Algorithms for NLP

Vector Semantics

Yulia Tsvetkov – CMU

Slides: Dan Jurafsky – Stanford, David Bamman – UC Berkeley
Neural LMs

\[ i\text{-th output} = P(w_i = i \mid \text{context}) \]

Image: (Bengio et al, 03)
Neural LM Example

\[ h = \sigma (B^T C_x) \]

\[ P(y|x) \propto e^{(A^T h)} \]
## Neural LMs

<table>
<thead>
<tr>
<th>Model</th>
<th>n</th>
<th>c</th>
<th>h</th>
<th>m</th>
<th>direct</th>
<th>mix</th>
<th>train.</th>
<th>valid.</th>
<th>test.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP1</td>
<td>5</td>
<td></td>
<td>50</td>
<td>60</td>
<td>yes</td>
<td>no</td>
<td>182</td>
<td>284</td>
<td>268</td>
</tr>
<tr>
<td>MLP2</td>
<td>5</td>
<td></td>
<td>50</td>
<td>60</td>
<td>yes</td>
<td>yes</td>
<td>201</td>
<td>275</td>
<td>257</td>
</tr>
<tr>
<td>MLP3</td>
<td>5</td>
<td></td>
<td>0</td>
<td>60</td>
<td>yes</td>
<td>no</td>
<td>209</td>
<td>296</td>
<td>279</td>
</tr>
<tr>
<td>MLP4</td>
<td>5</td>
<td></td>
<td>0</td>
<td>60</td>
<td>yes</td>
<td>yes</td>
<td>210</td>
<td>309</td>
<td>293</td>
</tr>
<tr>
<td>MLP5</td>
<td>5</td>
<td></td>
<td>50</td>
<td>30</td>
<td>yes</td>
<td>no</td>
<td>296</td>
<td>273</td>
<td>259</td>
</tr>
<tr>
<td>MLP6</td>
<td>5</td>
<td></td>
<td>50</td>
<td>30</td>
<td>yes</td>
<td>yes</td>
<td>273</td>
<td>284</td>
<td>270</td>
</tr>
<tr>
<td>MLP7</td>
<td>3</td>
<td></td>
<td>50</td>
<td>30</td>
<td>yes</td>
<td>no</td>
<td>284</td>
<td>280</td>
<td>280</td>
</tr>
<tr>
<td>MLP8</td>
<td>3</td>
<td></td>
<td>50</td>
<td>30</td>
<td>yes</td>
<td>yes</td>
<td>284</td>
<td>276</td>
<td>276</td>
</tr>
<tr>
<td>MLP9</td>
<td>5</td>
<td></td>
<td>100</td>
<td>30</td>
<td>no</td>
<td>no</td>
<td>265</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td>MLP10</td>
<td>5</td>
<td></td>
<td>100</td>
<td>30</td>
<td>no</td>
<td>yes</td>
<td>265</td>
<td>352</td>
<td>336</td>
</tr>
<tr>
<td>Del. Int.</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31</td>
<td>334</td>
<td>323</td>
</tr>
<tr>
<td>Kneser-Ney back-off</td>
<td>3</td>
<td></td>
<td>50</td>
<td>30</td>
<td>yes</td>
<td>no</td>
<td>332</td>
<td>332</td>
<td>321</td>
</tr>
<tr>
<td>Kneser-Ney back-off</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>321</td>
<td>332</td>
<td>321</td>
</tr>
<tr>
<td>Kneser-Ney back-off</td>
<td>5</td>
<td></td>
<td>150</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>332</td>
<td>332</td>
<td>321</td>
</tr>
<tr>
<td>class-based back-off</td>
<td>3</td>
<td></td>
<td>200</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>348</td>
<td>348</td>
<td>334</td>
</tr>
<tr>
<td>class-based back-off</td>
<td>3</td>
<td></td>
<td>500</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>354</td>
<td>354</td>
<td>340</td>
</tr>
<tr>
<td>class-based back-off</td>
<td>3</td>
<td></td>
<td>1000</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>326</td>
<td>326</td>
<td>312</td>
</tr>
<tr>
<td>class-based back-off</td>
<td>4</td>
<td></td>
<td>2000</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>335</td>
<td>335</td>
<td>319</td>
</tr>
<tr>
<td>class-based back-off</td>
<td>5</td>
<td></td>
<td>500</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>343</td>
<td>343</td>
<td>326</td>
</tr>
<tr>
<td>class-based back-off</td>
<td>5</td>
<td></td>
<td>500</td>
<td></td>
<td>yes</td>
<td>no</td>
<td>327</td>
<td>327</td>
<td>312</td>
</tr>
</tbody>
</table>

Table 1: Comparative results on the Brown corpus. The deleted interpolation trigram has a test perplexity that is 33% above that of the neural network with the lowest validation perplexity.
Low-dimensional Representations

