Distributional Semantics and Word Embeddings
Announcements

• Midterm returned at end of class today
  • Only exams that were taken on Thursday

• Today: moving into neural nets via word embeddings

• Tuesday: Introduction to basic neural net architecture. Chris Kedzie to lecture.

• Homework out on Tuesday

• Language applications using different architectures
Lexical semantics

- Hypernymy
  - mammal
  - pet
  - pack
  - meronymy
  - holonymy
  - paw
  - hyponymy
  - dog
  - poodle
  - puppy
  - canine

Slide from Kapil Thadani
Lexical semantics

Slide from Kapil Thadani
Methods so far

• **WordNet**: an amazing resource.. *But*

• **What are some of the disadvantages?**
Methods so far

- Bag of words
  - Simple and interpretable

- In vector space, represent a sentence
  
  \[
  \begin{bmatrix}
  0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
  \end{bmatrix}
  \]
  “one-hot” vector
  
  Values could be frequency, TF*IDF

- Sparse representation
  - Dimensionality: 50K unigrams, 500K bigrams

- Curse of dimensionality!
From Symbolic to Distributed Representations

• Its problem, e.g., for web search
  • If user searches for [Dell notebook battery], should match documents with “Dell laptop battery”
  • If user searches for [Seattle motel] should match documents containing “Seattle hotel”

• But
  • Motel [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
  • Hotel [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]

• Our query and document vectors are orthogonal
  There is no natural notion of similarity in a set of one-hot vectors

• -> Explore a direct approach where vectors encode it
Distributional Semantics

• “You shall know a word by the company it keeps” [J.R. Firth 1957]
  • *Marco saw a hairy little wampunuk hiding behind a tree*

• Words that occur in similar contexts have similar meaning

• Record word co-occurrence within a window over a large corpus
Word Context Matrices

• Each row \( i \) represents a word
• Each column \( j \) represents a linguistic context
• Matrix \( \text{Matrix}_{ij} \) represents strength of association
  • \( M^f \in \mathbb{R}, M^f_{i,j} = f(w_i, c_j) \) where \( f \) is an association measure of the strength between a word and a context

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>hamburger</th>
<th>book</th>
<th>gift</th>
<th>spoon</th>
</tr>
</thead>
<tbody>
<tr>
<td>ate</td>
<td>.45</td>
<td>.56</td>
<td>.02</td>
<td>.03</td>
<td>.3</td>
</tr>
<tr>
<td>gave</td>
<td>.46</td>
<td>.13</td>
<td>.67</td>
<td>.7</td>
<td>.25</td>
</tr>
<tr>
<td>took</td>
<td>.46</td>
<td>.1</td>
<td>.7</td>
<td>.5</td>
<td>.3</td>
</tr>
</tbody>
</table>
Associations and Similarity

• Effective association measure: Pointwise Mutual Information (PMI)
  \[
  \log \frac{P(w,c)}{P(w)P(c)} = \log \frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}
  \]

• Compute similarity between words and text
  • Cosine Similarity
    \[
    \sum_i u_i \cdot v_i / \sqrt{\sum_i (u_i)^2} \cdot \sqrt{\sum_i (v_i)^2}
    \]
Dimensionality Reduction

• Captures context, but still has sparseness issues

• Singular value decomposition (SVD)
  • Factors matrix $M$ into two narrow matrices: $W$, a word matrix, and $C$, a context matrix such that $WC^T = M'$ is the best rank-$d$ approximation of $M$

• A “smoothed” version of $M$
  • Adds words to contexts if other words in this context seem to co-locate with each other
  • Represents each word as a dense $d$-dimensional vector instead of a sparse $|V_C|$ one
Latent Semantic Analysis

Construct term-document matrix

\[ M = \begin{pmatrix}
  w_1^{(1)} & w_1^{(2)} & \cdots \\
  w_2^{(1)} & \ddots & \\
  \vdots & & \\
\end{pmatrix}
\]

Singular value decomposition

\[ M \approx \begin{pmatrix}
  u_1 & u_2 & u_3 & \cdots \\
  \lambda_1 & \lambda_2 & \lambda_3 & \cdots \\
  k & & & \\
\end{pmatrix}
\]

Select top \( k \) singular vectors for \( k \)-dim embeddings of words/docs
Neural Nets

• A family of models within deep learning

• The machine learning approaches we have seen to date rely on “feature engineering”

• With neural nets, instead we learn by optimizing a set of parameters
Why “Deep Learning”? 

- **Representation learning** attempts to automatically learn good features or representations.

- **Deep learning** algorithms attempt to learn (multiple levels of) representation and an output.

