Deep Learning for Natural Language Processing
Topics

- Word embeddings
- Recurrent neural networks
- Long-short-term memory networks
- Neural machine translation
- Automatically generating image captions
Word meaning in NLP

• How do we capture meaning and context of words?

Synonyms:
“I loved the movie.”
“I adored the movie.”

Homonyms:
“I deposited the money in the bank.”
“I buried the money in the bank.”

Polysemy:
“I read a book today.”
“I wasn’t able to book the hotel room.”

Synechdoche:
“Today, Washington affirmed its opposition to the trade pact.”
Word Embeddings

“One of the most successful ideas of modern NLP”.

One example: Google’s **Word2Vec** algorithm
Word2Vec algorithm

○○○○○  ⋯  ○

○○○○○⋯○○○○○○○

○○○○○  ⋯  ○
Word2Vec algorithm

Input: One-hot representation of input word over vocabulary 10,000 units
Word2Vec algorithm

Input: One-hot representation of input word over vocabulary
10,000 units
**Word2Vec algorithm**

**Output:** Probability (for each word $w_i$ in vocabulary) that $w_i$ is nearby the input word in a sentence.
10,000 units

\[ \cdots \cdots \cdots \cdots \cdots \cdots \cdots \]

**Hidden layer** (linear activation function)
300 units

\[ \cdots \cdots \cdots \cdots \cdots \cdots \cdots \]

**Input:** One-hot representation of input word over vocabulary
10,000 units
**Word2Vec algorithm**

**Output:** Probability (for each word $w_i$ in vocabulary) that $w_i$ is nearby the input word in a sentence.  
10,000 units

Input: One-hot representation of input word over vocabulary  
10,000 units

Hidden layer (linear activation function)  
300 units

10,000 units  
$10,000 \times 300$ weights

Drawing of the model architecture:

```
Input: One-hot representation of input word over vocabulary  
10,000 units

10,000 units  
$10,000 \times 300$ weights

Hidden layer (linear activation function)  
300 units

300 units  
$300 \times 10,000$ weights

Output: Probability (for each word $w_i$ in vocabulary) that $w_i$ is nearby the input word in a sentence.  
10,000 units
```
Word2Vec training

- Training corpus of documents

- Collect pairs of nearby words

- Example “document”:
  
  *Every morning she drinks Starbucks coffee.*

Training pairs (window size = 3):

- (every, morning)  (morning, drinks)  (drinks, Starbucks)
- (every, she)       (she, drinks)     (drinks, coffee)
- (morning, she)     (she, Starbucks)  (Starbucks, coffee)
Word2Vec training via backpropagation

Target (probability that “Starbucks” is nearby “drinks”)

Starbucks

300 × 10,000 weights

Linear activation function

10,000 × 300 weights

drinks
Word2Vec training via backpropagation

Target (probability that “coffee” is nearby “drinks”)

300 × 10,000 weights

10,000 × 300 weights

Linear activation function

300 × 10,000 weights

10,000 × 300 weights

Linear activation function
Learned word vectors

10,000 × 300 weights

drinks
Some surprising results of word2vec

Figure 2: Left panel shows vector offsets for three word pairs illustrating the gender relation. Right panel shows a different projection, and the singular/plural relation for two words. In high-dimensional space, multiple relations can be embedded for a single word.
Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.
In this section we evaluate the Hierarchical Softmax (HS), Noise Contrastive Estimation, Negative Sampling, and subsampling of the training words. We used the analogical reasoning task\(^1\) introduced by Mikolov et al. [8]. The task consists of analogies such as “Germany” : “Berlin” :: “France” : ?, which are solved by finding a vector \(x\) such that \(\text{vec}(x)\) is closest to \(\text{vec} (“Berlin”) - \text{vec} (“Germany”) + \text{vec} (“France”)\) according to the cosine distance (we discard the input words from the search). This specific example is considered to have been answered correctly if \(x\) is “Paris”. The task has two broad categories: the syntactic analogies (such as “quick” : “quickly” :: “slow” : “slowly”) and the semantic analogies, such as the country to capital city relationship.

<table>
<thead>
<tr>
<th>Newspapers</th>
<th>Baltimore Sun</th>
<th>Baltimore Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>New York Times</td>
<td>San Jose Mercury News</td>
</tr>
<tr>
<td>San Jose</td>
<td>San Jose Mercury News</td>
<td>Cincinnati</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cincinnati Enquirer</td>
</tr>
<tr>
<td>NHL Teams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boston</td>
<td>Boston Bruins</td>
<td>Montreal Canadiens</td>
</tr>
<tr>
<td>Phoenix</td>
<td>Phoenix Coyotes</td>
<td>Nashville Predators</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBA Teams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detroit</td>
<td>Detroit Pistons</td>
<td>Toronto Raptors</td>
</tr>
<tr>
<td>Oakland</td>
<td>Golden State Warriors</td>
<td>Memphis Grizzlies</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Airlines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Austria</td>
<td>Austrian Airlines</td>
<td>Spain</td>
</tr>
<tr>
<td>Belgium</td>
<td>Brussels Airlines</td>
<td>Greece</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Company executives</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steve Ballmer</td>
<td>Microsoft</td>
<td>Larry Page</td>
</tr>
<tr>
<td>Samuel J. Palmisano</td>
<td>IBM</td>
<td>Werner Vogels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Google</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Amazon</td>
</tr>
</tbody>
</table>

