Text factorisation — I: Bag-of-words methods

Course "Text-as-data analysis of international trade"

Dmitriy Skougarevskiy

The Institute for the Rule of Law, European University at Saint Petersburg

Saint Petersburg
25 April 2018

Course outline

1. Preferential Trade Agreements and International Economic Order
2. Gravity and Gravitas
3. **Text factorisation — I: Bag-of-words methods**
4. Text factorisation — II: Distributive semantics
5. Welfare effects of Preferential Trade Agreements

Outline

1. What's in a bag?
   - BOW representation
   - BOW applications
2. Improvements
   - Tokenisation
   - Re-weighting of DTM
3. DTM Factorisation
   - Latent Semantic Analysis

Outline

Bag-of-words

- A simple yet versatile way to quantify any text is a *bag of words*:
  - `John likes to watch movies. Mary likes movies too.`
    $\Updownarrow$
    `"John", "likes", "to", "watch", "movies", "Mary", "likes", "movies", "too"`
  - `John also likes to watch football games.`
    $\Updownarrow$
    `"John", "also", "likes", "to", "watch", "football", "games"`
- In itself, it is of limited use. However, one can build a vocabulary of unique terms and store their counts:

	John	likes	to	watch	movies	Mary	too	also	football	games
doc1	1	2	1	1	2	1	1	0	0	0
doc2	1	1	1	1	0	0	0	1	1	1

- This is a unigram document-term matrix

Bag-of-words & DTMs

- A simple yet versatile way to quantify any text is a *bag of words*:
    - `John likes to watch movies. Mary likes movies too.`
      $\Updownarrow$
      `"John", "likes", "to", "watch", "movies", "Mary", "likes", "movies", "too"`
    - `John also likes to watch football games.`
      $\Updownarrow$
      `"John", "also", "likes", "to", "watch", "football", "games"`
- In itself, it is of limited use. However, one can build a vocabulary of unique terms and store their counts:

	John	likes	to	watch	movies	Mary	too	also	football	games
doc1	1	2	1	1	2	1	1	0	0	0
doc2	1	1	1	1	0	0	0	1	1	1

- This is a unigram document-term matrix (DTM)

Unigrams vs. *n*-grams vs. *q*-grams

- One can put more than one word in a bag — bigram here, *n*-gram in general:
  - `"John_likes", "likes_to", "to_watch", "watch_movies", "movies_Mary", "Mary_likes", "likes_movies", "movies_too", ...`
- Or capitalise on character-level information — *q*-gram:
  - `"jo", "hn", "n_l", "li", "ik", "ke", "es", "s_", ...`
- Variance-bias trade-off: larger n, q leads to lower generalisability of the language model
  - need for sparse representations: most *n*-grams are never used
- What shall one do with punctuation?

Outline

1. What's in a bag?
   - BOW representation
   - BOW applications
2. Improvements
   - Tokenisation
   - Re-weighting of DTM
3. DTM Factorisation
   - Latent Semantic Analysis

Figure 1: Relative counts in Google *n*-grams (Michel et al., 2011) from Pechenick et al. (2015, fig. 2)

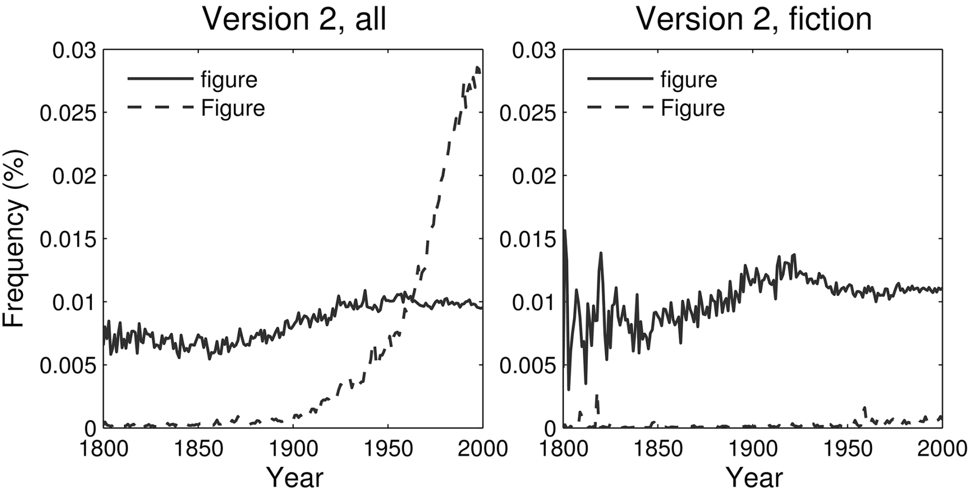


Figure 2: Dendrogram for *n*-gram distances between languages (Feinerer et al., 2013, fig. 4)

