Quantitative Text Analysis – Essex Summer School

Word embeddings

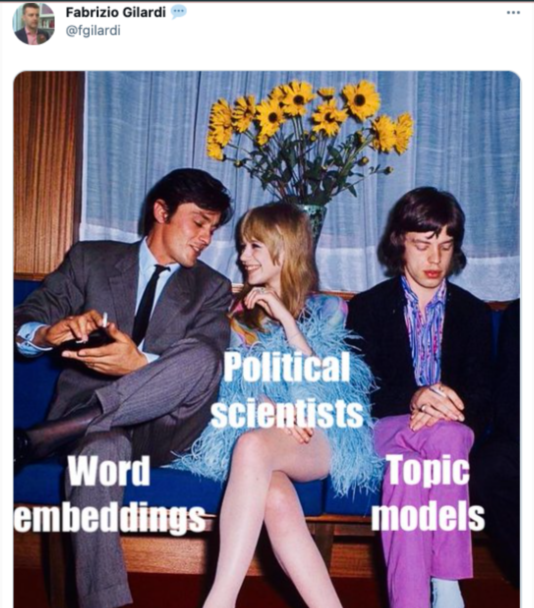
dr. Martijn Schoonvelde

University of Groningen

Today's class

- Word embeddings
- Lab session

Word Embeddings



Word Embeddings

- Many applications of text as data in political science: bag of words
  - But this is changing rapidly
- Word embeddings: different representation of text; words represented as a real-valued vector of of numbers
  - The approach takes advantage of the co-occurrences of words in the same text and constructs a representation of language by using dimension reduction techniques.
  - The length of the word embeddings vector "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

What are word embeddings

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

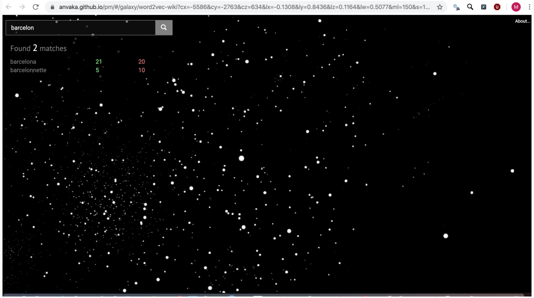
Distributional hypothesis – meaning of a word can be extracted by looking, over many texts, by the words that occur around it

- As opposed to, for example, relational or compositional perspectives on meaning (Eisenstein, 2019)

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)

Visualizing word embeddings



https://bit.ly/35WkD7K

The dataset used for this visualization comes from GloVe, and has 6B tokens, 400K vocabulary, 300-dimensional vectors

What are word embeddings

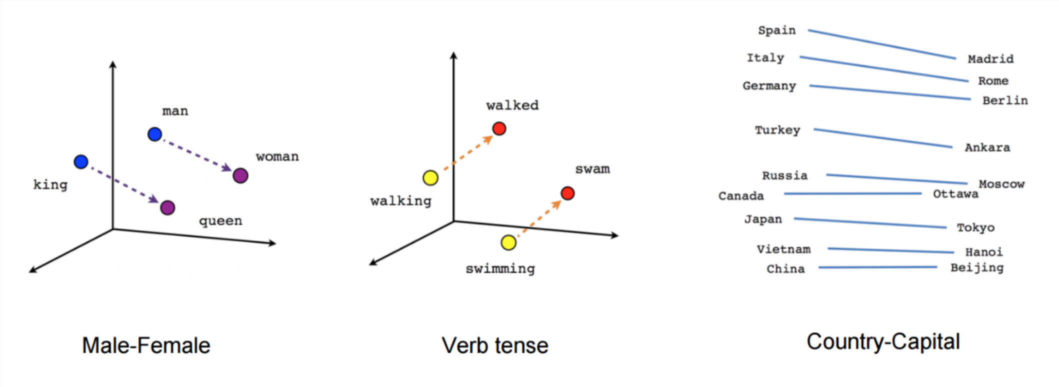
Word embeddings gained fame in NLP when it was demonstrated that they could be used to identify analogies.

- These analogical relationships can be expressed mathematically in terms of their word vectors:

$$\text{V(woman) - V(man)} + \text{V(king)} \approx \text{V(queen)}.$$

1. Start with the vector for "woman";
2. Subtract from it the vector for "man", leaving behind only what is unique about V(woman) as distinct from V(man);
3. Then, add this distinct difference to V(king).
4. You end up with a new vector position: V(woman-man+king). Which word vector, out of thousands of other words, is closest to V(woman-man+king)? In many word embedding models: V(queen).