- Learning representations by back-propagating errors
  - Rumelhart, Hinton & Williams, 1986
- A neural probabilistic language model
  - Bengio et al., 2003
- Natural Language Processing (almost) from scratch
  - Collobert & Weston, 2008
- Word representations: A simple and general method for semi-supervised learning
  - Turian et al., 2010
- Distributed Representations of Words and Phrases and their Compositionality
  - Word2Vec; Mikolov et al., 2013
## Word Vectors

<table>
<thead>
<tr>
<th>WORD</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>...</th>
<th>d50</th>
</tr>
</thead>
<tbody>
<tr>
<td>summer</td>
<td>0.12</td>
<td>0.21</td>
<td>0.07</td>
<td>0.25</td>
<td>0.33</td>
<td>...</td>
<td>0.51</td>
</tr>
<tr>
<td>spring</td>
<td>0.19</td>
<td>0.57</td>
<td>0.99</td>
<td>0.30</td>
<td>0.02</td>
<td>...</td>
<td>0.73</td>
</tr>
<tr>
<td>fall</td>
<td>0.53</td>
<td>0.77</td>
<td>0.43</td>
<td>0.20</td>
<td>0.29</td>
<td>...</td>
<td>0.85</td>
</tr>
<tr>
<td>light</td>
<td>0.00</td>
<td>0.68</td>
<td>0.84</td>
<td>0.45</td>
<td>0.11</td>
<td>...</td>
<td>0.03</td>
</tr>
<tr>
<td>clear</td>
<td>0.27</td>
<td>0.50</td>
<td>0.21</td>
<td>0.56</td>
<td>0.25</td>
<td>...</td>
<td>0.32</td>
</tr>
<tr>
<td>blizzard</td>
<td>0.15</td>
<td>0.05</td>
<td>0.64</td>
<td>0.17</td>
<td>0.99</td>
<td>...</td>
<td>0.23</td>
</tr>
</tbody>
</table>
Lexical Semantics

- How should we represent the meaning of the word?
  - Words, lemmas, senses, definitions

---

**pepper, n.**

**Pronunciation:** [ˈpɪpər], U.S. /ˈpɪpɚ/.

**Etymology:** A borrowing from Latin. The modern Latin *piper*.

**Sense:**

1. The spice or the plant.
   a. A hot pungent spice derived from the prepared fruits (peppercorns) of the pepper plant, *Piper nigrum* (see sense 2), used from early times to season food, either whole or ground to powder (often in association with salt). Also (locally, chiefly, with distinguishing word): a similar spice derived from the fruits of certain other species of the genus *Piper*; the fruits themselves.

   The ground spice: *Piper nigrum* comes in two forms, the more pungent black pepper, procured from black peppercorns, and the mildest white pepper, procured from white peppercorns. See also n. 2, (b) Special cases 2a, 2b, 2c, 2d, 2e, and 2f. Special cases 2, (a) Special case 1a.

2. The plant *Piper nigrum* (family Piperaceae), a climbing shrub indigenous to South Asia and also cultivated elsewhere in the tropics, which has alternate pointed entire leaves, with pendulous spikes of small green flowers opposite the leaves, succeeded by small berries turning red when ripe. Also more widely: any plant of the genus *Piper* or the family Piperaceae.

3. (a) With distinguishing word: any of numerous plants of other families having hot pungent fruits or leaves which resemble pepper ( 1a) in taste and in some cases are used as a substitute for it.

---

http://www.oed.com/
Lemma pepper

- **Sense 1:**
  - spice from pepper plant

- **Sense 2:**
  - the pepper plant itself

- **Sense 3:**
  - another similar plant (Jamaican pepper)

- **Sense 4:**
  - another plant with peppercorns (California pepper)

- **Sense 5:**
  - capsicum (i.e. chili, paprika, bell pepper, etc)
A sense or “concept” is the meaning component of a word
Lexical Semantics

- How should we represent the meaning of the word?
  - Words, lemmas, senses, definitions
  - Relationships between words or senses
Relation: Synonymity

- Synonyms have the same meaning in some or all contexts.
  - filbert / hazelnut
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H20

- Note that there are probably no examples of perfect synonymy
  - Even if many aspects of meaning are identical
  - Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.
Relation: Antonymy

Senses that are opposites with respect to one feature of meaning

- Otherwise, they are very similar!
  - dark/light  short/long  fast/slow  rise/fall
  - hot/cold  up/down  in/out

More formally: antonyms can

- define a binary opposition or be at opposite ends of a scale
  - long/short, fast/slow
- be reversives:
  - rise/fall, up/down
Relation: Similarity

Words with similar meanings.

