A Neural Language Model for Dynamically Representing the Meanings of Unknown Words and Entities in a Discourse

Sosuke Kobayashi

Naoaki Okazaki

Kentaro Inui

Tokyo Institute of Technology

Tohoku University / RIKEN
RNN Language Model

- **Output matrix**: calculate probability
- **RNN matrices**: encoding context
- **Embedding matrix**: represent word meaning
Neural Language Model

- Embedding matrix
- Output matrix

- Cannot cover all words
  → Unknown words

- Referents differ by discourses
  → Unknown entities
Neural Language Model

- Embedding matrix
- Output matrix
- Cannot cover all words → Unknown words
- Referents differ by discourses → Unknown entities
Dynamic Entity Representation

• Unknown’s meaning representation cannot be obtained statically...
  ↓
  Dynamically update meaning representation while reading text

• Infer on-the-fly meanings from context

  • “she contracted mumps” → mumps is a disease?
  • “John loves Fender” → “John” is a guitarist?
Usage: Input Embedding

- Language models encode context words and predict next words
- Input word embeddings can be replaced
- Dynamic modeling makes context informative

- “... with him, John played [???]”
- with dynamic model: “... with him, <John; guitarist> played [???]”
Usage: Output Matrix

- Language models encode context words and predict next words
- Output matrix’s rows can be replaced
- Dynamic modeling makes target informative

- “... she is a big fan of [???]”
  - John? Mary?
- with dynamic model:
  - “... she is a big fan of [???]”
  - <guitarist>? <mother>?
Recipe: Context Encoding

- Encode context of the target word
- e.g. bi-directional RNN
Recipe: Context Encoding

- Encode context of the target word
- e.g. bi-directional RNN
Recipe: Context Merging

- Merge multiple contexts where the target occurs
  - e.g. RNN, max-pooling
Recipe: Context Merging

- Merge multiple contexts where the target occurs
- e.g. RNN, max-pooling
Recipe: Context Merging

- Merge multiple contexts where the target occurs
- e.g. RNN, max-pooling
Dataset for Evaluation

- Dataset for language modeling from OnteNotes
- Coreferents are unified and anonymized

RAW

John loves guitars.
Mary did not prefer music.
But, many people are big fans of him. ...

OURS

[UNK1] loves guitars.
[UNK2] did not prefer music.
But, many people are big fans of [UNK1]. ...
Result: Language Modeling

- Dynamic modeling improves perplexity
- Especially when entities reappear

<table>
<thead>
<tr>
<th></th>
<th>All tokens</th>
<th>Reappearing entities</th>
<th>Tokens following them</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>64.8</td>
<td>48.0</td>
<td>128.6</td>
</tr>
<tr>
<td>+Dynamic</td>
<td>60.7</td>
<td>34.0</td>
<td>106.8</td>
</tr>
<tr>
<td>input only</td>
<td>62.8</td>
<td>42.4</td>
<td>109.5</td>
</tr>
<tr>
<td>output only</td>
<td>62.5</td>
<td>35.9</td>
<td>129.0</td>
</tr>
<tr>
<td>input &amp; output</td>
<td>60.7</td>
<td>34.0</td>
<td>106.8</td>
</tr>
</tbody>
</table>
Result: Language Modeling

- Dynamic modeling improves perplexity
- Especially when entities reappear

<table>
<thead>
<tr>
<th></th>
<th>All tokens</th>
<th>Reappearing entities</th>
<th>Tokens following them</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>64.8</td>
<td>48.0</td>
<td>128.6</td>
</tr>
<tr>
<td>input only</td>
<td>62.8</td>
<td>42.4</td>
<td>109.5</td>
</tr>
<tr>
<td>output only</td>
<td>62.5</td>
<td>35.9</td>
<td>129.0</td>
</tr>
<tr>
<td>input &amp; output</td>
<td>60.7</td>
<td>34.0</td>
<td>106.8</td>
</tr>
</tbody>
</table>

“... she is a big fan of [???]”
<John; guitarist>
### Result: Language Modeling

- Dynamic modeling improves perplexity
- Especially when entities reappear

<table>
<thead>
<tr>
<th></th>
<th>All tokens</th>
<th>Reappearing entities</th>
<th>Tokens following them</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>64.8</td>
<td>48.0</td>
<td>128.6</td>
</tr>
<tr>
<td><strong>input only</strong></td>
<td>62.8</td>
<td>42.4</td>
<td>109.5</td>
</tr>
<tr>
<td><strong>output only</strong></td>
<td>62.5</td>
<td>35.9</td>
<td>129.0</td>
</tr>
<tr>
<td><strong>input &amp; output</strong></td>
<td>60.7</td>
<td>34.0</td>
<td>106.8</td>
</tr>
</tbody>
</table>

"… <John; guitarist> [???]"
Result: Language Modeling

- Dynamic modeling improves perplexity
- Especially when entities reappear

<table>
<thead>
<tr>
<th></th>
<th>All tokens</th>
<th>Reappearing entities</th>
<th>Tokens following them</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>64.8</td>
<td>48.0</td>
<td>128.6</td>
</tr>
<tr>
<td>Input only</td>
<td>62.8</td>
<td>42.4</td>
<td>109.5</td>
</tr>
<tr>
<td>Output only</td>
<td>62.5</td>
<td>35.9</td>
<td>129.0</td>
</tr>
<tr>
<td>Input &amp; output</td>
<td>60.7</td>
<td>34.0</td>
<td>106.8</td>
</tr>
</tbody>
</table>

+Dynamic
Result: Language Modeling

- Dynamic modeling works well for long documents. The latter of a document, the more often targets occur, the more improved.
- The more targets occur,

- Organizing context is useful for long documents
Summary

• Dynamic modeling of word vectors improves language models
  • For prediction of the unknowns
  • For prediction of tokens following the unknowns

• Future work
  • Story generation with organizing entities
  • Joint modeling with coreference resolution
  • Joint modeling with character/subword vectors
Result: Cloze Test

- Pseudo coreference resolution task
- Solve this task by calculating the sentence likelihood by filling in with each entity

\[
\begin{align*}
[\text{UNK1}] & \text{ loves guitars.} \\
[\text{UNK2}] & \text{ did not prefer music.} \\
\text{But, many people are big fans of } & \text{[???]}. \ldots
\end{align*}
\]

- Mean Quantile (mean rank of answers) is improved \( .525 \rightarrow .642 \) by dynamic modeling