Word Embeddings



Source:: https://cbail.github.io/textasdata/word2vec/rmarkdown/word2vec.html

Word Representations

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

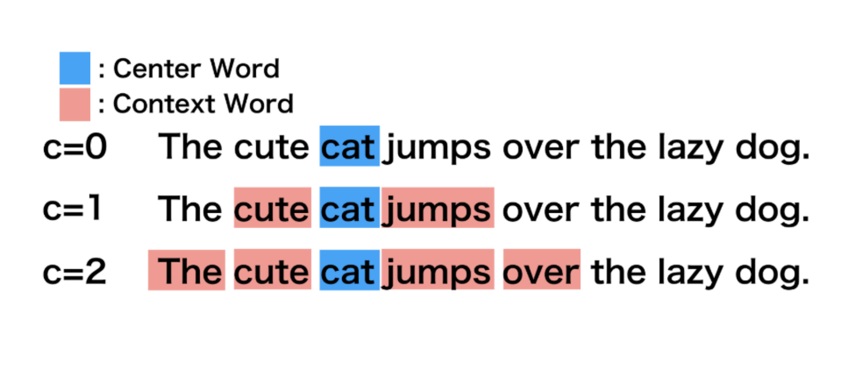
Word Representations

Table 1. Comparing Traditional Approaches with Embeddings

	Traditional Unsupervised	Traditional Supervised	Embeddings
Bag of words	Yes	Yes	No
Example models	Latent Dirichlet allocation; structural topic model	Support vector machines; random forest (RF)	GloVe, Word2Vec
Citations/applications	Quinn et al. (2010), Roberts et al. (2014)	Diermeier et al. (2012), Montgomery and Olivella (2018)	Rheault and Cochrane (2019), Rodman (2019)
Inputs	Document-term matrix	Document-term matrix; labeled *y*	Term-co-occurence matrix
Outputs	Document distribution over topics; topic distribution over words	Term importance matrix (for class prediction)	Word vectors
Example user decisions	Weighting of tokens; number of topics	Training/test split; weighting of tokens; number of trees (RF); number of variables at each split (RF); prior class probabilities	Pretrained or local fit; window size; embedding dimensions; weighting of tokens
Stability concerns	Multiple modes	Sensitivity to training/test set; labeling errors	Algorithmic; corpus characteristics

Source: Rodriguez & Spirling, 2022

Context Window



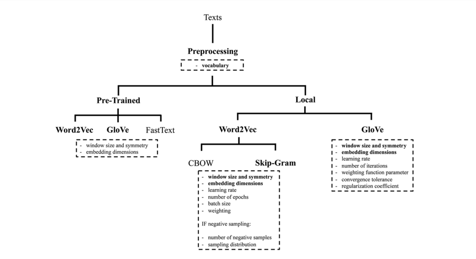
Source:: https://cbail.github.io/textasdata/word2vec/rmarkdown/word2vec.html

Learning word embeddings

There are various algorithms to learn word embeddings vectors:

- Word2Vec (Mikolov *et al.* 2013)
- GloVe (Pennington, Socher, Manning, 2014)

Important to keep in mind: researcher determines the size of the context window, the length of the word embeddings vector and whether to use pre-trained word embeddings or not (see Spirling & Rodriguez, 2022)



Source: Rodriguez & Spirling, 2022

Some applications of word embeddings for social science

- Detecting emergency rhetoric among EU executives (Rauh, 2021)
- Develop domain-specific sentiment dictionaries dictionaries (Rheault *et al.* 2016)

Normality-emergency in executive speeches

Procedure in Rauh (2021):

1. Identify a short list of key words:
    - emergency: *crisis, danger, peril, hazard, threat, risk, disaster, uncertainty, uncertain*
    - normality: *normal, safety, stability, regularity, routine, calm, usual, certainty, certain*.
2. Learn a word embeddings model (GloVe) on the 100 years of House of Commons speeches
3. Identify an additional set of 250 crisis and emergency words closest the average vectors of normality and emergency
4. Use these words to scale EU executive speeches on a normality - emergency dimension (LSS)

Normality-emergency in executive speeches

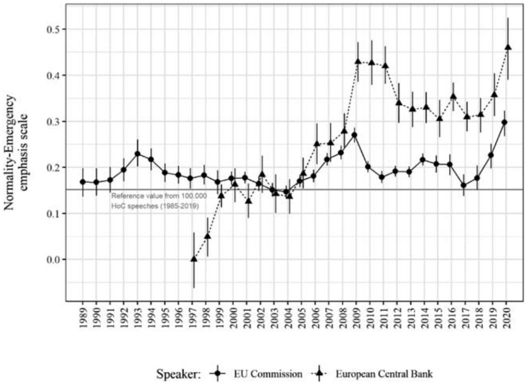


Figure 2. Emergency emphasis in public speeches of supranational executives over time.