- Not synonyms, but sharing some element of meaning
  - car, bicycle
  - cow, horse
Ask humans how similar 2 words are

<table>
<thead>
<tr>
<th>word1</th>
<th>word2</th>
<th>similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanish</td>
<td>disappear</td>
<td>9.8</td>
</tr>
<tr>
<td>behave</td>
<td>obey</td>
<td>7.3</td>
</tr>
<tr>
<td>belief</td>
<td>impression</td>
<td>5.95</td>
</tr>
<tr>
<td>muscle</td>
<td>bone</td>
<td>3.65</td>
</tr>
<tr>
<td>modest</td>
<td>flexible</td>
<td>0.98</td>
</tr>
<tr>
<td>hole</td>
<td>agreement</td>
<td>0.3</td>
</tr>
</tbody>
</table>

SimLex-999 dataset (Hill et al., 2015)
Relation: Word relatedness

Also called "word association"

- Words be related in any way, perhaps via a semantic frame or field
  - car, bicycle: similar
  - car, gasoline: related, not similar
Semantic field

Words that

- cover a particular semantic domain
- bear structured relations with each other.

**hospitals**

- surgeon, scalpel, nurse, anaesthetic, hospital

**restaurants**

- waiter, menu, plate, food, menu, chef

**houses**

- door, roof, kitchen, family, bed
Relation: Superordinate/ Subordinate

- One sense is a subordinate of another if the first sense is more specific, denoting a subclass of the other
  - car is a subordinate of vehicle
  - mango is a subordinate of fruit
- Conversely superordinate
  - vehicle is a superordinate of car
  - fruit is a subordinate of mango
Taxonomy

Superordinate  Basic  Subordinate

furniture  chair  office chair  piano chair  rocking chair

lamp  torchiere  desk lamp

table  end table  coffee table
Lexical Semantics

- How should we represent the meaning of the word?
  - Words, lemmas, senses, definitions
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness
Lexical Semantics

How should we represent the meaning of the word?
- Dictionary definition
- Lemma and wordforms
- Senses
- Relationships between words or senses
- Taxonomic relationships
- Word similarity, word relatedness
- Semantic frames and roles
  - *John hit Bill*
  - *Bill was hit by John*
Lexical Semantics

How should we represent the meaning of the word?

- Dictionary definition
- Lemma and wordforms
- Senses
- Relationships between words or senses
- Taxonomic relationships
- Word similarity, word relatedness
- Semantic frames and roles
- Connotation and sentiment

- *valence*: the pleasantness of the stimulus
- *arousal*: the intensity of emotion
- *dominance*: the degree of control exerted by the stimulus

<table>
<thead>
<tr>
<th></th>
<th>Valence</th>
<th>Arousal</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>courageous</td>
<td>8.05</td>
<td>5.5</td>
<td>7.38</td>
</tr>
<tr>
<td>music</td>
<td>7.67</td>
<td>5.57</td>
<td>6.5</td>
</tr>
<tr>
<td>heartbreak</td>
<td>2.45</td>
<td>5.65</td>
<td>3.58</td>
</tr>
<tr>
<td>cub</td>
<td>6.71</td>
<td>3.95</td>
<td>4.24</td>
</tr>
<tr>
<td>life</td>
<td>6.68</td>
<td>5.59</td>
<td>5.89</td>
</tr>
</tbody>
</table>
Electronic Dictionaries

WordNet

```python
from nltk.corpus import wordnet as wn
panda = wn.synset('panda.n.01')
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```

(here, for good):

