsupervised methods, warrants examination.

- Objective shift: Moving from classification goals to detecting latent structures.
- Methodological shift: Supervised methods provide clear benchmarks, whereas unsupervised methods often do not.

Preprocessing in Analysis - Denny and Spirling 2018

- "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 checking of robustness

Preprocessing steps

1. **P**unctuation
2. **N**umbers
3. **L**owercasing
4. **S**temming
5. **W** Stopword removal
6. **3** n-gram inclusion
7. **I**nfrequently Used Terms

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

Application 1: Analyzing UK Manifestos with Wordfish

The Wordfish model, developed by Slapin and Proksch (2008), is applied to Labour and Conservative party manifestos across four elections from 1983 to 1997.

- Purpose: Estimates the latent ideological positions of texts.

The following diagram represents the anticipated order of the manifestos based on their ideological positioning:

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

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

Using Latent Dirichlet Allocation (LDA) models, developed by Blei, Ng, and Jordan (2003), to analyze congressional press releases.

- Approach: An unsupervised method that identifies and learns about the underlying themes within the text.

The primary assumption of LDA is that words in a text are generated from a latent set of topics, which governs the thematic structure of the documents.

Topic models

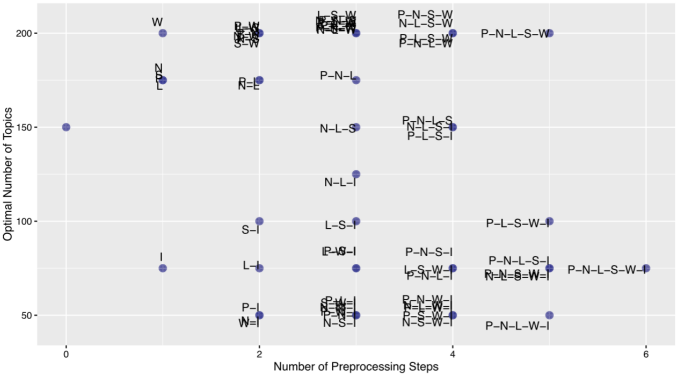

Figure 2. Plot depicting the optimal number of topics (as selected via perplexity) for each of 64 preprocessing specifications not including trigrams. On the x-axis is the number of preprocessing steps, and the y-axis is the number of topics. Each point is labeled according to its specification.

NB: "optimal" number of topics varies with preprocessing steps

Topic models

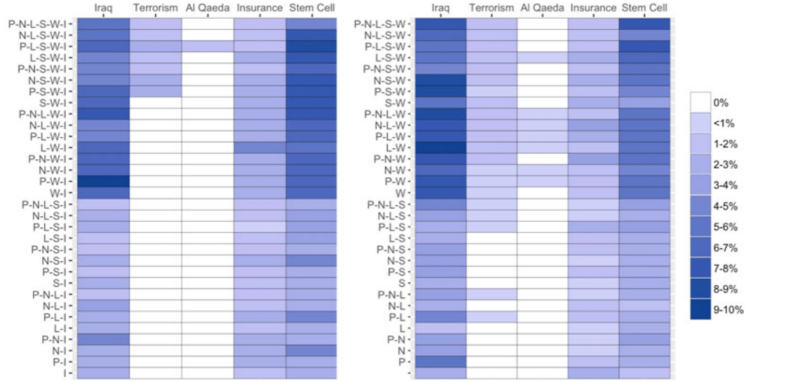

Figure 3. Plots depicting the percentage of topic top-20 terms which contain the stem of each of five keywords, for each of 64 preprocessing steps (thus excluding those which include trigrams). The number of topics for specifications fit to each of the 64 DFMs were determined through tenfold cross-validation,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

NB: topical content of documents varies with preprocessing steps

Pretext Scores: Enhancing Data Preprocessing

Use theory to guide the preprocessing steps in your analysis

Assess the sensitivity of your analysis outcomes to different preprocessing steps with the **preText** package.

- Reference: `http://www.mjdenny.com/getting_started_with_preText.html`
- The analysis is summarized by a **Pretext** score, which ranges between 0 and 1.

Interpreting Pretext Scores

- If Pretext scores do not vary with different preprocessing steps, your results are likely robust.
- If Pretext scores vary, further evaluate your outcomes against different preprocessing steps and the strength of your 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