Outline

- Vector Semantics
  - Sparse representation
  - Pointwise Mutual Information (PMI)
- Dense representation
  - Singular Value Decomposition (SVD)
  - Neural Language Model (Word2Vec)

Why vector models of meaning?

Computing the similarity between words

“fast” is similar to “rapid”
“tall” is similar to “height”

Question answering:
Q: “How tall is Mt. Everest?”
Candidate A: “The official height of Mount Everest is 29029 feet”

Automatically constructed clusters of semantically similar words (Charniak, 1997):

<table>
<thead>
<tr>
<th>Friday</th>
<th>Monday</th>
<th>Thursday</th>
<th>Wednesday</th>
<th>Tuesday</th>
<th>Saturday</th>
<th>Sunday</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>guys</td>
<td>folks</td>
<td>CEOs</td>
<td>consultants</td>
<td>blocks</td>
<td></td>
</tr>
<tr>
<td>water</td>
<td>gola</td>
<td>liquid</td>
<td>acid</td>
<td>carbon</td>
<td>shales</td>
<td></td>
</tr>
<tr>
<td>that</td>
<td>the</td>
<td>she</td>
<td>she</td>
<td>body</td>
<td>hands</td>
<td></td>
</tr>
<tr>
<td>head</td>
<td>body</td>
<td>hands</td>
<td>eyes</td>
<td>voice</td>
<td>arm</td>
<td></td>
</tr>
<tr>
<td>sent</td>
<td>eye</td>
<td>hair</td>
<td>mouth</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Smoothing for statistical language models

- Two alternative guesses of speech recognizer:
  For breakfast, she ate durian.
  For breakfast, she ate Dorian.

- Our corpus contains neither “ate durian” nor “ate Dorian”

- But, our corpus contains “ate orange”, “ate banana”

Distributional models of meaning

= vector-space models of meaning
= vector semantics

Intuitions:
Zellig Harris (1954):
- “oculist and eye-doctor … occur in almost the same environments”
- “If A and B have almost identical environments we say that they are synonyms.”

Firth (1957):
- “You shall know a word by the company it keeps!”
Intuition of distributional word similarity

- Example:
  - What is *tesgüino*?

- From context words humans can guess *tesgüino* means
  - an alcoholic beverage like beer
  - Intuition for algorithm:
    - Two words are similar if they have similar word contexts.

Four kinds of vector models

Sparse vector representations
1. Mutual-information weighted word co-occurrence matrices

Dense vector representations:
2. Singular value decomposition (and Latent Semantic Analysis)
3. Neural-network-inspired models (skip-grams, CBOW)
4. Brown clusters

Sample Lexical Vector Space
Term-context matrix for word similarity

The words in a term-document matrix

Term-document matrix

The words in a term-document matrix

The word-word matrix
**Positive Pointwise Mutual Information (PPMI)**

- Raw word frequency is not a great measure of association between words.
  - It's very skewed.
  - "the" and "a" are very frequent, but maybe not the most discriminative.

---

**Problem with raw counts**

- We showed only 4x6, but the real matrix is 50,000 x 50,000.
  - So it's very sparse.
  - Most values are 0.
  - That's OK, since there are lots of efficient algorithms for sparse matrices.

**Word-word matrix**

- The size of windows depends on your goals.
  - The shorter the window, the more syntactic the representation.
  - The longer the window, the more semantic the representation.

- Raw word frequency is not a great measure of association between words.
  - It's very skewed.
  - "the" and "a" are very frequent, but maybe not the most discriminative.

---

**Sample contexts ± 7 words**

- sugar, a sliced lemon, a tablespoonful of sugar, preserving jars, preserve or jam, a pinch each of sugar and another fruit whose name she liked.

---

**Example of a term-term matrix**

<table>
<thead>
<tr>
<th></th>
<th>apricots</th>
<th>pineapple</th>
<th>computer</th>
<th>data</th>
<th>pinch</th>
<th>result</th>
<th>sugar</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricots</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>pineapple</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>data</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pinch</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>sugar</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Problem with raw counts

• Raw word frequency is not a great measure of association between words
  • It’s very skewed
  • “the” and “of” are very frequent, but maybe not the most discriminative
  • We’d rather have a measure that asks whether a context word is particularly informative about the target word.

• Positive Pointwise Mutual Information (PPMI)

Pointwise Mutual Information

Pointwise mutual information:
Do events x and y co-occur more than if they were independent?
\[ \text{PMI}(X,Y) = \log_2 \frac{p(x,y)}{p(x)p(y)} \]

PMI between two words: (Church & Hanks 1989)
Do words x and y co-occur more than if they were independent?
\[ \text{PMI}(\text{word}_i, \text{word}_j) = \log_2 \frac{p(\text{word}_i, \text{word}_j)}{p(\text{word}_i)p(\text{word}_j)} \]