```
[S: (adj) full, good
S: (adj) estimable, good, honorable, respectable
S: (adj) beneficial, good
S: (adj) good, just, upright
S: (adj) adept, expert, good, practiced,
proficient, skillful
S: (adj) dear, good, near
S: (adj) good, right, ripe
... 
S: (adv) well, good
S: (adv) thoroughly, soundly, good
S: (n) good, goodness
S: (n) commodity, trade good, good
```
Problems with Discrete Representations

- Too coarse
  - expert ↔ skillful
- Sparse
  - wicked, badass, ninja
- Subjective
- Expensive
- Hard to compute word relationships

\[
\begin{align*}
\text{expert} & \quad [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\
\text{skillful} & \quad [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]
\end{align*}
\]
“The meaning of a word is its use in the language”

[Wittgenstein PI 43]

“You shall know a word by the company it keeps”

[Firth 1957]

If A and B have almost identical environments we say that they are synonyms.

[Harris 1954]
Example

What does ongchoi mean?

- Suppose you see these sentences:
  - Ongchoi is delicious *sautéed with garlic*.
  - Ongchoi is superb *over rice*
  - Ongchoi *leaves* with salty sauces

- And you've also seen these:
  - …spinach *sautéed with garlic over rice*
  - Chard stems and leaves are delicious
  - Collard greens and other *salty* leafy greens

Conclusion:
- Ongchoi is a leafy green like spinach, chard, or collard greens
Ongchoi: Ipomoea aquatica "Water Spinach"

Yamaguchi, Wikimedia Commons, public domain
- Each word = a vector
  - not just “word” or word45.
  - similar words are “nearby in space”
  - the standard way to represent meaning in NLP
We'll Introduce 3 Kinds of Embeddings

- **Count-based**
  - Words are represented by a simple function of the counts of nearby words

- **Brown clusters**
  - Representation is created through hierarchical clustering

- **Word2Vec**
  - Representation is created by training a classifier to distinguish nearby and far-away words

Next class:

- Fasttext
- ELMO
- Multilingual embeddings
## Term-Document Matrix

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>good</td>
<td>114</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>wit</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Context = appearing in the same document.
## Term-Document Matrix

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>good</td>
<td>114</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>wit</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Each document is represented by a vector of words
Vectors are the Basis of Information Retrieval

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>good</td>
<td>114</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>wit</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- Vectors are similar for the two comedies
- Different than the history
- Comedies have more fools and wit and fewer battles.
Visualizing Document Vectors

- Henry V [4,13]
- Julius Caesar [1,7]
- As You Like It [36,1]
- Twelfth Night [58,0]
### Words Can Be Vectors Too

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>good</td>
<td>114</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>wit</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- battle is "the kind of word that occurs in Julius Caesar and Henry V"
- fool is "the kind of word that occurs in comedies, especially Twelfth Night"
## Term-Context Matrix

<table>
<thead>
<tr>
<th></th>
<th>knife</th>
<th>dog</th>
<th>sword</th>
<th>love</th>
<th>like</th>
</tr>
</thead>
<tbody>
<tr>
<td>knife</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>sword</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>love</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>like</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

- Two words are “similar” in meaning if their context vectors are similar
  - Similarity == relatedness
### Count-Based Representations

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>good</td>
<td>114</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>wit</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- **Counts: term-frequency**
  - remove stop words
  - use $\log_{10}(tf)$
  - normalize by document length
What to do with words that are evenly distributed across many documents?

\[
tf_{t,d} = \begin{cases} 
1 + \log_{10} \text{count}(t,d) & \text{if } \text{count}(t,d) > 0 \\
0 & \text{otherwise}
\end{cases}
\]

\[
