Text Similarity
Announcements

• Note problems with Midterm multiple choice questions 2 and 3. If you got them wrong, you will get credit. Bring your exam back to TA hours.

• There will be a recitation tomorrow on HW3. Be sure to provide your interest and availability on Piazza.

• You have 2 weeks for HW3. Due date on the assignment is correct. I have updated web site.
Time to Reflect

• Why does a neural net work?

• What is it good at?

• Empirical vs theory: what do we know?
Supervised Machine Learning

Parameters (things we're learning)

\[ w \]

Input feature vector

\[ x \]

Sigmoid or other nonlinearity

\[ \sigma \]

Predicted value

\[ \hat{y} \]
Supervised Machine Learning

How wrong were we?

Update parameters

\[ \text{cost}(\hat{y}, y) \]

Actual value
Highlights of Neural Nets

• Learn a representation, not just to predict

• Critical component is the embedding layer
  • Mapping from discrete symbols to continuous vectors in low-dimensional space
  • *Semantic representation: Distributed*

• Feed-forward neural networks (multi-layer perceptron) can be used anywhere a linear classifier is used
  • Superior performance often due to non-linearity

• Which parameter values, which neural net (RNN, CNN, LSTM) are best for a task is determined experimentally
Huge leap forward in ‘Speech Recognition’ and ‘Image Recognition’

Accuracy

ASR

Image Recognition

~95

200 0 200 1 200 2 200 3 200 4 200 5 200 6 200 7 200 8 200 9 201 0 201 1 201 2 201 3 201 4 201 5 201 6 201 7

Slide credit: Omid Bakhshandeh
Trend in NLP Tasks

Accuracy


NP Chunking  PO Parsing  NER  Paraphrase Identification

Slide credit: Omid Bakhshandeh
Time to Reflect

• Your reactions to neural nets so far
  • Are they still confusing?
  • Do you need to see more?
  • Are you convinced (yet)?
  • Are they intriguing?
  • Do you want to see more?
  • Success is empirically determined: is empirical vs theoretical problematic?
Synonymy and Paraphrase

- A critical piece of text interpretation
- Can be domain-specific

- Word synonomy:
  - General domain: “hot” “sexy” but biology?

<table>
<thead>
<tr>
<th>General</th>
<th>Biology</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>Warm, sexy, exciting</td>
</tr>
<tr>
<td>treat</td>
<td>Address, handle</td>
</tr>
<tr>
<td>head</td>
<td>Leader, boss, mind</td>
</tr>
</tbody>
</table>
Sentential Paraphrase

- Paraphrases extracted from different translations of the same novel

<table>
<thead>
<tr>
<th>Original</th>
<th>Paraphrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emma burst into tears and he tried to comfort her, saying things to make her smile.</td>
<td>Emma cried, and he tried to console her, adorning his words with puns.</td>
</tr>
<tr>
<td>And finally, dazzlingly white, it shone high above them in the empty sky.</td>
<td>It appeared white and dazzling, in the empty heavens.</td>
</tr>
<tr>
<td>People said “The Evening Noise is sounding, the sun is setting.”</td>
<td>“The evening bell is ringing” people used to say.</td>
</tr>
</tbody>
</table>

Examples from Barzilay
## Phrasal paraphrases

<table>
<thead>
<tr>
<th>Original Phrase</th>
<th>Paraphrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>King’s son</td>
<td>Son of the king</td>
</tr>
<tr>
<td>In bottles</td>
<td>bottled</td>
</tr>
<tr>
<td>Start to talk</td>
<td>Start talking</td>
</tr>
<tr>
<td>Suddenly came</td>
<td>Came suddenly</td>
</tr>
<tr>
<td>Make appearance</td>
<td>appear</td>
</tr>
</tbody>
</table>

Examples from Barzilay
Types of Text Similarity

• Many types of text similarity exist:
  • Morphological similarity (e.g., respect-respectful)
  • Spelling similarity (e.g., theater-theatre)
  • Synonymy (e.g., talkative-chatty)
  • Homophony (e.g., raise-raze-rays)
  • Semantic similarity (e.g., cat-tabby)
  • Sentence similarity (e.g., paraphrases)
  • Document similarity (e.g., two news stories on the same event)
Tasks requiring text similarity

• Information retrieval

• Machine translation

• Summarization

• Inference
Using word embeddings to compute similarity

- Cosine similarity

\[ \text{sim}_{\cos}(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} \]

- When vectors have unit length, cosine similarity is the dot product

- Common to normalize embeddings matrix so that each row as unit length
Similarity Measures (Cont.)

