Recent Advances in Transferable Representation Learning

The Basics of Embeddings

Muhao Chen, Kai-Wei Chang, Dan Roth

AAAI 2020 Tutorial
Machine Learning Pipelines

Traditional ML framework

Input → Feature Extractor → ML Model → Inference Alg. → Evaluation

Features

Sub-scores
(e.g., CRF potentials)
Machine Learning Pipelines

Traditional ML framework

Input → Feature Extractor → ML Model → Inference Alg. → Evaluation

Feature Engineering
Machine Learning Pipelines

Traditional ML Framework

Input ➔ Feature Extractor ➔ ML Model ➔ Inference Alg. ➔ Evaluation

Representation Learning

Deep Learning Framework

Input ➔ DNN ➔ Evaluation

Representation ➔ Inference
Packet switching contrasts with another principal networking paradigm, circuit switching, a method which pre-allocates dedicated network bandwidth specifically for each communication session, each having a constant bit rate and latency between nodes. In cases of billable services, such as cellular communication services, circuit switching is characterized by a fee per unit of connection time, even when no data is transferred, while packet switching may be characterized by a fee per unit of information transmitted, such as characters, packets, or messages.

Q: Packet Switching contrast with what other principal
A: circuit switching
A History of Word Representation
How to Represent Words?

Credit: https://www.flickr.com/photos/18229932@N07/48688109908
Idea 1: Representing Words as Discrete Symbols

- Represent word as a “one-hot” vector, e.g.,

\[
\text{happy} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \ldots & 0 \end{bmatrix}^T
\]

\[
\text{glad} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \ldots & 0 \end{bmatrix}^T
\]

- How large is this vector?
  - Vector dimension = number of words in vocabulary
    - PTB data: ~50k
    - Google 1T data: 13M
- Issue: no notion of similarity.

\[\text{happy} \text{ and } \text{glad} \text{ are orthogonal.}\]
Idea 2: Similarity = Clustering

- Dictionary: e.g., WordNet *(Miller 1995)*
Idea 3: Vector Representations

- Discrete ⇒ continuous: A dense vector for each word
- Words with similar meaning are closer in the embedding space
- Word meanings are vector of “basic concepts”
  - The “basic concepts” might not be explicit
  
  \[
  v_{king} = [0.8, 0.9, 0.1, 0, \ldots] \\
  v_{queen} = [0.8, 0.1, 0.8, 0, \ldots] \\
  v_{apply} = [0.1, 0.2, 0.1, 0.8, \ldots]
  \]

- Difference between word vectors captures their relations
  [Mikolov+ 13, Pennington+ 14]
How to Learn word Vectors

**Distributional** hypothesis:
“You shall know a word by the company it keeps” (J. R. Firth 1957: 11)

linguistic items with similar distributions have similar meanings.
How to Learn word Vectors

- Learn word representations based on co-occurrences
- E.g., Word2vec [Mikolov et al. 2013]

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

Slide credit: Stanford cs224n
Continuous representations for entities

Embeddings can be learned from Freebase, Dbpedia, YAGO, NELL, etc.
Contextualized Word Representations

- Most words have multiple meanings
- Can we encode word also based on the surrounding contexts?

He taught himself to **play** the violin.

Do you enjoy the **play**?

Word Embedding

Contextualized Word Embedding

Embedding visualization

from context\(^1\)

from context\(^2\)
Embeddings from Language Models (ELMo)

Deep contextualized word representations (Peters+2018)

Train Separate Left-to-Right and Right-to-Left LMs

Apply as “Pre-trained Embeddings”

Learning transferable representations using language model objective.
Performance Boost with ELMo

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Previous SOTA</th>
<th>Baseline</th>
<th>Performance boost with ELMo</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNLI</td>
<td>88.6</td>
<td>86.0</td>
<td>88.7 (+5.8%)</td>
</tr>
<tr>
<td>NER</td>
<td>91.9</td>
<td>90.2</td>
<td>92.2 (+21%)</td>
</tr>
<tr>
<td>SQuAD</td>
<td>84.4</td>
<td>81.1</td>
<td>85.8 (+25%)</td>
</tr>
<tr>
<td>Coref</td>
<td>81.7</td>
<td>67.2</td>
<td>70.4 (+9.9%)</td>
</tr>
<tr>
<td>SRL</td>
<td>81.4</td>
<td>53.7</td>
<td>84.6 (+17%)</td>
</tr>
<tr>
<td>SST-5</td>
<td></td>
<td></td>
<td>54.7 (+6.8%)</td>
</tr>
</tbody>
</table>
How to jointly capture the context information from both directions?

**Unidirectional context**
Build representation incrementally

**Bidirectional context**
Words can “see themselves”
Masked Language Model

- How to jointly capture the context information from both directions?

Unidirectional context
Build representation incrementally

Bidirectional context
Words can “see themselves”

- Masked Language Model: Mask out k% of the input words

store
the man went to the [MASK] to buy a [MASK] of milk
gallon
Fine-Tuning Procedure