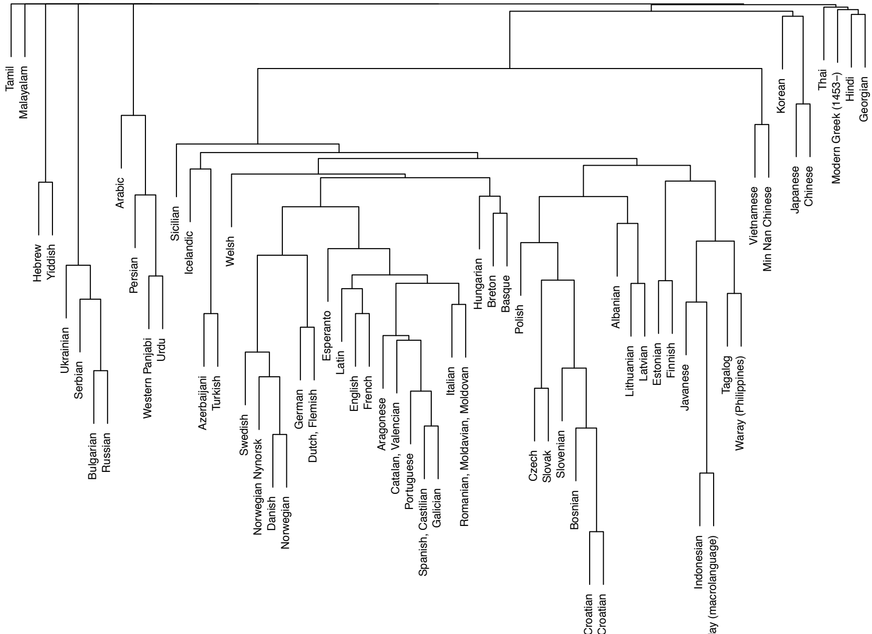
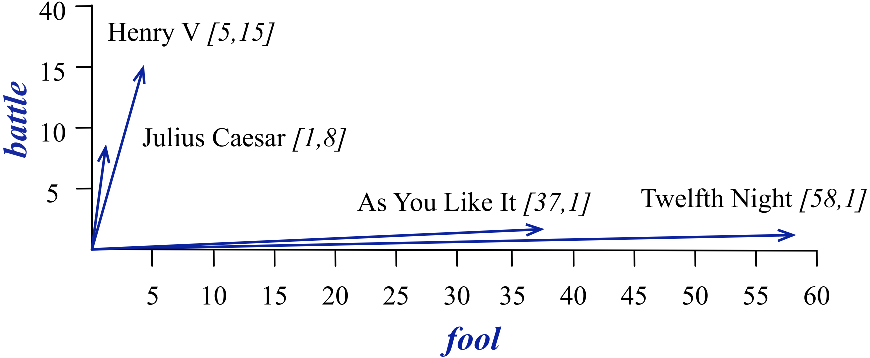


Figure 3: TDM of select terms in Shakespeare plays (Jurafsky and Martin, 2017, fig. 15.1)

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Figure 4: 2 dimensions of Shakespeare TDM (Jurafsky and Martin, 2017, fig. 15.3)



Outline

1. What's in a bag?
   - BOW representation
   - BOW applications
2. Improvements
   - Tokenisation
   - Re-weighting of DTM
3. DTM Factorisation
   - Latent Semantic Analysis

Figure 5: Partition examples of Dickens's "*Tale of Two Cities*" (Williams et al., 2015, fig. 1A): clause, phrase, word, grapheme, letter