- Cosine similarity: similarity of two vectors, normalized

\[
\cos(X, Y) = \frac{x_1y_1 + x_2y_2 + \ldots + x_ny_n}{\sqrt{x_1^2 + \ldots + x_n^2} \cdot \sqrt{y_1^2 + \ldots + y_n^2}} = \frac{\sum_{i=1}^{n} x_iy_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}
\]
Document Similarity

- Used in information retrieval to determine which document \((d_1 \text{ or } d_2)\) is more similar to a given query \(q\).
- Documents and queries are represented in the same space.
- Angle (or cosine) is a proxy for similarity between two vectors.

Slide from Radev
Quiz

• Given three documents
  \[ D_1 = <1,3> \]
  \[ D_2 = <10,30> \]
  \[ D_3 = <3,1> \]

• Compute the cosine scores
  \[ \sigma(D_1,D_2) \]
  \[ \sigma(D_1,D_3) \]

• What do the numbers tell you?

Slide from Radev
It is the cosine similarity between D1 = and D2 =
Quiz

• What is the range of values that the cosine scores can take?
Finding Similar Words

- \( \text{sim}_{\cos}(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} \)

Finding k most similar words where E an embedding matrix for all words.

- \( w = E_{[w]} \)
- \( S = Ew \)
  - A vector of similarities
  - \( S_{[i]} = \) similarity of \( w \) to ith word
  - K-most similar words?

- How can we find the k-most similar words that are also orthographically similar?
can we find the k-most similar words that are orthographically similar
Similarity to a group of words

• Given: \(w_i \ldots w_k\) that are semantically similar

• Find \(w_j\) such that it is the most semantically similar to the group

• Define similarity as average similarity to the group: \(1/k \sum_{i=1}^{k} \text{sim}_{\cos}(w, w_i)\)

\[
s = E(w) \frac{E(w_1 + w_2 + \ldots + w_k)}{k}
\]

• How would we compute odd word out?
Short Document Similarity

• We can train a model or *we can just use word embeddings*

• Suitable for very short texts such as queries, newspaper headlines or tweets

• Similarity = the sum of the pairwise similarities of all words in the document
Computing Document Similarity

- Where \( D_1 = w_1^1 \ldots w_m^1 \) and \( D_2 = w_1^2 \ldots w_n^2 \)

\[
\text{sim}_{\text{doc}}(D_1, D_2) = \sum_{i=1}^{m} \sum_{j=1}^{n} \cos(w_i^1, w_j^2)
\]

- Equivalent to:

\[
\text{sim}_{\text{doc}}(D_1, D_2) = (\sum_{i=1}^{m} w_i^1) \cdot (\sum_{j=1}^{n} w_j^2)
\]

- Allows: Document collection \( D \) is a matrix where each row \( i \) is a document. Similarity with a new document:

\[
s = D \cdot (\sum_{i=1}^{n} w_i')
\]
Analogy Solving Task

- \[ w_{\text{king}} - w_{\text{man}} + w_{\text{woman}} \approx w_{\text{queen}} \]

\[
\text{analogy}(m : w \rightarrow k : ?) = \arg\max_{v \in V \setminus \{m, w, k\}} \cos(v, k - m + w)
\]

- Equivalent to (COS-ADD) (Levy and Goldberg 2014)

\[
\text{analogy}(m : w \rightarrow k : ?) = \arg\max_{v \in V \setminus \{m, w, k\}} \cos(v, k) - \cos(v, m) + \cos(v, w)
\]

- “... it is not clear what success on a benchmark of analogy tasks says about the quality of word embeddings beyond their suitability for solving this particular task.” (Goldberg 2017)
Using WordNet and other paraphrase corpora

- (PPDB) Penn Paraphrase Database (Pavlick and Callison-Burch)

