Deep Learning Applications in Natural Language Processing

Jindřich Libovický

December 5, 2018
Outline

Information Search

Unsupervised Dictionary Induction

Image Captioning
Information Search
**Task:** Find an answer for a question given question in a coherent text.

[Machine Comprehension](http://demo.allennlp.org/machine-comprehension)
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail. Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

**What causes precipitation to fall?**

gravity

**What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?**

graupel

**Where do water droplets collide with ice crystals to form precipitation?**

within a cloud

- best articles from Wikipedia, of reasonable size (23k paragraphs, 500 articles)
- crowd-sourced more than 100k question-answer pairs
- complex quality testing (which got estimate of single human doing the task)

https://rajpurkar.github.io/SQuAD-explorer/explore/1.1/dev/

Method Overview

1. Get text and question representation from
   - pre-trained word embeddings
   - character-level CNN

   ...using your favourite architecture.

2. Compute a similarity between all pairs of words in the text and in the question.

3. Collect all informations we have for each token.

4. Classify where the span is.

Representing Words

- pre-trained word embeddings
- concatenate with trained character-level representations
- character-level representations allow searching for out-of-vocabulary structured informations (numbers, addresses)
Contextual Embeddings Layer

- process both question and context with bidirectional LSTM layer → one state per word
- parameters are shared → representations share the space
Attention Flow

Captures affinity / similarity between pairs of question and context words.

\[ S_{ij} = w^T [h_i, c_j, h_i \odot c_j] \]
Context-to-query Attention

\[
\tilde{u}_1 \times \tilde{u}_2 \times \tilde{u}_3 \times \tilde{u}_4 \times \tilde{u}_5
\]

\[
\text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax} \quad \text{softmax}
\]

\[
\tilde{u}_1 \quad \tilde{u}_2 \quad \tilde{u}_3 \quad \tilde{u}_4 \quad \tilde{u}_5 \quad \tilde{u}_6 \quad \tilde{u}_7 \quad \tilde{u}_8 \quad \tilde{u}_9 \quad \tilde{u}_{10} \quad \tilde{u}_{11} \quad \tilde{u}_{12}
\]
Query-to-Context Attention

\[
\begin{align*}
&\text{query } U \\
&u_1 \quad u_2 \quad u_3 \\
&u_4 \quad u_5 \\
&\text{context } H \\
&h_1 \quad h_2 \quad h_3 \quad h_4 \\
&h_5 \quad h_6 \quad h_7 \quad h_8 \\
&h_9 \quad h_{10} \quad h_{11} \quad h_{12} \\
\end{align*}
\]

weighted sum

maximum
Modeling Layer

• concatenate: LSTM outputs for each context word, context-to-query-vectors
• copy query-to-context vector to each of them
• apply one non-linear layer and bidirectional LSTM
1. Start-token probabilities: project each state to scalar $\rightarrow$ apply softmax over the context

2. End-token probabilities:
   - Compute weighted average using the start-token probabilities $\rightarrow$ single vector
   - Concatenate the vector to each state
   - Project states to scalar, renormalize with softmax

3. At the end select the most probable span
Method Overview: Recap

Deep Learning Applications in Natural Language Processing
Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi’s Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the “golden anniversary” with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as “Super Bowl L”), so that the logo could prominently feature the Arabic numerals 50.
There are **13** natural reserves in Warsaw—among others, Bielany Forest, Kabaty Woods, Czerniaków Lake. About 15 kilometres (9 miles) from Warsaw, the Vistula river's environment changes strikingly and features a perfectly preserved ecosystem, with a habitat of animals that includes the otter, beaver and hundreds of bird species. There are also several lakes in Warsaw—mainly the oxbow lakes, like Czerniaków Lake, the lakes in the Łazienki or Wilanów Parks, Kamionek Lake. There are lots of small lakes in the parks, but only a few are permanent—the majority are emptied before winter to clean them of plants and sediments.
Replace LSTMs by dilated convolutions.

Dilation = 1

Dilation = 2
Convolutional Blocks
Using Pre-Trained Representations

Just replace the contextual embeddings with ELMo or BERT...
<table>
<thead>
<tr>
<th>method</th>
<th>Exact Match</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human performance</td>
<td>82.304</td>
<td>91.221</td>
</tr>
<tr>
<td>BiDAF with BERT</td>
<td>87.433</td>
<td>93.160</td>
</tr>
<tr>
<td>BiDAF with ELMo</td>
<td>81.003</td>
<td>87.432</td>
</tr>
<tr>
<td>BiDAF trained from scratch</td>
<td>73.744</td>
<td>81.525</td>
</tr>
</tbody>
</table>
Unsupervised Dictionary Induction
Task: Get a translation dictionary between two languages using monolingual data only.