idf_i = \log \left( \frac{N}{df_i} \right)
\]

Words like "the" or "good" have very low idf

\[
w_{t,d} = tf_{t,d} \times idf_t
\]
In word-context matrix

Do words $w$ and $c$ co-occur more than if they were independent?

$$\text{PPMI}_\alpha(w, c) = \max(\log_2 \frac{P(w, c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{\text{count}(c)^\alpha}{\sum_c \text{count}(c)^\alpha}$$

PMI is biased toward infrequent events

- Very rare words have very high PMI values
- Give rare words slightly higher probabilities $\alpha=0.75$
Dimensionality Reduction

  - High dimensionality of word--document matrix
  - Sparsity
  - The order of rows and columns doesn’t matter
- Goal:
  - good similarity measure for words or documents
  - dense representation
- Sparse vs Dense vectors
  - Short vectors may be easier to use as features in machine learning (less weights to tune)
  - Dense vectors may generalize better than storing explicit counts
  - They may do better at capturing synonymy
  - In practice, they work better
Singular Value Decomposition (SVD)

- Solution idea:
  - Find a projection into a low-dimensional space (~300 dim)
  - That gives us a best separation between features
Truncated SVD

We can approximate the full matrix by only considering the leftmost $k$ terms in the diagonal matrix (the $k$ largest singular values).

$$A_{m \times n} \approx U_{m \times k} \Sigma_{k \times k} V_{k \times n}^T \quad \text{with} \quad k \ll m, n$$
## Latent Semantic Analysis

<table>
<thead>
<tr>
<th>#0</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
</tr>
</thead>
<tbody>
<tr>
<td>we</td>
<td>music</td>
<td>company</td>
<td>how</td>
<td>program</td>
<td>10</td>
</tr>
<tr>
<td>said</td>
<td>film</td>
<td>mr</td>
<td>what</td>
<td>project</td>
<td>30</td>
</tr>
<tr>
<td>have</td>
<td>theater</td>
<td>its</td>
<td>about</td>
<td>russian</td>
<td>11</td>
</tr>
<tr>
<td>they</td>
<td>mr</td>
<td>inc</td>
<td>their</td>
<td>space</td>
<td>12</td>
</tr>
<tr>
<td>not</td>
<td>this</td>
<td>stock</td>
<td>or</td>
<td>russia</td>
<td>15</td>
</tr>
<tr>
<td>but</td>
<td>who</td>
<td>companies</td>
<td>this</td>
<td>center</td>
<td>13</td>
</tr>
<tr>
<td>be</td>
<td>movie</td>
<td>sales</td>
<td>are</td>
<td>programs</td>
<td>14</td>
</tr>
<tr>
<td>do</td>
<td>which</td>
<td>shares</td>
<td>history</td>
<td>clark</td>
<td>20</td>
</tr>
<tr>
<td>he</td>
<td>show</td>
<td>said</td>
<td>be</td>
<td>aircraft</td>
<td>sept</td>
</tr>
<tr>
<td>this</td>
<td>about</td>
<td>business</td>
<td>social</td>
<td>ballet</td>
<td>16</td>
</tr>
<tr>
<td>there</td>
<td>dance</td>
<td>share</td>
<td>these</td>
<td>its</td>
<td>25</td>
</tr>
<tr>
<td>you</td>
<td>its</td>
<td>chief</td>
<td>other</td>
<td>projects</td>
<td>17</td>
</tr>
<tr>
<td>are</td>
<td>disney</td>
<td>executive</td>
<td>research</td>
<td>orchestra</td>
<td>18</td>
</tr>
<tr>
<td>what</td>
<td>play</td>
<td>president</td>
<td>writes</td>
<td>development</td>
<td>19</td>
</tr>
<tr>
<td>if</td>
<td>production</td>
<td>group</td>
<td>language</td>
<td>work</td>
<td>21</td>
</tr>
</tbody>
</table>

[Deerwester et al., 1990]
LSA++

- Probabilistic Latent Semantic Indexing (PLSI)
  - Hofmann, 1999
- Latent Dirichlet Allocation (LDA)
  - Blei et al., 2003
- Nonnegative Matrix Factorization (NMF)
  - Lee & Seung, 1999
Word Similarity

\[
\text{cosine}(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}
\]
## Evaluation

<table>
<thead>
<tr>
<th>WORD</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>...</th>
<th>d50</th>
</tr>
</thead>
<tbody>
<tr>
<td>summer</td>
<td>0.12</td>
<td>0.21</td>
<td>0.07</td>
<td>0.25</td>
<td>0.33</td>
<td>...</td>
<td>0.51</td>
</tr>
<tr>
<td>spring</td>
<td>0.19</td>
<td>0.57</td>
<td>0.99</td>
<td>0.30</td>
<td>0.02</td>
<td>...</td>
<td>0.73</td>
</tr>
<tr>
<td>fall</td>
<td>0.53</td>
<td>0.77</td>
<td>0.43</td>
<td>0.20</td>
<td>0.29</td>
<td>...</td>
<td>0.85</td>
</tr>
<tr>
<td>light</td>
<td>0.00</td>
<td>0.68</td>
<td>0.84</td>
<td>0.45</td>
<td>0.11</td>
<td>...</td>
<td>0.03</td>
</tr>
<tr>
<td>clear</td>
<td>0.27</td>
<td>0.50</td>
<td>0.21</td>
<td>0.56</td>
<td>0.25</td>
<td>...</td>
<td>0.32</td>
</tr>
<tr>
<td>blizzard</td>
<td>0.15</td>
<td>0.05</td>
<td>0.64</td>
<td>0.17</td>
<td>0.99</td>
<td>...</td>
<td>0.23</td>
</tr>
</tbody>
</table>

- Intrinsic
- Extrinsic
- Qualitative
### Intrinsic Evaluation

<table>
<thead>
<tr>
<th>word1</th>