- From “raw” inputs $x$ (e.g., sound, characters, words)

Slide adapted from Chris Manning
Reasons for Exploring Deep Learning

• Manually designed features can be over-specific or take a long time to design
  • ... but can provide an intuition about the solution

• Learned features are easy to adapt

• Deep learning provides a very flexible framework for representing word, visual, and linguistic information

• Both supervised and unsupervised methods
Progress with deep learning

- Huge leaps forward with
  - Speech
  - Vision
- More modest advances in other areas
From Distributional Semantics to Neural Networks

• Instead of count-based methods, distributed representations of word meaning

• Each word associated with a vector where meaning is captured in different dimensions as well as in dimensions of other words

• Dimensions in a distributed representation are not interpretable

• Specific dimensions do not correspond to specific concepts
Basic Idea of Learning Neural Network Embeddings

• Define a model that aims to predict between a center word $w_t$ and context words in terms of word vectors
  \[ p(\text{context} | w_t) = \ldots \]

  Which has a loss function, e.g.,
  \[ J = 1 - p(w_{-t} | w_t) \]

• We look at many positions $t$ in a large corpus

• We keep adjusting the vector representations of words to minimize loss

Slide adapted from Chris Manning
Embeddings Are Magic

\[ \text{vector('king')} - \text{vector('man')} + \text{vector('woman')} \approx \text{vector('queen')} \]
Relevant approaches: Yoav and Goldberg

• Chapter 9: A neural probabilistic language model (Bengio et al 2003)

• Chapter 10, p. 113 NLP (almost) from Scratch (Collobert & Weston 2008)

• Chapter 10, p 114 Word2vec (Mikolog et al 2013)
Main Idea of word2vec

- Predict between every word and its context

- Two algorithms
  - Skip-gram (SG)
    Predict context words given target (position independent)
  - Continuous Bag of Words (CBOW)
    Predict target word from bag-of-words context

Slide adapted from Chris Manning
Training Methods

• Two (moderately efficient) training methods
  - Hierarchical softmax
  - Negative sampling

Today: naïve softmax

Slide adapted from Chris Manning
Instead, a **bank** can hold the investments in a custodial account.

But as agriculture burgeons on the east **bank**, the river will shrink.
Objective Function

- Maximize the probability of context words given the center word

\[
J'(\Theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m} P(w_{t+j} | w_t \Theta) \\
\prod_{j \neq 0}
\]

Negative log likelihood

\[
J'(\Theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m} \log P(w_{t+j} | w_t) \\
\sum_{j \neq 0}
\]

Where \( \Theta \) represents all variables to be optimized

Slide adapted from Chris Manning
Softmax
using word c to obtain probability of word o

- Convert $P(w_{t+j} | w_t)$

$$P(o | c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^v \exp(u_w^T v_c)}$$

exponentiate normalize to make positive

where $o$ is the outside (or output) word index and $c$ is the center word index, $v_c$ and $u_o$ are center and outside vectors of indices c and o

Slide adapted from Chris Manning
Softmax
Dot Product

- \( u^T v = u \cdot v = \sum_{i=1}^{n} u_i v_i \)

- Bigger if \( u \) and \( v \) are more similar
word2vec

Skip-gram

- Predict context $w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}$ given target $w_t$

$$\ell(W, U) = - \sum_{i=0}^{c} \log p(w_{t+i}|w_t)$$

$$p(w_j|w_t) = \frac{e^{U_j h_t}}{\sum_k e^{U_k h_t}}$$

$$h_t = W^\top w_t$$
Embeddings Are Magic

\[ \text{vector('king')} - \text{vector('man')} + \text{vector('woman')} \approx \text{vector('queen')} \]
<table>
<thead>
<tr>
<th>Czech + currency</th>
<th>Vietnam + capital</th>
<th>German + airlines</th>
<th>Russian + river</th>
<th>French + actress</th>
</tr>
</thead>
<tbody>
<tr>
<td>koruna</td>
<td>Hanoi</td>
<td>airline Lufthansa</td>
<td>Moscow</td>
<td>Juliette Binoche</td>
</tr>
<tr>
<td>Check crown</td>
<td>Ho Chi Minh City</td>
<td>carrier Lufthansa</td>
<td>Volga River</td>
<td>Vanessa Paradis</td>
</tr>
<tr>
<td>Polish zolty</td>
<td>Viet Nam</td>
<td>flag carrier Lufthansa</td>
<td>upriver Russia</td>
<td>Charlotte Gainsbourg</td>
</tr>
<tr>
<td>CTK</td>
<td>Vietnamese</td>
<td>Lufthansa</td>
<td>Russia</td>
<td>Cecile De</td>
</tr>
</tbody>
</table>

