# Vector Semantics



P. S. Langeslag

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

# Topic Vectors vs Word Embeddings

Both are described as company-they-keep approaches; but:

## Topic Vectors

- Use SVD on BoW/TF-IDF models to infer topics
- Hence cannot rely on token context; they compare terms across documents

## Word Embeddings

- Use context tokens to predict the target word, or vice versa.

# Topic Vectors vs Word Embeddings

Both are described as company-they-keep approaches; but:

## Topic Vectors

- Use SVD on BoW/TF-IDF models to infer topics
- Hence cannot rely on token context; they compare terms across documents

## Word Embeddings

- Use context tokens to predict the target word, or vice versa.

But it's a difference of degree:

> When the vectors are computed from short context windows, the most similar words to a target word *w* tend to be semantically similar words with the same parts of speech. When vectors are computed from long context windows, the highest cosine words to a target word *w* tend to be words that are topically related but not similar.
> (Jurafsky & Martin 125)

# Defining Our Terms

## Vector Semantics

Using word vectors to encode *meaning*, whether using a method based on the BoW model or on token context.

## Word Embeddings

A synonym of word vectors, but often used narrowly to refer to a token-context model. We'll use it in the narrow sense, and we'll also refer directly to skipgram or word2vec to avoid ambiguity.

# Term-Document Matrices Contain Two Sets of Vectors

1. Columns may be compared for similarity of terms between documents
2. Rows may be compared to compare terms for the documents in which they occur

(See Jurafsky & Martin §6.3.)

# Word-Word Matrices Show Co-Occurrences

- $|V| * |V|$
- Context may be full document or window of e.g. 4 tokens on either side of target

(See Jurafsky & Martin §6.3.3.)

# Positive Pointwise Mutual Information

- Like TF-IDF but for word-word co-occurrences
- Expresses the observed frequency of co-occurrences relative to their statistically expected co-occurrence, but with negative results zeroed out:
  $PPMI(w, c) = max(log_2 \frac{P(w,c)}{P(w)P(c)}, 0)$

# Word Embeddings

- Static embedding: learn one fixed embedding for each term*
- Contextual embedding: learn different embeddings for different contexts

* or two: one as a target word, one as a context word.

# Skip-Grams with Negative Sampling: Method

- Train a binary window co-occurrence predictor (classifier), **use its weights as embeddings**
- **Self-supervision**: any corpus provides its own gold standard data
- "Negative sampling" refers to randomly* selected words from the corpus serving as gold standard negative matches ("noise words"), against the positive matches from the context window

* Within a certain frequency range.

# Skip-Grams with Negative Sampling: Procedure

1. Assign random embedding vectors
2. Use context to improve the embeddings iteratively, aiming for a maximum score for positive matches and a minimum score for negative matches
3. One matrix for each term as a target, one as a context word

# Window Size Matters!

Levy and Goldberg (2014a) showed that using skip-gram with a window of ±2, the most similar words to the word *Hogwarts* (from the *Harry Potter* series) were names of other fictional schools: *Sunnydale* (from *Buffy the Vampire Slayer*) or *Evernight* (from a vampire series). With a window of ±5, the most similar words to *Hogwarts* were other words topically related to the *Harry Potter* series: *Dumbledore*, *Malfoy*, and *half-blood*.

(Jurafsky & Martin 125)

# Types of Co-Occurrence

## First-Order Co-Occurrence

Terms tend to occur near each other (e.g. *severe* and *storm*).

## Second-Order Co-Occurrence

Terms occur in similar or interchangeable contexts (e.g. *severe* and *harsh*).

## Relational Similarity (the Parallelogram Model)

```
Tokyo - Japan + China = Beijing
```

# Bibliography


Jurafsky, Dan, and James H. Martin. *Speech and Language Processing*. 3rd ed. draft., 2021. http://web.stanford.edu/~jurafsky/slp3/.

Lynn, Shane. "An Introduction to Word Embeddings for Text Analysis," n.d. https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction/.

———. "Word Embeddings in Python with Spacy and Gensim," n.d. https://www.shanelynn.ie/word-embeddings-in-python-with-spacy-and-gensim/.