SI485i : NLP

Set 11
Distributional Similarity

slides adapted from Dan Jurafsky and Bill MacCartney
Distributional methods

• Firth (1957)
  “You shall know a word by the company it keeps!”

• Example from Nida (1975) noted by Lin:
  A bottle of tezgüino is on the table
  Everybody likes tezgüino
  Tezgüino makes you tipsy
  We make tezgüino out of corn

• Intuition:
  • Just from these contexts, a human could guess meaning of tezgüino
  • So we should look at the surrounding contexts, see what other words have similar context
You can get a quick & dirty impression of what words show up in a given context by putting a * in your Google query:

"drank a bottle of *"

Hi I'm Noreen and I once drank a bottle of wine in under 4 minutes
SHE DRANK A BOTTLE OF JACK?! harleyabshireblondie.
he drank a bottle of beer like any man
I topped off some salted peanuts and drank a bottle of water
The partygoers drank a bottle of champagne.
MR WEST IS DEAD AS A HAMMER HE DRANK A BOTTLE OF ROGAINE
aug 29th 2010 i drank a bottle of Odwalla Pomegranate Juice and got ...
The 3 of us drank a bottle of Naga Viper Sauce ...
We drank a bottle of Lemelson pinot noir from Oregon ($52)
she drank a bottle of bleach nearly killing herself, "to clean herself from her wedding"
Context vector

• Target word \( w \)
• We have a binary feature \( f_i \) for each word \( v_i \)
  • \( f_i = \) “word \( v_i \) occurs in the neighborhood of \( w \)”

\[ w = (f_1, f_2, f_3, \ldots, f_N) \]

If \( w = \) tezgüino, \( v_1 = \) bottle, \( v_2 = \) drunk, \( v_3 = \) matrix
\[ w = (1, 1, 0, \ldots) \]
Intuition

- Define two words by these sparse feature vectors
- Apply a vector distance metric
- Call two words similar if their vectors are similar

<table>
<thead>
<tr>
<th></th>
<th>arts</th>
<th>boil</th>
<th>data</th>
<th>function</th>
<th>large</th>
<th>sugar</th>
<th>summarized</th>
<th>water</th>
</tr>
</thead>
<tbody>
<tr>
<td>apricot</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>pineapple</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>digital</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>information</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Distributional similarity

We just need to specify 3 things:

1. How the co-occurrence terms are defined
2. How terms are weighted
   • (Boolean? Frequency? Logs? Mutual information?)
3. What vector similarity metric should we use?
   • Euclidean distance? Cosine? Jaccard? Dice?
1. Defining co-occurrence vectors

- Windows of neighboring words (n words to the left…)
  - Bag-of-words
  - We generally remove stop words
- But we lose any sense of syntax
- Instead, use the words occurring in particular grammatical relations
Defining co-occurrence vectors

“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.” Zellig Harris (1968)

Idea: parse the sentence, extract grammatical dependencies

I discovered dried tangerines:
- discover (subject I)  I (subj-of discover)
- tangerine (obj-of discover)  tangerine (adj-mod dried)
- dried (adj-mod-of tangerine)  (adj-mod of tangerine)
Co-occurrence vectors based on grammatical dependencies

For the word *cell*: vector of $N*R$ features

$(R$ is the number of dependency relations$)$

<table>
<thead>
<tr>
<th></th>
<th>subj-of, absorb</th>
<th>subj-of, adapt</th>
<th>subj-of, behave</th>
<th>pobj-of, inside</th>
<th>pobj-of, into</th>
<th>nmod-of, abnormality</th>
<th>nmod-of, anemia</th>
<th>nmod-of, architecture</th>
<th>obj-of, attack</th>
<th>obj-of, call</th>
<th>obj-of, come from</th>
<th>obj-of, decorate</th>
<th>nmod, bacteria</th>
<th>nmod, bone</th>
<th>nmod, body</th>
<th>nmod, bone marrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>cell</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>30</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>6</td>
<td>11</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Group Exercise

• Search “Naval Academy” and create a vector.
• What other school is most similar? Most different?
  • Compare vectors
2. Weighting the counts
(“Measures of association with context”)