<th>word2</th>
<th>similarity (humans)</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanish</td>
<td>disappear</td>
<td>9.8</td>
</tr>
<tr>
<td>behave</td>
<td>obey</td>
<td>7.3</td>
</tr>
<tr>
<td>belief</td>
<td>impression</td>
<td>5.95</td>
</tr>
<tr>
<td>muscle</td>
<td>bone</td>
<td>3.65</td>
</tr>
<tr>
<td>modest</td>
<td>flexible</td>
<td>0.98</td>
</tr>
<tr>
<td>hole</td>
<td>agreement</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>similarity (embeddings)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>1.7</td>
</tr>
<tr>
<td>0.98</td>
</tr>
<tr>
<td>0.3</td>
</tr>
</tbody>
</table>

- WS-353 (Finkelstein et al. ‘02)
- MEN-3k (Bruni et al. ‘12)
- SimLex-999 dataset (Hill et al., 2015)
Extrinsic Evaluation

- Chunking
- POS tagging
- Parsing
- MT
- SRL
- Topic categorization
- Sentiment analysis
- Metaphor detection
- etc.
Visualisation

- Visualizing Data using t-SNE (van der Maaten & Hinton’08)

[Figure 6.5: Monolingual (top) and multilingual (bottom; marked with apostrophe) word projections of the antonyms (shown in red) and synonyms of “beautiful”.]

[Faruqui et al., 2015]
vector(‘king’) - vector(‘man’) + vector(‘woman’) ≈ vector(‘queen’)
vector(‘Paris’) - vector(‘France’) + vector(‘Italy’) ≈ vector(‘Rome’)

\[
\min \cos(\text{man} - \text{woman}, \text{king} - x) \text{ s.t. } \|\text{king} - x\|_2 < \delta
\]

[Mikolov et al.’13]
and also human biases

$$\min \cos(he - she, x - y) \; s.t. \; \|x - y\|_2 < \delta$$

<table>
<thead>
<tr>
<th>Extreme she</th>
<th>Extreme he</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. homemaker</td>
<td>1. maestro</td>
</tr>
<tr>
<td>2. nurse</td>
<td>2. skipper</td>
</tr>
<tr>
<td>3. receptionist</td>
<td>3. protege</td>
</tr>
<tr>
<td>4. librarian</td>
<td>4. philosopher</td>
</tr>
<tr>
<td>5. socialite</td>
<td>5. captain</td>
</tr>
<tr>
<td>6. hairdresser</td>
<td>6. architect</td>
</tr>
<tr>
<td>7. nanny</td>
<td>7. financier</td>
</tr>
<tr>
<td>8. bookkeeper</td>
<td>8. warrior</td>
</tr>
<tr>
<td>9. stylist</td>
<td>9. broadcaster</td>
</tr>
<tr>
<td>10. housekeeper</td>
<td>10. magician</td>
</tr>
</tbody>
</table>

**Gender stereotype she-he analogies**
- sewing-carpentry
- nurse-surgeon
- blond-burly
- giggle-chuckle
- sassy-snappy
- volleyball-football

- registered nurse-physician
- interior designer-architect
- feminism-conservatism
- vocalist-guitarist
- diva-superstar
- cupcakes-pizzas

- housewife-shopkeeper
- softball-baseball
- cosmetics-pharmaceuticals
- petite-lanky
- charming-affable
- lovely-brilliant

**Gender appropriate she-he analogies**
- queen-king
- waitress-waiter
- ovarian cancer-prostate cancer convent-monastery

Figure 1: **Left** The most extreme occupations as projected on to the she—he gender direction on w2vNEWS. Occupations such as businesswoman, where gender is suggested by the orthography, were excluded. **Right** Automatically generated analogies for the pair she-he using the procedure described in text. Each automatically generated analogy is evaluated by 10 crowd-workers to whether or not it reflects gender stereotype.

[Bolukbasi et al., ‘16]
What we’ve seen by now

- Meaning representation
- Distributional hypothesis
- Count-based vectors
  - term-document matrix
  - word-in-context matrix
  - normalizing counts: tf-idf, PPMI
  - dimensionality reduction
  - measuring similarity
  - evaluation

Next:

- Brown clusters
  - Representation is created through hierarchical clustering
Brown Clustering

dog [0000]
cat [0001]
ant [001]
river [010]
lake [011]
blue [10]
red [11]
Brown Clustering

Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays
June March July April January December October November September August
people guys folks fellows CEOs doubters commies unfortunates blokes
down backwards ashore sideways southward northward overboard aloft downwards adrift
water gas coal liquid acid sand carbon steam shale iron
great big vast sudden mere sheer gigantic lifelong scant colossal
man woman boy girl lawyer doctor guy farmer teacher citizen
American Indian European Japanese German African Catholic Israeli Italian Arab
pressure temperature permeability density porosity stress velocity viscosity gravity tension
mother wife father son husband brother daughter sister boss uncle
machine device controller processor CPU printer spindle subsystem compiler plotter
John George James Bob Robert Paul William Jim David Mike
