Overview

● Task:
  ○ Machine Reading Comprehension: Given a question and a piece of text, find the answer in the text

● Contributions:
  ○ Using external knowledge for MRC
  ○ Incorporate the structure of the knowledge base into the model
  ○ Extract sub-graphs from the external knowledge base

● Why this paper?
  ○ An example of how we can use external knowledges, especially the structure of an external knowledge base, to improve the representations of words for our tasks
Model
Model - continue

- Word representations using BERT:

\[
[CLS] \text{Question} [SEP] \text{Paragraph} [SEP]
\]

- Knowledge sub-graph construction:
  - For each word in paragraph, find all triples that their head or tail is identical to the lemma of the word, e.g., (shortage, related_to, lack)
  - For each triple found in the previous step, add the neighbors that have identical lemma with one of the words in the question, e.g.: (need, related_to, lack)
  - Merge the extracted triples with the same head or tail to create a connected sub-graph, e.g., (shortage, related_to, lack, related_to, need)
Model - continue

- Update word representations using GAT on the sub-graphs of each word:

\[ h_{j}^{l+1} = \sum_{n=1}^{N_j} \alpha_n t_n \]

\[ \alpha_n = \frac{\exp(\beta_n)}{\sum_{j=1}^{N_j} \exp(\beta_j)} \]

\[ \beta_n = (W_r r_n) \top \tanh \left( W_h h_n + W_i t_n \right) \]

- Combine BERT and knowledge-based representations of the words:

\[ w_i = \sigma(\langle W [t_{b_i}, t_{k_i}] \rangle) \]

\[ t'_i = w_i \odot t_{b_i} + (1 - w_i) \odot t_{k_i} \]
Results

- External knowledge base:
  - WordNet
  - ConceptNet
- Entity embedding extracted using OpenKE

<table>
<thead>
<tr>
<th>Model</th>
<th>EM Dev</th>
<th>EM Test</th>
<th>F1 Dev</th>
<th>F1 Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>QANet (Yu et al., 2018)</td>
<td>35.38</td>
<td>36.51</td>
<td>36.75</td>
<td>37.79</td>
</tr>
<tr>
<td>SAN (Liu et al., 2018)</td>
<td>38.14</td>
<td>39.77</td>
<td>39.09</td>
<td>40.72</td>
</tr>
<tr>
<td>DocQA w/o ELMo (Clark and Gardner, 2018)</td>
<td>36.59</td>
<td>38.52</td>
<td>37.89</td>
<td>39.76</td>
</tr>
<tr>
<td>DocQA w/ ELMo (Clark and Gardner, 2018)</td>
<td>44.13</td>
<td>45.44</td>
<td>45.39</td>
<td>46.65</td>
</tr>
<tr>
<td>SKG+BERT-Large(ours)</td>
<td>70.94</td>
<td>72.24</td>
<td>71.55</td>
<td>72.78</td>
</tr>
</tbody>
</table>
CaRe: Open Knowledge Graph Embeddings
Overview

- **Task:**
  - Learn embeddings of the nodes and relations in knowledge base

- **Contributions:**
  - Introduce a method for encoding nodes and edges in an open knowledge base
  - Augmenting the open knowledge graph with the canonical knowledge graph
  - Robust to errors in the canonical knowledge graph

- **Why this paper:**
  - An example to show how to encode the nodes and the edges in knowledge graphs
  - How to overcome the multiple surface forms for entities and relations in open knowledge graph
Model Overview

- Avoid directly converting the knowledge base into a canonical form as it propagates errors in the canonicalization to the final knowledge base.
- Augment the original knowledge base by finding the nodes that refer to the same entity and add a bidirectional edge between them.
- Use pre-trained word embedding to encode the similarity of the relations with different surface forms.
- Use common knowledge base encoders to embed the nodes and the relations in the augmented open knowledge base.
Augment with canonical edges

- Nodes that refer to the same entity are connected with new bidirectional edges:
Relation and Node Representation

- Represent words using word embedding of the words in their surface form, then feed them into a bidirectional GRU:

\[
(\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_T}) = \overrightarrow{GRU}(x_1, x_2, \ldots, x_T)
\]

\[
(\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_T}) = \overleftarrow{GRU}(x_1, x_2, \ldots, x_T)
\]

- Merge representations of the nodes that are connected in the canonical form:

\[
e^c_n = \left( \sum_{i \in \mathcal{N}(n)} \frac{1}{|\mathcal{N}(n)|} e_i \right), \forall n \in \mathbb{N}
\]

\[
e'_n = \frac{e_n}{2} + \frac{e^c_n}{2}
\]
Update Node and Relation Rep.

- Use one of the existing knowledge base encoder to update the representations of the nodes and relations:
  - TransE:
    \[ \psi(s, r, o) = -\|e_s + r_r - e_o\|_p \]
  - ConvE:
    \[ \psi(s, r, o) = f(vec(f([e_s; r_r] * w))W)e_o \]
  - GCN:
    \[ e_n^{l+1} = f \left( \sum_{i \in \mathcal{N}(n)} (W^l e_i^l + b^l) \right), \forall n \in \mathbb{N} \]
  - GAT:
    \[ e_n^{l+1} = \|_{k=1}^{K} f \left( \sum_{i \in \mathcal{N}(n)} (\alpha_k(e_n^l, e_i^l)W_k e_i^l) \right) \]
Results

- GloVe embedding
- Find canonical clusters using CESI
- Link Prediction Performance:

<table>
<thead>
<tr>
<th>Method</th>
<th>ReVerb45K</th>
<th>ReVerb20K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MR</td>
<td>MRR</td>
</tr>
<tr>
<td>TransE</td>
<td>2955.8</td>
<td>.193</td>
</tr>
<tr>
<td>TransH</td>
<td>2998.2</td>
<td>.194</td>
</tr>
<tr>
<td>DistMult</td>
<td>8988.8</td>
<td>.051</td>
</tr>
<tr>
<td>ComEx</td>
<td>7786.5</td>
<td>.047</td>
</tr>
<tr>
<td>R-GCN</td>
<td>2866.8</td>
<td>.042</td>
</tr>
<tr>
<td>ConvE</td>
<td>2650.8</td>
<td>.233</td>
</tr>
<tr>
<td>CaRe(B=ConvE)</td>
<td>1308.0</td>
<td>.324</td>
</tr>
</tbody>
</table>
Questions?
Thanks