- We have been using the frequency count of some feature as its weight or value
  - But we could use any function of this frequency

- Let’s consider one feature \( f = (r, w') = (\text{obj-of}, \text{attack}) \)
  - \( P(f \mid w) = \frac{\text{count}(f, w)}{\text{count}(w)} \)

- \( \text{Assoc}_{\text{prob}}(w, f) = p(f \mid w) \)
Frequency-based problems

Objects of the verb *drink*:

- Water 7
- Champagne 4
- It 3
- Much 3
- Anything 3
- Liquid 2
- Wine 2

- “drink it” is more common than “drink wine”!
  - But “wine” is a better “drinkable” thing than “it”
- We need to control for *expected frequency*
- Instead, normalize by the *expected frequency*
Weighting: Mutual Information

- **Pointwise mutual information**: measure of how often two events $x$ and $y$ occur, compared with what we would expect if they were independent:

$$I(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)}$$

- PMI between a target word $w$ and a feature $f$:

$$\text{assocPMI}(w, f) = \log_2 \frac{P(w, f)}{P(w)P(f)}$$
# Mutual information intuition

## Objects of the verb *drink*

<table>
<thead>
<tr>
<th>Object</th>
<th>Count</th>
<th>PMI assoc</th>
<th>Object</th>
<th>Count</th>
<th>PMI assoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>bunch beer</td>
<td>2</td>
<td>12.34</td>
<td>wine</td>
<td>2</td>
<td>9.34</td>
</tr>
<tr>
<td>tea</td>
<td>2</td>
<td>11.75</td>
<td>water</td>
<td>7</td>
<td>7.65</td>
</tr>
<tr>
<td>Pepsi</td>
<td>2</td>
<td>11.75</td>
<td>anything</td>
<td>3</td>
<td>5.15</td>
</tr>
<tr>
<td>champagne</td>
<td>4</td>
<td>11.75</td>
<td>much</td>
<td>3</td>
<td>5.15</td>
</tr>
<tr>
<td>liquid</td>
<td>2</td>
<td>10.53</td>
<td>it</td>
<td>3</td>
<td>1.25</td>
</tr>
<tr>
<td>beer</td>
<td>5</td>
<td>10.20</td>
<td>&lt;SOME AMOUNT&gt;</td>
<td>2</td>
<td>1.22</td>
</tr>
</tbody>
</table>
Summary: weightings

- See Manning and Schuetze (1999) for more

\[
\begin{align*}
\text{assoc}_{\text{prob}}(w, f) &= P(f|w) \\
\text{assoc}_{\text{PMI}}(w, f) &= \log_2 \frac{P(w,f)}{P(w)P(f)} \\
\text{assoc}_{\text{Lin}}(w, f) &= \log_2 \frac{P(w,f)}{P(w)P(r|w)P(w'|w)} \\
\text{assoc}_{t-test}(w, f) &= \frac{P(w,f) - P(w)P(f)}{\sqrt{P(f)P(w)}}
\end{align*}
\]
3. Defining vector similarity

Euclidean($\vec{a}, \vec{b}$) = L2($\vec{a}, \vec{b}$)

Manhattan($\vec{a}, \vec{b}$) = L1($\vec{a}, \vec{b}$)
Summary of similarity measures

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

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

\[ \text{sim}_{\text{Dice}}(\vec{v}, \vec{w}) = \frac{2 \times \sum_{i=1}^{N} \min(v_i, w_i)}{\sum_{i=1}^{N} (v_i + w_i)} \]

\[ \text{sim}_{\text{JS}}(\vec{v} || \vec{w}) = D(\vec{v} | \frac{\vec{v} + \vec{w}}{2}) + D(\vec{w} | \frac{\vec{v} + \vec{w}}{2}) \]
Evaluating similarity measures

- Intrinsic evaluation
  - Correlation with word similarity ratings from humans

- Extrinsic (task-based, end-to-end) evaluation
  - Malapropism (spelling error) detection
  - WSD
  - Essay grading
  - Plagiarism detection
  - Taking TOEFL multiple-choice vocabulary tests
  - Language modeling in some application
MAINFRAMES
Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Ebay, Amazon, and computing-giant

MAINFRAMES
Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.