Positive Pointwise Mutual Information

• PMI ranges from $-\infty$ to $+\infty$
• But the negative values are problematic
  • Things are co-occurring less than we expect by chance
  • Unreliable without enormous corpora
    • Imagine w1 and w2 whose probability is each $10^{-6}$
      • Hard to be sure $p(\text{w}_1, \text{w}_2)$ is significantly different than $10^{-12}$
    • Plus it’s not clear people are good at “unrelatedness”
• So we just replace negative PMI values by 0
• Positive PMI (PPMI) between word1 and word2:
  \[ \text{PPMI}(\text{word}_i, \text{word}_j) = \max \left( \log_2 \frac{p(\text{word}_i, \text{word}_j)}{p(\text{word}_i)p(\text{word}_j)}, 0 \right) \]

Computing PPMI on a term-context matrix

• Matrix F with W rows (words) and C columns (contexts, e.g. in the form of words)
• $f_{ij}$ is number of times $w_i$ occurs in context $c_j$
• Imagine $w_1$ and $w_2$ whose probability is each $10^{-6}$
• Hard to be sure $p(\text{w}_1, \text{w}_2)$ is significantly different than $10^{-12}$
• Plus it’s not clear people are good at “unrelatedness”
• So we just replace negative PMI values by 0
• Positive PMI (PPMI) between word1 and word2:
  \[ \text{PPMI}(\text{word}_i, \text{word}_j) = \max \left( \log_2 \frac{p(\text{word}_i, \text{word}_j)}{p(\text{word}_i)p(\text{word}_j)}, 0 \right) \]

\[ p_{ij} = \frac{f_{ij}}{N} \]
\[ p_{ri} = \frac{\sum f_{ri}}{N} \]
\[ p_{p} = \frac{\sum f_{p}}{N} \]
\[ p_{mj} = \log_2 \frac{p_{mj}}{p_{mj}^{*}} \]
\[ \text{ppmi}_j = \begin{cases} \text{ppmi}_j & \text{if } \text{ppmi}_j > 0 \\ 0 & \text{otherwise} \end{cases} \]
Weighting PMI

- PMI is biased toward infrequent events
- Very rare words have very high PMI values
- Two solutions:
  - Give rare words slightly higher probabilities
  - Use add-k smoothing (which has a similar effect)

Add-k smoothing

<table>
<thead>
<tr>
<th>Count(w,context)</th>
<th>computer</th>
<th>data</th>
<th>pinch</th>
<th>result</th>
<th>sugar</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricot</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>pineapple</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>digital</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>information</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ p_{ij} = \frac{f_{ij}}{\sum_{j} f_{ij}} \]

\[ p(w) = \sum_{j} p(w|j) \]

\[ p(w,context) = \sum_{j} p(w|j)p(j) \]

Adding smoothing:

1. PMI is biased toward infrequent events
2. Very rare words have very high PMI values
3. Two solutions:
   1. Give rare words slightly higher probabilities
   2. Use add-k smoothing (which has a similar effect)

Weighting PMI: Giving rare context words slightly higher probability

- Raise the context probabilities to \( \alpha = 0.75 \):
  \[ \text{PPMI}(w,c) = \max(\log_{\alpha} \frac{p(w,c)}{p(w)p(c)}) \]

\[ P_{\alpha}(c) = \frac{\text{count}(c)^\alpha}{\sum_{c'} \text{count}(c')^\alpha} \]

- This helps because \( P_{\alpha}(c) > P(c) \) for rare \( c \)
- Consider two events, \( P(a) = .99 \) and \( P(b) = .01 \) (here we use probability to show the effect)
  \[ P_{\alpha}(a) = \frac{.97^{.75}}{.97^{.75} + .03^{.75}} = .97 \]
  \[ P_{\alpha}(b) = \frac{.03^{.75}}{.97^{.75} + .03^{.75}} = .03 \]

Add-k Smoothing

<table>
<thead>
<tr>
<th>Count(w,context)</th>
<th>computer</th>
<th>data</th>
<th>pinch</th>
<th>result</th>
<th>sugar</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricot</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>pineapple</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>digital</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>information</td>
<td>3</td>
<td>8</td>
<td>2</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

\[ p_{ij} = \frac{f_{ij}}{\sum_{j} f_{ij}} \]

\[ p(w) = \sum_{j} p(w|j) \]

Adding smoothing:

1. PMI is biased toward infrequent events
2. Very rare words have very high PMI values
3. Two solutions:
   1. Give rare words slightly higher probabilities
   2. Use add-k smoothing (which has a similar effect)
Measuring similarity

- Given 2 target words $v$ and $w$
- We'll need a way to measure their similarity.
- Most measure of vectors similarity are based on the:

**Dot product or inner product** from linear algebra (raw counts)

$$\text{dot-product}(\vec{v}, \vec{w}) = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- High when two vectors have large values in same dimensions.
- Low (in fact 0) for orthogonal vectors with zeros in complementary distribution.

