Sharing is Caring in the Land of The Long Tail

Samy Bengio
“Real problems rarely come packaged as 1M images uniformly belonging to a set of 1000 classes…”
The long tail

• Well known phenomena where a small number of generic objects/entities/words appear very often and most others appear more rarely.
• Also known as Zipf or Power law, or Pareto distribution.
• The web is littered by this kind of distributions:
  • the frequency of each unique query on search engines,
  • the occurrences of each unique word in text documents,
  • etc.
Example of a long tail

Frequency of words in Wikipedia

- the
- anyways
- trickiest
- h-plane
Representation sharing

• How do we design a classifier or a ranker when data follows a long tail distribution?
• If we train one model per class, it is hard for poor classes to be well trained.
• How come we humans are able to recognize objects we have seen only once or even never?
• Most likely answer: representation sharing: all class models share/learn a joint representation.
• Poor classes can then benefit from knowledge learned from semantically similar but richer classes.
• Extreme case: zero-shot setting!
Outline

In this talk, I will cover the following ideas:

- Wsabie: a joint embedding space of images and labels
- The many facets of text embeddings
- Zero-shot setting through embeddings
- Incorporate Knowledge Graph constraints
- Use of a language model

I will NOT cover the following important issues:

- Prediction time issues for extreme classification
- Memory issues
Wsabie

Learn to embed images & labels to optimize top-ranked items.

Labels
Obama
Eiffel Tower
Shark
Dolphin
Lion
...

100-dimensional embedding space

Wsabie: J. Weston et al, ECML 2010, IJCAI 2011
Wsabie: summary

\[ \text{sim}(i, x) = \langle W_i, V_x \rangle \]

Label \( i \)  
\[
\begin{array}{c}
W \\
000100
\end{array}
\]

Image \( x \)  
\[
\begin{array}{c}
V \\
\text{real values}
\end{array}
\]

Triplet Loss: \( \text{sim}(\text{image, dolphin}) > \text{sim}(\text{image, obama}) + 1 \)

Trained by stochastic gradient descent and smart sampling of negative examples
## Wsabie: experiments - results

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet 2010</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>prec@1</td>
<td>prec@10</td>
</tr>
<tr>
<td>approx kNN</td>
<td>1.55%</td>
<td>0.41%</td>
</tr>
<tr>
<td>One-vs-Rest</td>
<td>2.27%</td>
<td>1.02%</td>
</tr>
<tr>
<td>Wsabie</td>
<td>4.03%</td>
<td>1.48%</td>
</tr>
<tr>
<td>Ensemble of 10 Wsabies</td>
<td>10.03%</td>
<td>3.02%</td>
</tr>
</tbody>
</table>

- **ImageNet 2010**: 16000 labels and 4M images
- **Web**: 109000 labels and 16M images
## Wsabie: embeddings

<table>
<thead>
<tr>
<th>Label</th>
<th>Nearest Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>barack obama</td>
<td>barak obama, obama, barack, barrack obama, bow wow</td>
</tr>
<tr>
<td>david beckham</td>
<td>beckham, david beckam, alessandro del piero, del piero</td>
</tr>
<tr>
<td>santa</td>
<td>santa claus, papa noel, pere noel, santa clause, joyeux noel</td>
</tr>
<tr>
<td>dolphin</td>
<td>delphin, dauphin, whale, delfin, delfini, baleine, blue whale</td>
</tr>
<tr>
<td>cows</td>
<td>cattle, shire, dairy cows, kuh, horse, cow, shire horse, kone</td>
</tr>
<tr>
<td>rose</td>
<td>rosen, hibiscus, rose flower, rosa, roze, pink rose, red rose</td>
</tr>
<tr>
<td>eiffel tower</td>
<td>eiffel, tour eiffel, la tour eiffel, big ben, paris, blue mosque</td>
</tr>
<tr>
<td>ipod</td>
<td>ipod, ipod nano, apple ipod, ipod apple, new ipod</td>
</tr>
<tr>
<td>f18</td>
<td>f 18, eurofighter, f14, fighter jet, tomcat, mig 21, f 16</td>
</tr>
</tbody>
</table>
### Wsabie: annotations

