University of BRISTOL

# 3.4 Vector Semantics

Edwin Simpson

Department of Computer Science,

University of Bristol, UK.

bristol.ac.uk

# Vector Semantics

- Represent words, sentences and documents as points in a **multi-dimensional space**
- Distance encodes **semantic similarity** (similarity of meaning)
- Aim: extract numerical vectors (**embeddings**) from text that capture the text's meaning
- These embeddings can be input to clustering methods and classifiers

# Term-Document Matrix

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

Word vector

Document vector

Counts from Shakespeare plays. Figure 6.3, from Chapter 6, Speech and Language Processing, 3rd edition draft, Jurafsky & Martin (2019).

# Term-Document Matrix

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

Document vector

This is a bag-of-words represented by a fixed-length vector, as produced by CountVectorizer

Counts from Shakespeare plays. Figure 6.3, from Chapter 6, Speech and Language Processing, 3rd edition draft, Jurafsky & Martin (2019).

# Improving Document Vectors with TF-IDF

- The term-document matrix only uses **term frequency,** $tf(t, d)$:
  - Very frequent words like “the” and “it” carry little information…
  - But strong influence on the document vectors.
- TF-IDF emphasises words that occur in fewer documents by incorporating **inverse document frequency (IDF):**

$$idf(t, \boldsymbol{D}) = \log\left(\frac{|\boldsymbol{D}|}{number\ of\ documents\ \ in\ \boldsymbol{D}\ that\ t\ occurs\ in}\right)$$

$$tf \cdot idf(t, \boldsymbol{D}) = tf(t, d) \cdot idf(t, \boldsymbol{D})$$

# Term-Document Matrix with TF-IDF

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 0.074 | 0 | 0.22 | 0.28 |
| good | 0 | 0 | 0 | 0 |
| fool | 0.19 | 0.021 | 0.0036 | 0.083 |
| wit | 0.049 | 0.044 | 0.018 | 0.022 |

Word vector

Document vector

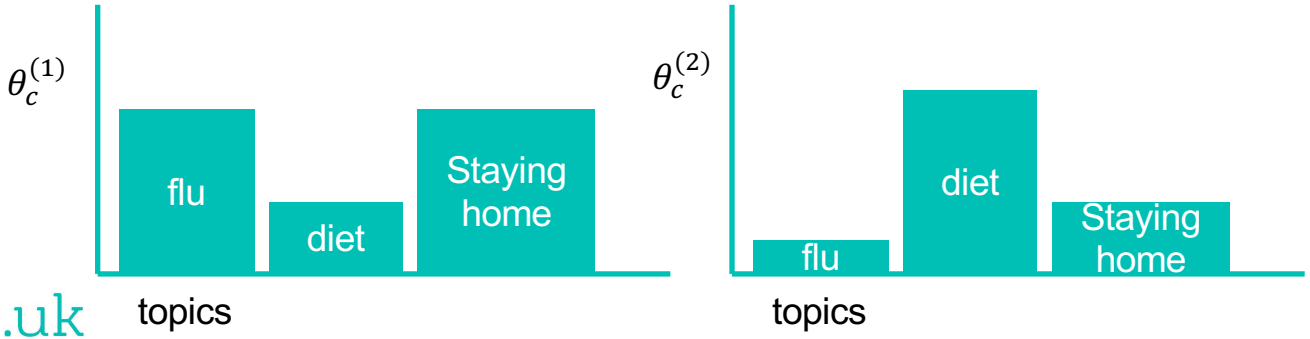
IDF for 'good' is zero as it occurs in all documents.

Figure 6.8, Chapter 6, Jurafsky & Martin (2019).

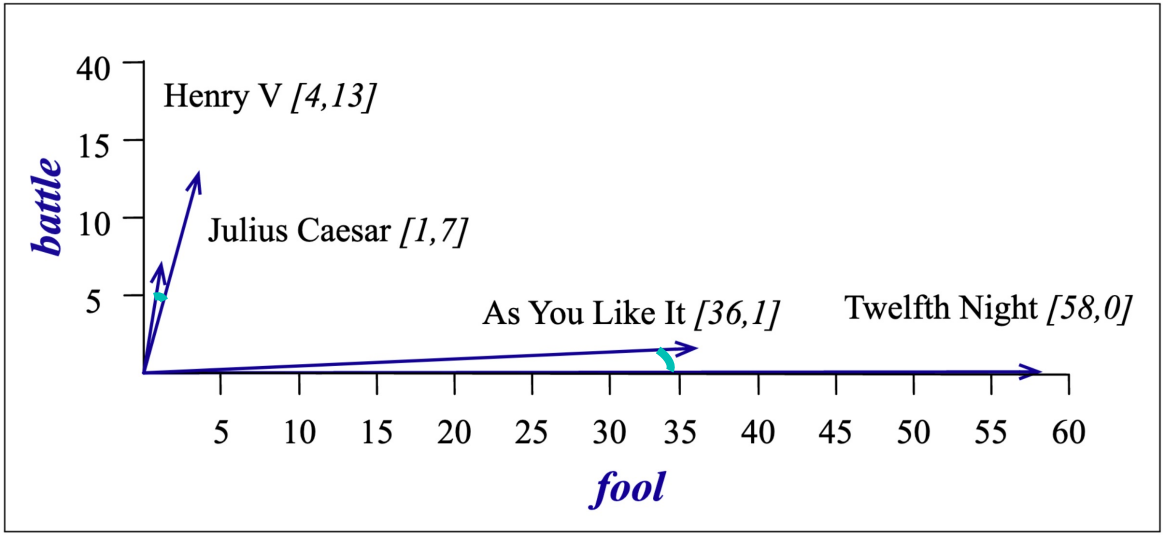
Here, $tf(t,d) = \log(count + 1)$ is used to 'squash' frequencies so small differences have less effect

# Dense Vector Representations

- Bag of words is a **sparse** representation: for any given document, most of the words have counts of zero!
- LDA outputs the expected topic distribution, $\mathbb{E}\left[\boldsymbol{\theta}^{(d)}\right]$, for document *d.*
- $\mathbb{E}\left[\boldsymbol{\theta}^{(d)}\right]$ is a dense vector representation of a document with no 0s.
- Can be treated as a feature vector



# Cosine Similarity




$$cosine(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}||\boldsymbol{b}|} = \frac{\sum_{i=1}^{N} a_i b_i}{\sqrt{\sum_{i=1}^{N} a_i^2} \sqrt{\sum_{i=1}^{N} b_i^2}}$$

- Words: paraphrases, related entities, tracking meaning changes.
- Documents: search, filtering, recommendation.

# Summary

- *Embeddings* are numerical vectors representing words, sentences or documents
- While the simplest embeddings are just term counts, TF-IDF can improve document embeddings
- Topic distributions provide a dense vector representation of a document that has some advantages as a feature vector for classification
- *Cosine similarity* is used to compare embeddings of different words or documents.