Normality-emergency in executive speeches

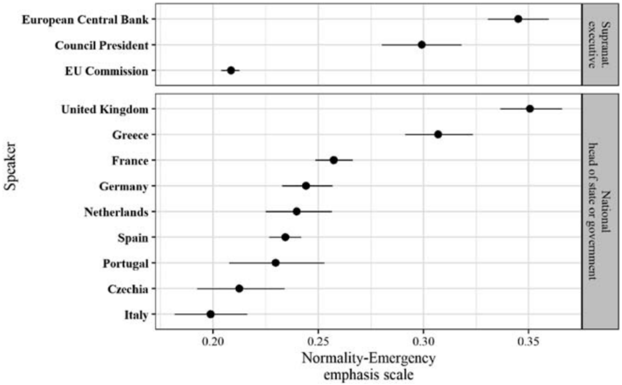


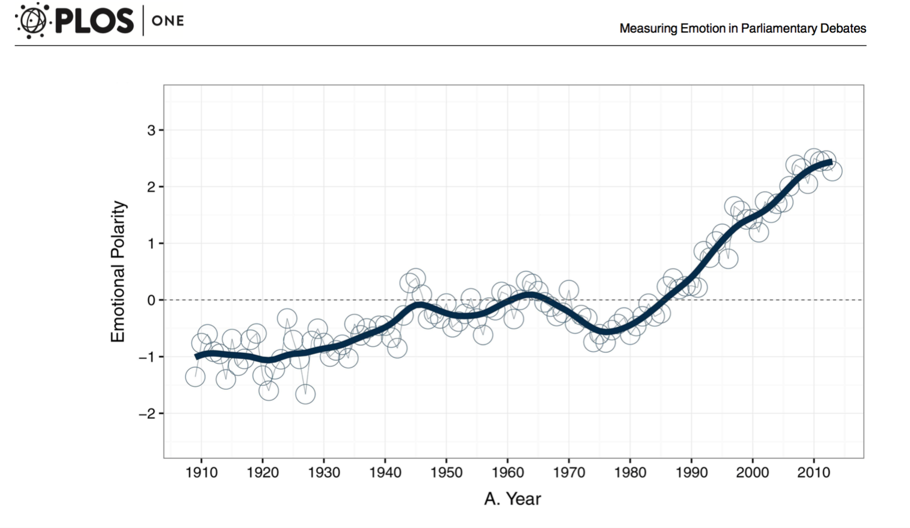
Figure 3. Emergency emphasis in executives' public speeches during the Eurocrisis (2009–2015).

Rheault et al. (2016)

Sentiment analysis – use word embeddings to develop a "domain specific sentiment dictionary", relying on the assumption that words with similar meanings have similar vectors.

- British House of Commons speeches between 1909 and 2013
  - After preprocessing, total of 108,506 unique tokens
  - Create a feature co-occurrence matrix
  - Use GloVe to learn word embeddings
  - Then locate 200 positive and negative 'seed' words in this space
  - With these words located, they can locate other words nearby, leading to a total of 4200 words denoting positive and negative sentiment

Overall sentiment in the HoC



Government and opposition sentiment in the HoC

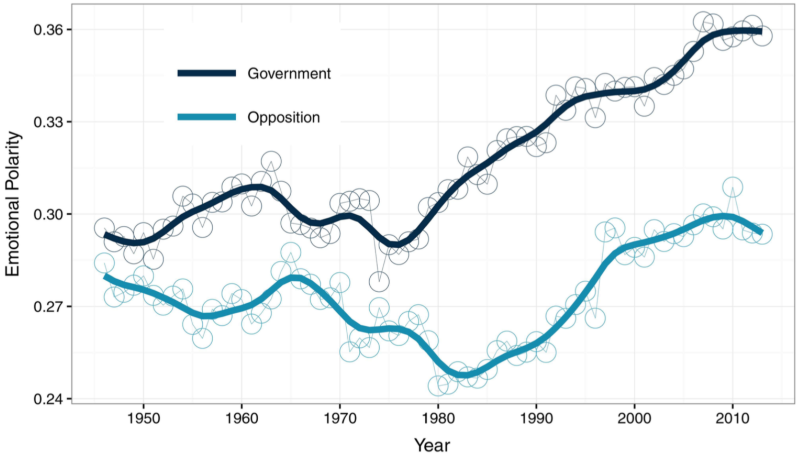


Fig 2. Emotional Polarity of Government and Opposition in Britain, 1946-2013.

doi:10.1371/journal.pone.0168843.g002

Word Embeddings

- Lots of cool possibilities: For example, how does the semantic meaning of words change over time (e.g., liberal and conservative)?
- Do parties shift in *how* they use particular words? For example, does debate vocabulary change over time?
  - See, e.g., work by Milan van Lange and Ralf Futselaar on War debates in Dutch parliament `https://github.com/MilanvanL/debating_evil`

Validation

For on a discussion on validation strategies for word embeddings models in political science, see Spirling & Rodriguez (2022)

Turing test: for embeddings

1. Generate human-generated nearest neighbors for a concept of interest
2. Generate model-produced nearest neighbors for a concept of interest
3. Let coders rate which nearest fit better the definition of a context word
4. Calculate whether coders are equally likely to choose human-generated or model-produced vectors

Transformer models

New developments in embeddings are transformer models

- Can take larger contexts into account when training than earlier embedding did
- Can be trained much more efficiently and thus on more texts
- Can have several embeddings for each word depending on the context in which they appear in a corpus
- Can be fine-tuned on new data which contains different vocabulary

At this point little support in R, but accessible from Python through libraries such as **grafzahl** (Chong, 2023) and **spacyr** (Benoit & Matsuo, 2020)

`Source: Johannes Gruber`