<table>
<thead>
<tr>
<th>Image</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Dolphins" /></td>
<td>delfini, orca, <strong>dolphin</strong>, mar, delfin, dauphin, whale, cancun, killer whale, sea world</td>
</tr>
<tr>
<td><img src="image2.png" alt="Whales" /></td>
<td>blue whale, whale shark, great white shark, underwater, white shark, shark, manta ray, <strong>dolphin</strong>, requin, blue shark, diving</td>
</tr>
<tr>
<td><img src="image3.png" alt="Obama" /></td>
<td>barrack obama, barak obama, barack hussein obama, <strong>barack obama</strong>, james marsden, jay z, obama, nelly, falco, barack</td>
</tr>
<tr>
<td><img src="image4.png" alt="Eiffel Tower" /></td>
<td>eiffel, paris by night, la tour eiffel, tour eiffel, <strong>eiffel tower</strong>, las vegas strip, eifel, tokyo tower, eiffel tower</td>
</tr>
</tbody>
</table>
“Why not an embedding of text only?”
Skip-Gram (Word2Vec)

Learn dense embedding vectors from an unannotated text corpus, e.g. Wikipedia

“an exceptionally large male tiger shark can grow up to”

http://code.google.com/p/word2vec

Tomas Mikolov, Kai Chen, Greg Corrado, Jeff Dean (ICLR 2013)
Skip-gram trained on Wikipedia, 155K terms

**tiger shark**
- bull shark
- blacktip shark
- shark
- oceanic whitetip shark
- sandbar shark
- dusky shark
- blue shark
- requiem shark
- great white shark
- lemon shark

**car**
- cars
- muscle car
- sports car
- compact car
- autocar
- automobile
- pickup truck
- racing car
- passenger car
- dealership

**t-SNE visualization of ImageNet labels**
- reptiles
- birds
- musical instruments
- insects
- clothing
- dogs
- aquatic life
- animals
- food
- transportation
Embeddings are powerful

\[ E(\text{Rome}) - E(\text{Italy}) + E(\text{Germany}) \approx E(\text{Berlin}) \]

\[ E(\text{hotter}) - E(\text{hot}) + E(\text{big}) \approx E(\text{bigger}) \]
Let’s go back to images!
Deep convolutional models for images

But what about the long tail of classes?

What about using our semantic embeddings for that?
ConSE: Convex Combination of Semantic Embeddings [Norouzi et al, ICLR’2014]
ConSE: Convex Combination of Semantic Embeddings

from Skip-Gram for instance:

\[ s(y) = \text{embedding position of } y \]

\[
f(x) = \sum_{i} p(y_i|x) s(y_i)
\]

\[
f(x) = p(\text{Lion}|x)s(\text{Lion}) + p(\text{Apple}|x)s(\text{Apple}) + p(\text{Orange}|x)s(\text{Orange}) + p(\text{Tiger}|x)s(\text{Tiger}) + p(\text{Bear}|x)s(\text{Bear})
\]

Do a nearest neighbor search around \( f(x) \) to find the corresponding label
ConSE(T): Convex Combination of Semantic Embeddings

In practice, consider the average of only a few labels:

\[ top(T) = \{ i \mid p(y_i|x) \text{ is among top } T \text{ probabilities} \} \]

\[ f(x) = \frac{1}{Z} \sum_{i \in top(T)} p(y_i|x)s(y_i) \]
ConSE(T): experiments on ImageNet

- Model trained with 1.2M ILSVRC 2012 images from 1,000 classes
- Evaluated on images from same classes.
- Results are measured as hit@$k$. 
ConSe(T) experiments