Additive compositionality
Evaluating Embeddings

• Nearest Neighbors
• Analogies
  • (A:B)::(C:?)
• Information Retrieval
• Semantic Hashing
## Similarity Data Sets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Word pairs</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>RG</td>
<td>65</td>
<td>Rubenstein and Goodenough (1965)</td>
</tr>
<tr>
<td>MC</td>
<td>30</td>
<td>Miller and Charles (1991)</td>
</tr>
<tr>
<td>WS-353</td>
<td>353</td>
<td>Finkelstein et al. (2002)</td>
</tr>
<tr>
<td>MTurk-287</td>
<td>287</td>
<td>Radinsky et al. (2011)</td>
</tr>
<tr>
<td>MTurk-771</td>
<td>771</td>
<td>Halawi et al. (2012)</td>
</tr>
<tr>
<td>MEN</td>
<td>3000</td>
<td>Bruni et al. (2012)</td>
</tr>
<tr>
<td>RW</td>
<td>2034</td>
<td>Luong et al. (2013)</td>
</tr>
<tr>
<td>Verb</td>
<td>144</td>
<td>Baker et al. (2014)</td>
</tr>
<tr>
<td>SimLex</td>
<td>999</td>
<td>Hill et al. (2014)</td>
</tr>
</tbody>
</table>

[Table from Faruqui et al. 2016]
Table 1: Examples of five types of semantic and nine types of syntactic questions in the Semantic-Syntactic Word Relationship test set.

<table>
<thead>
<tr>
<th>Type of relationship</th>
<th>Word Pair 1</th>
<th>Word Pair 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common capital city</td>
<td>Athens</td>
<td>Greece</td>
</tr>
<tr>
<td>All capital cities</td>
<td>Astana</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>Currency</td>
<td>Angola</td>
<td>kwanza</td>
</tr>
<tr>
<td>City-in-state</td>
<td>Chicago</td>
<td>Illinois</td>
</tr>
<tr>
<td>Man-Woman</td>
<td>brother</td>
<td>sister</td>
</tr>
<tr>
<td>Adjective to adverb</td>
<td>apparent</td>
<td>apparently</td>
</tr>
<tr>
<td>Opposite</td>
<td>possibly</td>
<td>impossibly</td>
</tr>
<tr>
<td>Comparative</td>
<td>great</td>
<td>greater</td>
</tr>
<tr>
<td>Superlative</td>
<td>easy</td>
<td>easiest</td>
</tr>
<tr>
<td>Present Participle</td>
<td>think</td>
<td>thinking</td>
</tr>
<tr>
<td>Nationality adjective</td>
<td>Switzerland</td>
<td>Swiss</td>
</tr>
<tr>
<td>Past tense</td>
<td>walking</td>
<td>walked</td>
</tr>
<tr>
<td>Plural nouns</td>
<td>mouse</td>
<td>mice</td>
</tr>
<tr>
<td>Plural verbs</td>
<td>work</td>
<td>works</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rapid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ethical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tough</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lucky</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reading</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cambodian</td>
</tr>
<tr>
<td></td>
<td></td>
<td>swam</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dollars</td>
</tr>
<tr>
<td></td>
<td></td>
<td>speaks</td>
</tr>
</tbody>
</table>

[Mikolov et al. 2013]
Semantic Hashing

Figure 5: A 2-dimensional embedding of the 128-bit codes using stochastic neighbor embedding for the 20 Newsgroups data (left panel) and the Reuters RCV2 corpus (right panel). See in color for better visualization.
<table>
<thead>
<tr>
<th>Newspapers</th>
<th>Baltimore</th>
<th>Baltimore Sun</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New York</strong></td>
<td><strong>New York Times</strong></td>
<td><strong>Cincinnati</strong></td>
</tr>
<tr>
<td>San Jose</td>
<td>San Jose Mercury News</td>
<td>Cincinnati Enquirer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NHL Teams</th>
<th>Montreal</th>
<th>Montreal Canadiens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston</td>
<td>Boston Bruins</td>
<td></td>
</tr>
<tr>
<td>Phoenix</td>
<td>Phoenix Coyotes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NBA Teams</th>
<th>Toronto</th>
<th>Toronto Raptors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detroit</td>
<td>Detroit Pistons</td>
<td></td>
</tr>
<tr>
<td>Oakland</td>
<td>Golden State Warriors</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Airlines</th>
<th>Spain</th>
<th>Spainair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>Austrian Airlines</td>
<td></td>
</tr>
<tr>
<td>Belgium</td>
<td>Brussels Airlines</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company executives</th>
<th>Larry Page</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steve Ballmer</td>
<td>Microsoft</td>
<td></td>
</tr>
<tr>
<td>Samuel J. Palmisano</td>
<td>IBM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Werner Vogels</td>
<td>Amazon</td>
</tr>
</tbody>
</table>

**Phrase analogies**
How are word embeddings used?

• As features in supervised systems

• As the main representation with a neural net application/task
Are Distributional Semantics and Word Embeddings all that different?
Homework2

- Max 99.6, Min 4, Stdev: 21.4
- Mean 82.2, Median 92.1
- Vast majority of F1 scores between 90 and 96.5.
Midterm

• Max: 95, Min: 22.5

• Mean: 66.6, Median 68.5

• Standard Deviation: 15

• Will be curved and the curve will be provided in the next lecture