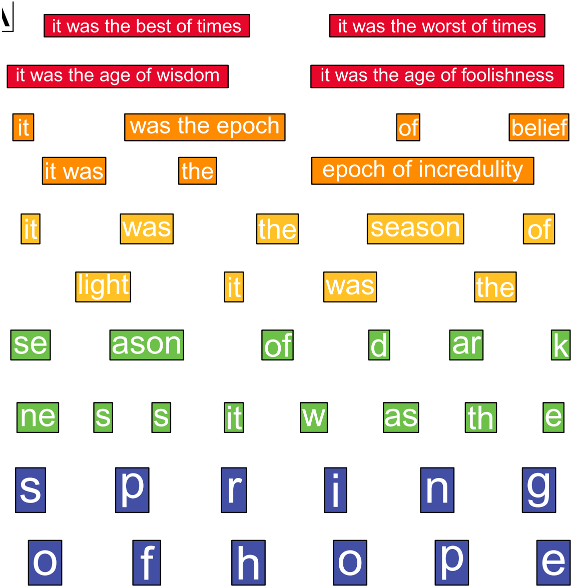


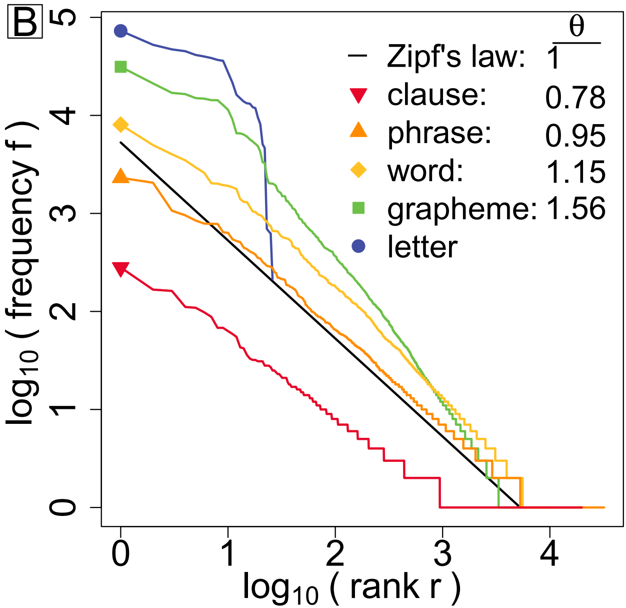
Table 1: Tokenisation is not as easy as it seems (`https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization`)

“I said, ’what’re you? Crazy?”’ said Sandowsky.

	Naïve whitespace	Apache Open NLP	Stanford	Custom	Ideal
1		“	“	“	“
2	“i	i	i	i	i
3	said,	said	said	said	said
4		,	,	,	,
5	what’re	what	‘		
6			what	what’re	what
7		re	re		are
8	you?	you	you	you	you
9		?	?	?	?
10	crazy?’”	crazy	crazy	crazy	crazy
11		?	?	?	?
12					
13	said	said	said	said	said
14	sandowsky.	sandowsky	sandowsky	sandowsky	sandowsky
15		.	.	.	.

Outline

Figure 6: Zipf's law ($F(w) \propto rank(w)^{-\theta}$) at various token level (Williams et al., 2015, fig. 1B): clause, phrase, word, grapheme, letter



Dealing with Zipf's law

- Zipf's law suggests that word frequency counts are very skewed
- At the same time, frequent words are not necessarily discriminative "the" vs "a"
- Solution 1: "stopwords" remove such stop words
  - problem: influence word order and any *n*-/*q*-gram measures
- Solution 2: term frequency-inverse document frequency weighting (tf-idf):

$$\mathrm{tf}\left(token\right)_i = \frac{\#occurences\ in\ document\ i}{\#tokens\ in\ document\ i}$$

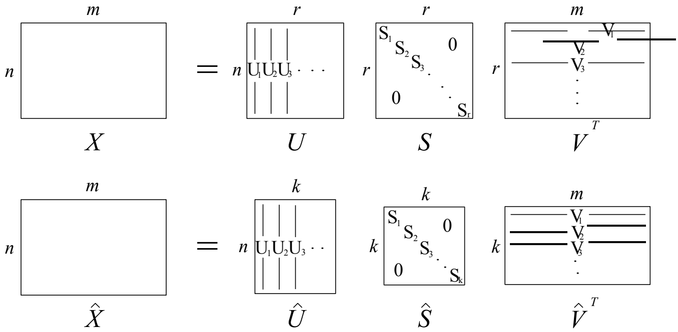
$$\mathrm{idf}\left(token\right) = \ln \frac{\#documents}{\#documents\ with\ token + 1}$$

Outline

How to extract meaning from DTMs

- Widely known way to extract meaning is to apply singular value decomposition to (tf-idf-weighted) DTM (Deerwester et al., 1990):

Figure 7: SVD example (from Radinsky, 2017)



See also: `https://en.wikipedia.org/wiki/File:Topic_model_scheme.webm`

LSA in practice

- In practice, LSA has large computational cost — $O\left(mn^2\right)$ — not always scaleable
- Need to reduce the resulting still-high-dimensional LSA output further, e.g. with PCA
- tf-idf makes a huge difference

Take-aways

- BOW model is a powerful albeit limited way to quantify texts
- Tokenisation is often overlooked but can be a large problem in practice
- Simply applying SVD to DTM can take you a long way

Thank you for your attention!

References — I

Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science 41*(6), 391.

Feinerer, I., C. Buchta, W. Geiger, J. Rauch, P. Mair, and K. Hornik (2013). The `textcat` package for *n*-gram based text categorization in R. *Journal of Statistical Software 52*(6), 1–17.

Jurafsky, D. and J. Martin (2017). *Speech and language processing* (3 ed.). Pearson.

Michel, J.-B., Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. (2011). Quantitative analysis of culture using millions of digitized books. *Science 331*(6014), 176–182.

References — II

Pechenick, E., C. Danforth, and P. Dodds (2015). Characterizing the Google Books corpus: Strong limits to inferences of socio-cultural and linguistic evolution. *PloS one 10*(10), e0137041.

Williams, J., P. Lessard, S. Desu, E. Clark, J. Bagrow, C. Danforth, and P. Dodds (2015). Zipf's law holds for phrases, not words. *Scientific reports 5*, 12209.