<table>
<thead>
<tr>
<th>Test Label Set</th>
<th># Candidate Labels</th>
<th>Model</th>
<th>Flat hit@k (%)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>2-hops</td>
<td>1,589</td>
<td>DeviSE</td>
<td>6.0</td>
<td>10.0</td>
<td>18.1</td>
<td>26.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE</td>
<td>9.3</td>
<td>14.4</td>
<td>23.7</td>
<td>30.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1)</td>
<td><strong>9.4</strong></td>
<td><strong>15.1</strong></td>
<td><strong>24.7</strong></td>
<td><strong>32.7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(10)</td>
<td>9.2</td>
<td>14.8</td>
<td>24.1</td>
<td>32.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-hops (+1K)</td>
<td>1,589 +1000</td>
<td>DeviSE</td>
<td>0.8</td>
<td>2.7</td>
<td>7.9</td>
<td>14.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE</td>
<td>0.2</td>
<td>7.1</td>
<td><strong>17.2</strong></td>
<td>24.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1)</td>
<td>0.3</td>
<td>6.2</td>
<td>17.0</td>
<td><strong>24.9</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(10)</td>
<td>0.3</td>
<td>6.2</td>
<td>16.7</td>
<td>24.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-hops</td>
<td>7,860</td>
<td>DeviSE</td>
<td>1.7</td>
<td>2.9</td>
<td>5.3</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE</td>
<td>2.6</td>
<td>4.2</td>
<td>7.3</td>
<td>10.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1)</td>
<td>2.7</td>
<td>4.4</td>
<td><strong>7.8</strong></td>
<td><strong>11.5</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(10)</td>
<td>2.6</td>
<td>4.3</td>
<td>7.6</td>
<td>11.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-hops (+1K)</td>
<td>7,860 +1000</td>
<td>DeviSE</td>
<td>0.5</td>
<td>1.4</td>
<td>3.4</td>
<td>5.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE</td>
<td>0.2</td>
<td>2.4</td>
<td><strong>5.9</strong></td>
<td>9.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1)</td>
<td>0.2</td>
<td>2.2</td>
<td><strong>5.9</strong></td>
<td><strong>9.7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(10)</td>
<td>0.2</td>
<td>2.2</td>
<td>5.8</td>
<td>9.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageNet 2011 21K</td>
<td>20,841</td>
<td>DeviSE</td>
<td>0.8</td>
<td>1.4</td>
<td>2.5</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE</td>
<td>1.3</td>
<td>2.1</td>
<td>3.6</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1)</td>
<td><strong>1.4</strong></td>
<td><strong>2.2</strong></td>
<td><strong>3.9</strong></td>
<td><strong>5.8</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(10)</td>
<td>1.3</td>
<td>2.1</td>
<td>3.8</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageNet 2011 21K (+1K)</td>
<td>20,841 +1000</td>
<td>DeviSE</td>
<td>0.3</td>
<td>0.8</td>
<td>1.9</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE</td>
<td>0.1</td>
<td>1.2</td>
<td>3.0</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1)</td>
<td>0.2</td>
<td>1.2</td>
<td>3.0</td>
<td><strong>5.0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(10)</td>
<td>0.2</td>
<td>1.2</td>
<td>3.0</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ConSE(1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Knowledge Graph
Multiclass Classifiers

GoogleLeNet model

Softmax

Logistic
Object labels have rich relations
Visual Model + Knowledge Graph

Hierarchy and Exclusion (HEX) Graph

[Deng et al, ECCV 2014]
HEX Classification Model

\[ x \in \mathbb{R}^n \]
Input scores

\[ y \in \{0,1\}^n \]
Binary Label vector

\[ \Pr(y | x) = \frac{1}{Z(x)} \prod_i \phi_i(x_i, y_i) \prod_{i,j} \psi_{i,j}(y_i, y_j) \]

Unary: same as logistic regression

\[ \phi_i(x_i, y_i) = \begin{cases} \text{sigmoid}(x_i) & \text{if } y_i = 1 \\ 1 - \text{sigmoid}(x_i) & \text{if } y_i = 0 \end{cases} \]

Pairwise: set illegal configuration to zero

\[ \psi_{i,j}(y_i, y_j) = \begin{cases} 0 & \text{If violates constraints} \\ 1 & \text{Otherwise} \end{cases} \]

All illegal configurations have probability zero.
Exp: Learning with weak labels

- ILSVRC 2012: “relabel” or “weaken” a portion of fine-grained leaf labels to basic level labels.
- Evaluate on fine-grained recognition

Original ILSVRC 2012 (leaf labels) | Training (“weakened” labels) | Test
Exp: Learning with weak labels

- ILSVRC 2012: “relabel” or “weaken” a portion of fine-grained leaf labels to basic level labels.
- Evaluate on fine-grained recognition.
- Consistently outperforms baselines.