- makes NLP accessible for low-resourced languages
- basic for unsupervised machine translation
- hot research topic (at least 10 research papers on this topic this year)

How it is done

1. Train word embeddings on large monolingual corpora.
2. Find a mapping between the two languages.

So far looks simple...
$X$, $Z$ embedding matrices for 2 languages.
Dictionary matrix $D_{ij} = 1$ if $X_i$ is translation of $Z_j$.

Supervised projection between embeddings
Given existing dictionary $D$ (small seed dictionary):

$$\arg\max_{W_Z, W_X} \sum_i \sum_j D_{ij} \cdot \text{similarity} (X_i; W_X, Z_j; W_Z) \left( X_i W_X (Z_j; W_Z)^T \right)$$

…but we need to find all $D$, $W_X$, and $W_Z$. 
Question: *How would you interpret this matrix?*

It is a table of similarities between pairs of words.
If the Vocabularies were Isometric...

- $M_X = XX^T$ and $M_Z = ZZ^T$ would only have permuted rows and columns
- if we sorted values in each row of $M_X$ and $M_Z$, corresponding words would have the same vectors

Let’s assume, it is true (at least approximately)

$$D_{i,:} \leftarrow 1 \left[ \arg\min_j (M_X)_{i,:}(M_Z)^T_{j,:} \right]$$

Assign nearest neighbor from the other language.

……in practice tragically bad but at least good initialization.
Iterate until convergence:

1. Optimize $W_Z$ and $W_X$, w.r.t to current dictionary

$$\arg\max_{W_Z, W_X} \sum_i \sum_j D_{ij} \cdot \left( X_i W_X (Z_j W_Z)^T \right)$$

2. Update dictionary matrix $D$

$$D_{ij} = \begin{cases} 1, & \text{if } i \text{ is nearest neighbor of } j \text{ or wise versa} \\ 0, & \text{otherwise} \end{cases}$$
### Accuracy on Large Dictionary

<table>
<thead>
<tr>
<th>Supervision</th>
<th>Method</th>
<th>EN-IT</th>
<th>EN-DE</th>
<th>EN-FI</th>
<th>EN-ES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mikolov et al. (2013)</td>
<td>34.93†</td>
<td>35.00†</td>
<td>25.91†</td>
<td>27.73†</td>
</tr>
<tr>
<td></td>
<td>Faruqui and Dyer (2014)</td>
<td>38.40*</td>
<td>37.13*</td>
<td>27.60*</td>
<td>26.80*</td>
</tr>
<tr>
<td></td>
<td>Shigeto et al. (2015)</td>
<td>41.53†</td>
<td>43.07†</td>
<td>31.04†</td>
<td>33.73†</td>
</tr>
<tr>
<td></td>
<td>Dinu et al. (2015)</td>
<td>37.7</td>
<td>38.93*</td>
<td>29.14*</td>
<td>30.40*</td>
</tr>
<tr>
<td>5k dict.</td>
<td>Lazaridou et al. (2015)</td>
<td>40.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Xing et al. (2015)</td>
<td>36.87†</td>
<td>41.27†</td>
<td>28.23†</td>
<td>31.20†</td>
</tr>
<tr>
<td></td>
<td>Zhang et al. (2016)</td>
<td>36.73†</td>
<td>40.80†</td>
<td>28.16†</td>
<td>31.07†</td>
</tr>
<tr>
<td></td>
<td>Artetxe et al. (2016)</td>
<td>39.27</td>
<td>41.87*</td>
<td>30.62*</td>
<td>31.40*</td>
</tr>
<tr>
<td></td>
<td>Artetxe et al. (2017)</td>
<td>39.67</td>
<td>40.87</td>
<td>28.72</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Smith et al. (2017)</td>
<td>43.1</td>
<td>43.33†</td>
<td>29.42†</td>
<td>35.13†</td>
</tr>
<tr>
<td></td>
<td>Artetxe et al. (2018a)</td>
<td>45.27</td>
<td>44.13</td>
<td><strong>32.94</strong></td>
<td>36.60</td>
</tr>
<tr>
<td>25 dict.</td>
<td>Artetxe et al. (2017)</td>
<td>37.27</td>
<td>39.60</td>
<td>28.16</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Smith et al. (2017), cognates</td>
<td>39.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Artetxe et al. (2017), num.</td>
<td>39.40</td>
<td>40.27</td>
<td>26.47</td>
<td>-</td>
</tr>
<tr>
<td>None</td>
<td>Zhang et al. (2017a), λ = 1</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
</tr>
<tr>
<td></td>
<td>Zhang et al. (2017a), λ = 10</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.01*</td>
<td>0.01*</td>
</tr>
<tr>
<td></td>
<td>Conneau et al. (2018), code‡</td>
<td>45.15*</td>
<td>46.83*</td>
<td>0.38*</td>
<td>35.38*</td>
</tr>
<tr>
<td></td>
<td>Conneau et al. (2018), paper‡</td>
<td>45.1</td>
<td>0.01*</td>
<td>0.01*</td>
<td>35.44*</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td><strong>48.13</strong></td>
<td><strong>48.19</strong></td>
<td>32.63</td>
<td><strong>37.33</strong></td>
</tr>
</tbody>
</table>
Try it yourself!