- Can we use word pairs that reflect similarity better for the task?
  - Pre-trained embeddings $E$
  - Graph $G$ representing similar word pairs
  - Search for a new word embedding matrix $E'$ whose rows are close to $E$ but also close to $G$

- Methods for combining pre-trained word embeddings with smaller, specialized embeddings
Caveats

• Don’t just use off-the-shelf word embeddings blindly

• Experiment with corpus and hyper-parameter settings

• When using off-the-shelf embeddings, use the same tokenization and normalization
Resources

• Word embeddings
  • https://code.google.com/p/word2vec/

• Neural net platforms
  • Keras https://keras.io/
  • Pytorch http://pytorch.org/
  • Tensorflow https://www.tensorflow.org/
  • Theano http://deeplearning.net/software/theano/
Language is made up of sequences

• So far we have seen embeddings for words
  • (and methods for combining through vector concatenation and arithmetic)

• But how can we account for sequences?
  • Words as sequences of letters
  • Sentences as sequences of words
  • Documents as sequences of sentences
Recurrent Neural Networks

- Represent arbitrarily sized sequences in fixed-size vector

- Good at capturing statistical regularities in sequences (order matters)

- Include simple RNNs, Long short-term memory (LSTMs), Gated Recurrent Unit (GRUs)
Learning word meaning from their morphs

Logical entailment using compositional semantics via RNNs

Figure 1: Morphological Recursive Neural Network. A vector representation for the word “unfortunately” is constructed from morphemic vectors: \textit{un}_\text{pre}, \textit{fortunate}_\text{stm}, \textit{ly}_\text{suf}. Dotted nodes are computed on-the-fly and not in the lexicon.

[Thang et al. 2013]

Figure 1: In our model, two separate tree-structured networks build up vector representations for each of two sentences using either NN or NTN layer functions. A comparison layer then uses the resulting vectors to produce features for a classifier.

[Bowman et al. 2014]
Machine Translation (Sequences)

- Sequence-to-sequence
  - Sutskever et al. 2014

![Diagram of sequence-to-sequence model]

**Figure 1**: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.
RNN Abstraction

- RNN is a function that takes an arbitrary length sequence as input and returns a single $d_{\text{out}}$ dimensional vector as output
  - Input: $x_{1:n} = x_1 \ x_2 \ldots x_n \ (x_i \in \mathbb{R}^{d_{\text{in}}})$
  - Output: $y_n \in \mathbb{R}^{d_{\text{out}}}$

\[
y_n = \text{RNN}(x_{1:n})
\]
\[
x_i \in \mathbb{R}^{d_{\text{in}}} \quad y_n \in \mathbb{R}^{d_{\text{out}}}
\]

\[
y_{1:n} = \text{RNN}^*(x_{1:n})
\]
\[
y_i = \text{RNN}(x_{1:i})
\]
\[
x_i \in \mathbb{R}^{d_{\text{in}}} \quad y_i \in \mathbb{R}^{d_{\text{out}}}
\]

Output vector $y$ used for further prediction
RNN Characteristics

• Can condition on the entire sequence without resorting to the Markov assumption

• Can get very good language models as well as good performance on many other tasks
RNNs are defined recursively

- By means of a function $R$ taking as input a state vector $h_{i-1}$ and an input vector $x_i$

- Returns a new state vector $h_i$

- The state vector can be mapped to an output vector $y_i$ using a simple deterministic function

- And fed through softmax for classification.
Recurrent Neural Networks

\[ h^t = \sigma(W_{hh} h^{t-1} + W_{hx} x^t) \]

Slide from Radev
RNN

\[
\begin{align*}
    h_{t} &= \sigma(W_{h}h_{t-1} + W_{x}x_{t}) \\
    y_{t} &= \text{softmax}(W_{y}h_{t})
\end{align*}
\]

Slide from Radev
RNN

The cat sat

\[ y_3 = \text{softmax}(\text{max}(w_{01} h_0 + w_{11} x_1 + w_{21} h_1 + w_{31} h_2 + w_{23} h_2, y_{30})) \]
Updating Parameters of an RNN

Backpropagation through time

The cat sat

Slide from Radev
Next Time

• More on RNNs and their use in sentiment analysis