<table>
<thead>
<tr>
<th>relabeling</th>
<th>softmax-leaf</th>
<th>softmax-all</th>
<th>logistic</th>
<th>ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>50.5(74.7)</td>
<td>56.4(79.6)</td>
<td>21.0(45.2)</td>
<td>58.2(80.8)</td>
</tr>
<tr>
<td>90%</td>
<td>26.2(47.3)</td>
<td>52.9(77.2)</td>
<td>9.3(27.2)</td>
<td>55.3(79.4)</td>
</tr>
<tr>
<td>95%</td>
<td>16.0(32.2)</td>
<td>50.8(76.0)</td>
<td>5.6(17.2)</td>
<td>52.4(77.2)</td>
</tr>
<tr>
<td>99%</td>
<td>2.5 (7.2)</td>
<td>41.5(68.1)</td>
<td>1.0(3.8)</td>
<td>41.5(68.5)</td>
</tr>
</tbody>
</table>

Top 1 accuracy (top 5 accuracy)
What about textual descriptions?

- We have considered the long tail of objects.
- What about more complex descriptions, involving multi-word descriptions, or captions?
- We can use language models to help.
Neural Image Caption Generator

[Vinyals et al, CVPR 2015]

1. Two pizzas sitting on top of a stove top oven.
2. A pizza sitting on top of a pan on top of a stove.

A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.
Let I be an image (pixels).
Let S be the corresponding sentence (sequence of words).
Likelihood of producing the right sentence given the image:

\[
\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1})
\]

We maximize the likelihood of producing the right sentence given the image:

\[
\theta^* = \arg \max_\theta \sum_{(I,S)} \log p(S|I; \theta)
\]
NIC: model

Image Embedding

P(word 1) P(word 2) P(<end>)

Convolution Neural Net

word 1 word N

Recurrent Neural Net
Two dogs play in the grass.

Two hockey players are fighting over the puck.

A skateboarder does a trick on a ramp.

A little girl in a pink hat is blowing bubbles.

A herd of elephants walking across a dry grass field.

A group of young people playing a game of frisbee.

A close up of a cat laying on a couch.

A red motorcycle parked on the side of the road.

A dog is jumping to catch a frisbee.

A yellow school bus parked in a parking lot.

A refrigerator filled with lots of food and drinks.
It doesn’t always work…

Human: A blue and black dress ... No! I see white and gold!

Our model: A close up of a vase with flowers.
Scheduled Sampling

\[ P(y_t|h_t) \text{ with } h_t = f(h_{t-1}, y_{t-1}) \]
Scheduled Sampling

\[ P(y_t|h_t) \text{ with } h_t = f(h_{t-1}, \hat{y}_{t-1}) \]
Scheduled Sampling

Exponential decay
Inverse sigmoid decay
Linear decay

Sample
Softmax over y(t-1)

Sample
Softmax over y(t)

h(1)

X

h(t-1)

h(t)

sampled y(t-2)
true y(t-2)

true y(t-1)
Conclusions

• The long tail problem happens in most of the interesting tasks.
• Sharing approaches can help “poor” classes to generalize thanks to “rich” classes.
• At the extreme, semantic embeddings can represent classes with zero training examples.
• Sharing approaches are interesting not only for text and images, but also for complete sentences.
• Other recent approaches: represent text using characters; good for long tail words.
• … but never under-estimate the long tail …
Open source machine learning

tensorflow.org
Google Brain Residency Programme

New one year immersion program in deep learning research

• Learn to conduct deep learning research w/experts in our team
• Fixed one-year employment with salary, benefits, ...
• Goal after one year is to have conducted several research projects
• Interesting problems, TensorFlow, and access to computational resources
• Apply before January 15, 2016.

• For more information: g.co/brainresidency
• Contact us: brain-residency@google.com