- Pre-train monolingual word embeddings using FastText / Word2Vec
- Install VecMap
  
  https://github.com/artetxem/vecmap

```bash
python3 map_embeddings.py --unsupervised SRC.EMB TRG.EMB SRC_MAPPED.EMB TRG_MAPPED.EMB
```
Image Captioning
Task: Generate a caption in natural language given an image.

Example:

A group of people wearing snowshoes, and dressed for winter hiking, is standing in front of a building that looks like it’s made of blocks of ice.

The people are quietly listening while the story of the ice cabin was explained to them.

A group of people standing in front of an igloo.

Several students waiting outside an igloo.
1. Obtain pre-trained image representation.
2. Use autoregressive decoder to generate the caption using the image representation.
2D Convolution over an Image

Basic method in deep learning for computer vision.

RGB image $9 \times 9 \times 3$

convolutional map $4 \times 4 \times 6$

stride 2
kernel size 3
filter size 6

Deep Learning Applications in Natural Language Processing
• Trained for 1k classes classification, millions of training examples
• Architecture: convolutions, max-pooling, residual connections, batch normalization, 50–150 layers
Reminder: Autoregressive Decoder

Deep Learning Applications in Natural Language Processing
Attention Model in Equations (1)

**Inputs:**
- decoder state $s_i$
- encoder states $h_j = [\overrightarrow{h_j}; \overrightarrow{h_j}]$ $\forall i = 1 \ldots T_x$

**Attention energies:**

$$e_{ij} = v_a^\top \tanh (W_a s_{i-1} + U_a h_j + b_a)$$

**Attention distribution:**

$$\alpha_{ij} = \frac{\exp (e_{ij})}{\sum_{k=1}^{T_x} \exp (e_{ik})}$$

**Context vector:**

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$
Attention Model in Equations (2)

Output projection:

\[ t_i = \text{MLP}(U_os_{i-1} + V_oE_yi-1 + C_oc_i + b_o) \]

...attention is mixed with the hidden state

Output distribution:

\[ p(y_i = k|s_i, y_{i-1}, c_i) \propto \exp(W_ot_i)_k + b_k \]
Example Outputs: Correct

A woman is throwing a \textbf{frisbee} in a park.

A \textbf{dog} is standing on a hardwood floor.

A \textbf{stop} sign is on a road with a mountain in the background.

A little \textbf{girl} sitting on a bed with a teddy bear.

A group of \textbf{people} sitting on a boat in the water.

A \textbf{giraffe} standing in a forest with \textbf{trees} in the background.
Example Outputs: Incorrect

A large white bird standing in a forest.
A woman holding a clock in her hand.
A man wearing a hat and a hat on a skateboard.

A person is standing on a beach with a surfboard.
A woman is sitting at a table with a large pizza.
A man is talking on his cell phone while another man watches.
Employing Transformer Decoder

- **Input Embeddings** ⊕ **Position Encoding**
- **Self-Attentive Sublayer**
  - Multihead Attention
  - Keys & Values
  - Queries
  - ⊕ **Layer Normalization**
- **Cross-Attention Sublayer**
  - Multihead Attention
  - Keys & Values
  - Queries
  - ⊕ **Layer Normalization**
- **Encoder**
  - **Feed-Forward Sublayer**
    - Non-linear Layer
    - Linear Layer
    - ⊕ **Layer Normalization**
- **Output Symbol Probabilities**
  - Softmax
  - Linear

Deep Learning Applications in Natural Language Processing
## Quantitative Results

<table>
<thead>
<tr>
<th>Model (Variation)</th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN + attention (original)</td>
<td>24.3</td>
</tr>
<tr>
<td>RNN + attention (with better image representation)</td>
<td>32.6</td>
</tr>
<tr>
<td>Transformer (with better image representation)</td>
<td>33.3</td>
</tr>
</tbody>
</table>