anyone someone anybody somebody
feet miles pounds degrees inches barrels tons acres meters bytes
director chief professor commissioner commander treasurer founder superintendent dean cus-
todian
liberal conservative parliamentary royal progressive Tory provisional separatist federalist PQ
had hadn’t hath would’ve could’ve should’ve must’ve might’ve
asking telling wondering instructing informing kidding reminding bothering thanking deposing
that tha theat
head body hands eves voice arm seat eve hair mouth

[Brown et al, 1992]
Brown Clustering

<table>
<thead>
<tr>
<th>Term</th>
<th>Bit String</th>
</tr>
</thead>
<tbody>
<tr>
<td>lawyer</td>
<td>1000001101000</td>
</tr>
<tr>
<td>newspaperman</td>
<td>1000001101001</td>
</tr>
<tr>
<td>stewardess</td>
<td>10000011010101</td>
</tr>
<tr>
<td>toxicologist</td>
<td>1000001101011</td>
</tr>
<tr>
<td>slang</td>
<td>1000001101010</td>
</tr>
<tr>
<td>babysitter</td>
<td>1000001101100</td>
</tr>
<tr>
<td>conspirator</td>
<td>10000011011010</td>
</tr>
<tr>
<td>womanizer</td>
<td>10000011011011</td>
</tr>
<tr>
<td>mailman</td>
<td>1000001101111</td>
</tr>
<tr>
<td>salesman</td>
<td>1000001110000</td>
</tr>
<tr>
<td>bookkeeper</td>
<td>10000011100010</td>
</tr>
<tr>
<td>troubleshooter</td>
<td>10000011100011</td>
</tr>
<tr>
<td>bouncer</td>
<td>100000111000111</td>
</tr>
<tr>
<td>technician</td>
<td>1000001110100</td>
</tr>
<tr>
<td>janitor</td>
<td>1000001110101</td>
</tr>
<tr>
<td>saleswoman</td>
<td>1000001110110</td>
</tr>
</tbody>
</table>

| ...          | 10110111001010111100 |
| Nike         | 10110111001010111010 |
| Maytag       | 10110111001010111111 |
| Generali     | 10110111001010111111 |
| Gap          | 10110111010010111111 |
| Harley-Davidson | 10110111010101111111 |
| Enfield      | 10110111010101111111 |
| genus        | 10110111010101111111 |
| Microsoft    | 10110111010101110000 |
| Ventrix      | 10110111010101110000 |
| Traciebel    | 10110111010101110000 |
| Synopsisys   | 10110111010101110000 |
| WordPerfect  | 10110111010101110000 |

| ...          | 101110010000000000 |
| John         | 101110010000000000 |
| Consuelo     | 101110010000000000 |
| Jeffrey      | 101110010000000000 |
| Kenneth      | 101110010000000000 |
| Phillip      | 101110010000000000 |
| WILLIAM      | 101110010000000000 |
| Timothy      | 101110010000000000 |
| Terrence     | 101110010000000000 |
| Jerald       | 101110010000000000 |
| Harold       | 101110010000000000 |
| Frederic     | 101110010000000000 |
| Wendell      | 101110010000000000 |

Table 1: Sample bit strings

[Miller et al., 2004]
Brown Clustering

- $\mathcal{V}$ is a vocabulary

- $C: \mathcal{V} \to \{1, 2, \ldots, k\}$ is a partition of the vocabulary into $k$ clusters

- $p(C(w_i)|C(w_{i-1}))$ is a probability of cluster of $w_i$ to follow the cluster of $w_{i-1}$

- $p(w_i|C(w_i)) = \frac{\text{count}(w_i)}{\sum_{x \in C(w_i)} \text{count}(x)}$

The model:

$$\text{Quality}(C) = \prod_{i=1}^{n} p(w_i|C(w_i))p(C(w_i)|C(w_{i-1}))$$
Next

- How do we measure the quality of a partition $\text{Quality}(C)$?
- How to cluster?
Quality(C) = \frac{1}{n} \sum_{i=1}^{n} \log P(C(w_i)|C(w_{i-1}))P(w_i|C(w_i))

= \sum_{w,w'} \frac{n(w,w')}{n} \log \frac{P(C'(w')|C(w))P(w'|C(w'))}{\frac{n(C'(w')C(w))}{n(C(w))} \frac{n(w')}{n(C'(w'))}}

= \sum_{w,w'} \frac{n(w,w')}{n} \log \frac{n(C'(w'),C(w'))\frac{n(w')}{n(C'(w'))}}{n(C(w))n(C(w'))} + \sum_{w,w'} \frac{n(w,w')}{n} \log \frac{n(w')}{n}

= \sum_{c,c'} \frac{n(c,c')}{n} \log \frac{n(c,c')\frac{n}{n(c)n(c')}}{n(c)n(c')} + \sum_{w'} \frac{n(w')}{n} \log \frac{n(w')}{n}
Quality(C)

Define

\[ P(c, c') = \frac{n(c, c')}{n} \quad P(w) = \frac{n(w)}{n} \quad P(c) = \frac{n(c)}{n} \]

Then (again from Percy Liang, 2005):

\[
\text{Quality}(C) = \sum_{c, c'} P(c, c') \log \frac{P(c, c')}{P(c)P(c')} + \sum_w P(w) \log P(w)
\]

\[ = I(C) - H \]

The first term \(I(C)\) is the mutual information between adjacent clusters and the second term \(H\) is the entropy of the word distribution. Note that the quality of \(C\) can be computed as a sum of mutual information weights between clusters minus the constant \(H\), which does not depend on \(C\). This decomposition allows us to make optimizations.
A Naive Algorithm

- We start with $|V|$ clusters: each word gets its own cluster
- Our aim is to find $k$ final clusters
- We run $|V| - k$ merge steps:
  - At each merge step we pick two clusters $c_i$ and $c_j$, and merge them into a single cluster