**Problem with dot product**

$$\text{dot-product}(\vec{v}, \vec{w}) = \vec{v} \cdot \vec{w} = \frac{N}{N} \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- Dot product is longer if the vector is longer. Vector length:

$$|\vec{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

- Vectors are longer if they have higher values in each dimension
- That means more frequent words will have higher dot products
- That’s bad: we don’t want a similarity metric to be sensitive to word frequency

**Solution: cosine**

- Just divide the dot product by the length of the two vectors!

$$\frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$

- This turns out to be the cosine of the angle between them!

$$\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$

**Cosine for computing similarity**

$$\cos(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}$$

$v$ is the PPMI value for word $v$ in context $i$

$w$ is the PPMI value for word $w$ in context $i$

$\cos(\vec{v}, \vec{w})$ is the cosine similarity of $\vec{v}$ and $\vec{w}$

**Cosine as a similarity metric**

- 1: vectors point in opposite directions
- 0: vectors point in same directions
- 0: vectors are orthogonal

- Raw frequency or PPMI are non-negative, so cosine range 0-1
Which pair of words is more similar?

\[
\text{cosine}(\text{apricot}, \text{information}) = \frac{2 + 0 + 0}{\sqrt{4 + 1 + 1 + 1}} = \frac{2}{\sqrt{6}} \approx 0.63
\]

\[
\text{cosine}(\text{digital}, \text{information}) = \frac{0 + 0 + 2}{\sqrt{1 + 1 + 2 + 2}} = \frac{2}{\sqrt{6}} \approx 0.63
\]

\[
\text{cosine}(\text{apricot}, \text{digital}) = \frac{0 + 0 + 0}{\sqrt{1 + 1 + 1 + 1}} = 0
\]

Visualizing vectors and angles

Clustering vectors to visualize similarity in co-occurrence matrices

Other possible similarity measures

\[
\sin_{\cosine}(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}
\]

\[
\sin_{\text{Jaccard}}(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N} \min(v_i, w_i)}{\sum_{i=1}^{N} \max(v_i, w_i)}
\]

\[
\sin_{\text{Dice}}(\vec{v}, \vec{w}) = 2 \sum_{i=1}^{N} \min(v_i, w_i) / \left( \sum_{i=1}^{N} v_i + \sum_{i=1}^{N} w_i \right)
\]

Using syntax to define a word’s context

- Zellig Harris (1968)
  “The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities”

- Two words are similar if they have similar syntactic contexts

Duty and responsibility have similar syntactic distribution:

<table>
<thead>
<tr>
<th>Modified by adjectives</th>
<th>additional, administrative, assumed, collective, congressional, constitutional, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects of verbs</td>
<td>assert, assign, assume, attend to, avoid, become, breach, ...</td>
</tr>
</tbody>
</table>

Co-occurrence vectors based on syntactic dependencies

Dekang Lin, 1998: Automatic Retrieval and Clustering of Similar Words

- Each dimension: a context word in one of 8 grammatical relations
- Subject-of: “absorb”
- Instead of a vector of |V| features, a vector of |R|V|
- Example: counts for the word cell:
Co-occurrence vectors based on syntactic dependencies

- Each dimension: a context word in one of R grammatical relations
- Subject of "absorb"

Instead of a vector of $|V|$ features, a vector of $R|V|$

Example: counts for the word cell:

Hindle, Don. 1990. Noun Classification from Predicate-Argument Structure. ACL

PMI applied to dependency relations

Object of "drink" | Count | IDF
--- | --- | ---
tea | 2 | 11.8
liquid | 2 | 10.5
wine | 2 | 9.3
anything | 3 | 5.2
it | 3 | 1.3

"Drink it" more common than "drink wine"
But "wine" is a better "drinkable" thing than "it"

Syntactic dependencies for dimensions

- Alternative (Pado and Lapata 2007): Instead of having a $|V| \times |V|$ matrix
- Have a $|V| \times |V|$ matrix
- Counts of words that occur in one of R dependencies (subject, object, etc).
- So $M("cell","absorb") = \text{count}(\text{subj}(\text{cell},\text{absorb})) + \text{count}(\text{obj}(\text{cell},\text{absorb})) + ...$

Alternative to PPMI for measuring association

- The combination of two factors
  - Term frequency (Luhn 1957): frequency of the word (can be logged)
  - Inverse document frequency (IDF) (Spark Jones 1972)
- $N$ is the total number of documents
- $df_i = \text{document frequency of word } i$
- $w_{ij} = \text{frequency of word } i \text{ in document } j$

$$\text{idf}_i = \log \left( \frac{N}{df_i} \right)$$

Evaluating similarity (Revisit)

- Extrinsic (task-based, end-to-end) Evaluation:
  - Question Answering
  - Spell Checking
  - Essay grading
- Intrinsic Evaluation:
  - Correlation between algorithm and human word similarity ratings
  - Workman: 3-10 noun pairs rated 0.10. example: car+house=0.77
  - Taking TOEFL multiple-choice vocabulary tests
  - Bed was closest in meaning to:
  - imposed, believed, requested, correlated
Summary and next step

- Distributional (vector) models of meaning
  - Sparse (PPMI-weighted word-word co-occurrence matrices)
  - Dense:
    - Word-word SVD (50-2000 dimensions)
    - Neural language models: Skip-grams and CBOW (100-1000 dimensions)