Table 2: Examples of the analogical reasoning task for phrases (the full test set has 3218 examples). The goal is to compute the fourth phrase using the first three. Our best model achieved an accuracy of 72% on this dataset.
<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></td>
<td>Cecile De</td>
</tr>
</tbody>
</table>

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.
Word embeddings demo

http://bionlp-www.utu.fi/wv_demo/
Recurrent Neural Network (RNN)

Figure 4: A simple recurrent network.

From http://axon.cs.byu.edu/~martinez/classes/678/Slides/Recurrent.pptx
Recurrent Neural Network “unfolded” in time

Training algorithm: “Backpropagation in time”
Encoder-decoder (or “sequence-to-sequence”) networks for translation

http://book.paddlepaddle.org/08.machine_translation/image/encoder_decoder_en.png
Problem for RNNs: learning long-term dependencies.

“The cat that my mother’s sister took to Hawaii the year before last when you were in high school is now living with my cousin.”

Backpropagation through time: problem of vanishing gradients
Long Short Term Memory (LSTM)

- A “neuron” with a complicated memory gating structure.
- Replaces ordinary hidden neurons in RNNs.
- Designed to avoid the long-term dependency problem
Long-Short-Term-Memory (LSTM) Unit

Simple RNN (hidden) unit

LSTM (hidden) unit

From https://deeplearning4j.org/lstm.html
Comments on LSTMs

• LSTM unit replaces simple RNN unit

• LSTM internal weights still trained with backpropagation

• Cell value has feedback loop: can remember value indefinitely

• Function of gates ("input", "forget", "output") is learned via minimizing loss
Google “Neural Machine Translation”: (unfolded in time)

Neural Machine Translation:

Training: Maximum likelihood, using gradient descent on weights

$$\theta^* = \arg\max_{\theta} \sum_{X,Y} \log P(X \mid Y, \theta)$$

Trained on very large corpus of parallel texts in source ($X$) and target ($Y$) languages.
How to evaluate automated translations?

Human raters’ side-by-side comparisons: Scale of 0 to 6

0: “completely nonsense translation”

2: “the sentence preserves some of the meaning of the source sentence but misses significant parts”

4: “the sentence retains most of the meaning of the source sentence, but may have some grammar mistakes”

6: “perfect translation: the meaning of the translation is completely consistent with the source, and the grammar is correct.”
Results from Human Raters

Table 9: Human side-by-side evaluation scores of WMT En→Fr models.

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>Side-by-side averaged score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBMT [15]</td>
<td>37.0</td>
<td>3.87</td>
</tr>
<tr>
<td>NMT before RL</td>
<td>40.35</td>
<td>4.46</td>
</tr>
<tr>
<td>NMT after RL</td>
<td>41.16</td>
<td>4.44</td>
</tr>
<tr>
<td>Human</td>
<td>4.82</td>
<td></td>
</tr>
<tr>
<td>Translation</td>
<td>PBMT</td>
<td>GNMT</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>English → Spanish</td>
<td>4.885</td>
<td>5.428</td>
</tr>
<tr>
<td>English → French</td>
<td>4.932</td>
<td>5.295</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>4.035</td>
<td>4.594</td>
</tr>
<tr>
<td>Spanish → English</td>
<td>4.872</td>
<td>5.187</td>
</tr>
<tr>
<td>French → English</td>
<td>5.046</td>
<td>5.343</td>
</tr>
<tr>
<td>Chinese → English</td>
<td>3.694</td>
<td>4.263</td>
</tr>
</tbody>
</table>
Automating Image Captioning

a polar bear is standing in the snow

a baseball player swinging a bat at a ball
Automating Image Captioning

Training:
Large dataset of image/caption pairs from Flickr and other sources

Vinyals et al., “Show and Tell: A Neural Image Caption Generator”, CVPR 2015
“NeuralTalk” sample results

From http://cs.stanford.edu/people/karpathy/deepimagesent/generationdemo/
a man riding a skateboard down a sidewalk
logprob: -6.49

a bunch of bananas sitting on top of a wooden table
logprob: -8.52
a cat is sitting on a chair in front of a tv
logprob: -10.45

a table with a plate of food and a cup of coffee
logprob: -11.36
A young boy is holding a baseball bat.

logprob: -7.65
a man is sitting at a table with a laptop
logprob: -9.73

a dog is sitting on a bench next to a man
logprob: -13.87
Microsoft Captionbot

https://www.captionbot.ai/