  - We greedily pick merges such that Quality$(C)$ for the clustering $C$ after the merge step is maximized at each stage
- Cost? Naive = $O(|V|^5)$. Improved algorithm gives $O(|V|^3)$: still too slow for realistic values of $|V|$
Brown Clustering Algorithm

- Parameter of the approach is $m$ (e.g., $m = 1000$)
- Take the top $m$ most frequent words, put each into its own cluster, $c_1, c_2, \ldots, c_m$
- For $i = (m + 1) \ldots |\mathcal{V}|$
  - Create a new cluster, $c_{m+1}$, for the $i$’th most frequent word. We now have $m + 1$ clusters
  - Choose two clusters from $c_1 \ldots c_{m+1}$ to be merged: pick the merge that gives a maximum value for $\text{Quality}(C)$. We’re now back to $m$ clusters

- Carry out $(m - 1)$ final merges, to create a full hierarchy

- Running time: $O(|\mathcal{V}|m^2 + n)$ where $n$ is corpus length
## Part-of-Speech Tagging for Twitter

<table>
<thead>
<tr>
<th>Binary path</th>
<th>Top words (by frequency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 111010100010</td>
<td>lmao lmfao lmaoo imaoog imaooc imaooo hahahahah laol lmfao lmfao lmfao lmfao lmfao lmaoo0 lmaooo lma000 lmo lolo lolo</td>
</tr>
<tr>
<td>A2 111010100011</td>
<td>haha hahah hehe hahahahahah aha hehehe ahah hahahahahahkk hahaaa ahah</td>
</tr>
<tr>
<td>A3 111010100100</td>
<td>yes yep yup nope yess yessss yessssss ofcourse yeap likewise yepp yesh yw yuup yus</td>
</tr>
<tr>
<td>A4 111010100101</td>
<td>yesh yea nah naw yewah noah yew noo nooo yeesa ikr nvm yeahh nahn noooo0</td>
</tr>
<tr>
<td>A5 11101011011100</td>
<td>smh jk #fail #random #fact smfh #smh #winning #realtalk smdh #dead #justsaying</td>
</tr>
<tr>
<td>B 011101011</td>
<td>yu yuh yhu yu uu yuu yoh yuhh yohh yhhuu iget yoy yoooh yuoo yyy yoe jye juu D dyay ouzz ouyouou</td>
</tr>
<tr>
<td>C 11100101111011</td>
<td>w fo fa fr fro ov fer fir whit abou aft serie fohu fuh w/her w/that fron isn again</td>
</tr>
<tr>
<td>D 11110101110000</td>
<td>facebook fb itunes myspace skype twitter tumblr bbm flickr aim msn netflix pandora</td>
</tr>
<tr>
<td>E1 0011001</td>
<td>tryna gonn finna boutta trynna boutta gne fina gonn tryina fenna qone trynaa qon</td>
</tr>
<tr>
<td>E2 0011000</td>
<td>gonna gunna gona gna guna gnaa ganna gonnna ganna gunna gonne goona</td>
</tr>
<tr>
<td>F 0110110111</td>
<td>soo soo soooo sooooo sooooooooo soooooooooo sooooooooooo soooooooooooo sooooooooooo</td>
</tr>
<tr>
<td>G1 11101011000100</td>
<td>;:] :p :d :xd ::o :d (; :3:p :=p :p =) ;x :gno xddd :&gt; ;p &gt;:d 8-) ;d</td>
</tr>
<tr>
<td>G2 11101011000111</td>
<td>:o (: =) :) 8 ) @ :) =] ^<em><code>) (^ [ :) ) ( (: ^_</code>) (= ^</em>`))</td>
</tr>
<tr>
<td>G3 1110101100111</td>
<td>:/ _` _ -- (: (? d : s : s -- -- -- ( = &lt; &gt; -- &gt; ) -- ) &lt;/3 \ -- -- - ; ( / : ( = &lt; &gt; = [ :) #fml</td>
</tr>
<tr>
<td>G4 1110101100011</td>
<td>&lt;3 &lt;3 xoxo &lt;33 xo &lt;333 &lt;3 &lt;3 &lt;3 &lt;3 love s2 &lt;URL-twitition.com&gt; #neversaynever &lt;3333</td>
</tr>
</tbody>
</table>

Figure 2: Example word clusters (HMM classes): we list the most probable words, starting with the most probable, in descending order. Boldfaced words appear in the example tweet (Figure 1). The binary strings are root-to-leaf paths through the binary cluster tree. For example usage, see e.g. search.twitter.com, bing.com/social and urbandictionary.com.

[Owoputi et al., 2013]
Plan for Today

- **Count-based**
  - Words are represented by a simple function of the counts of nearby words
- **Brown clusters**
  - Representation is created through hierarchical clustering
- **Word2Vec**
  - Representation is created by training a classifier to distinguish nearby and far-away words

Next class:
- Fasttext
- ELMo
- Multilingual embeddings
Dense Embeddings You Can Download

**Word2vec** (Mikolov et al.’ 13)
https://code.google.com/archive/p/word2vec/

**Fasttext** (Bojanowski et al.’ 17)
http://www.fasttext.cc/

**Glove** (Pennington et al., 14)
http://nlp.stanford.edu/projects/glove/
Word2Vec

- Popular embedding method
- Very fast to train
- Code available on the web
- Idea: predict rather than count
Instead of counting how often each word w occurs near "apricot"
  ▪ Train a classifier on a binary prediction task:
  ▪ Is w likely to show up near "apricot"?
Word2Vec

Skip-gram

CBOV

[Mikolov et al.’ 13]
Next Class

- Word2Vec
- Fasttext
- ELMo
- Multilingual embeddings