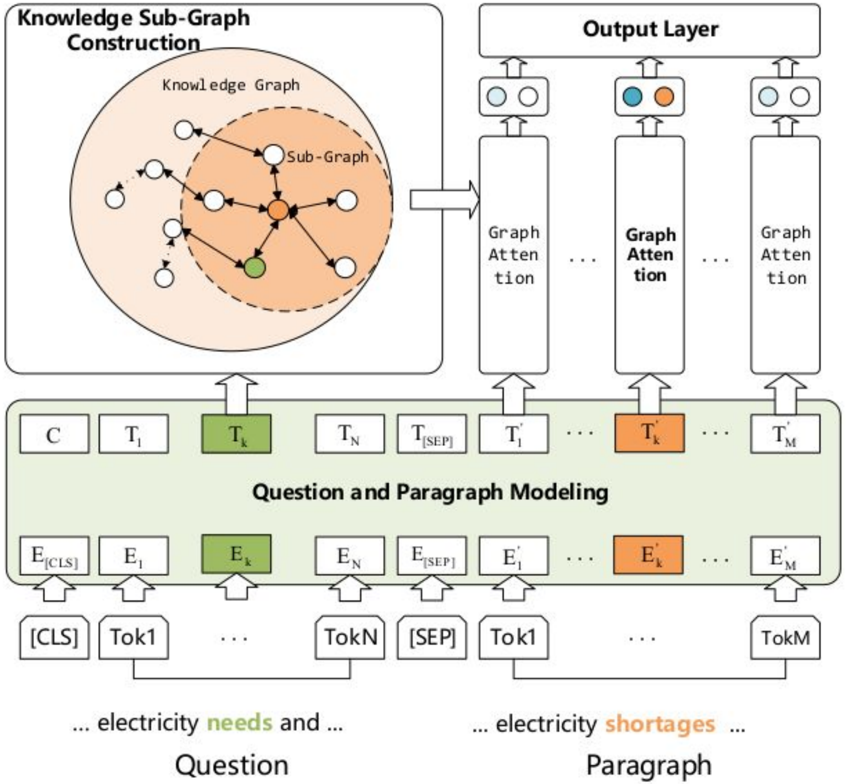
Machine Reading Comprehension Using Structural Knowledge Graph-aware Network

EMNLP 2019

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



Knowledge Sub-Graph Construction
Knowledge Graph
Sub-Graph
Output Layer
Graph Attention
Graph Attention
Graph Attention
C
T1
Tk
TN
T[SEP]
T'1
T'k
T'M
Question and Paragraph Modeling
E[CLS]
E1
Ek
EN
E[SEP]
E'1
E'k
E'M
[CLS]
Tok1
TokN
[SEP]
Tok1
TokM
... electricity needs and ...
Question
... electricity shortages ...
Paragraph

Model - continue

- Word representations using BERT:

$$[CLS]Question[SEP]Paragraph[SEP]$$

- 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:

$$\boldsymbol{h}_j^{l+1} = \sum_{n=1}^{N_j} \alpha_n \boldsymbol{t}_n^l$$

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

$$\beta_n = \left(\mathbf{W}_r^l \boldsymbol{r}_n^l\right)^\top \tanh\left(\mathbf{W}_h^l \boldsymbol{h}_n^l + \mathbf{W}_t^l \boldsymbol{t}_n^l\right)$$

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

$$w_i = \sigma((W\,[\boldsymbol{t}_{b_i}; \boldsymbol{t}_{k_i}]))$$

$$\boldsymbol{t'}_i = w_i \odot \boldsymbol{t}_{b_i} + (1 - w_i) \odot \boldsymbol{t}_{k_i}$$

Results

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

Model	EM		F1	
	Dev	Test	Dev	Test
QANet (Yu et al., 2018)	35.38	36.51	36.75	37.79
SAN (Liu et al., 2018)	38.14	39.77	39.09	40.72
DocQA w/o ELMo (Clark and Gardner, 2018)	36.59	38.52	37.89	39.76
DocQA w/ ELMo (Clark and Gardner, 2018)	44.13	45.44	45.39	46.65
SKG+BERT-Large(ours)	**70.94**	**72.24**	**71.55**	**72.78**

CaRe: Open Knowledge Graph Embeddings

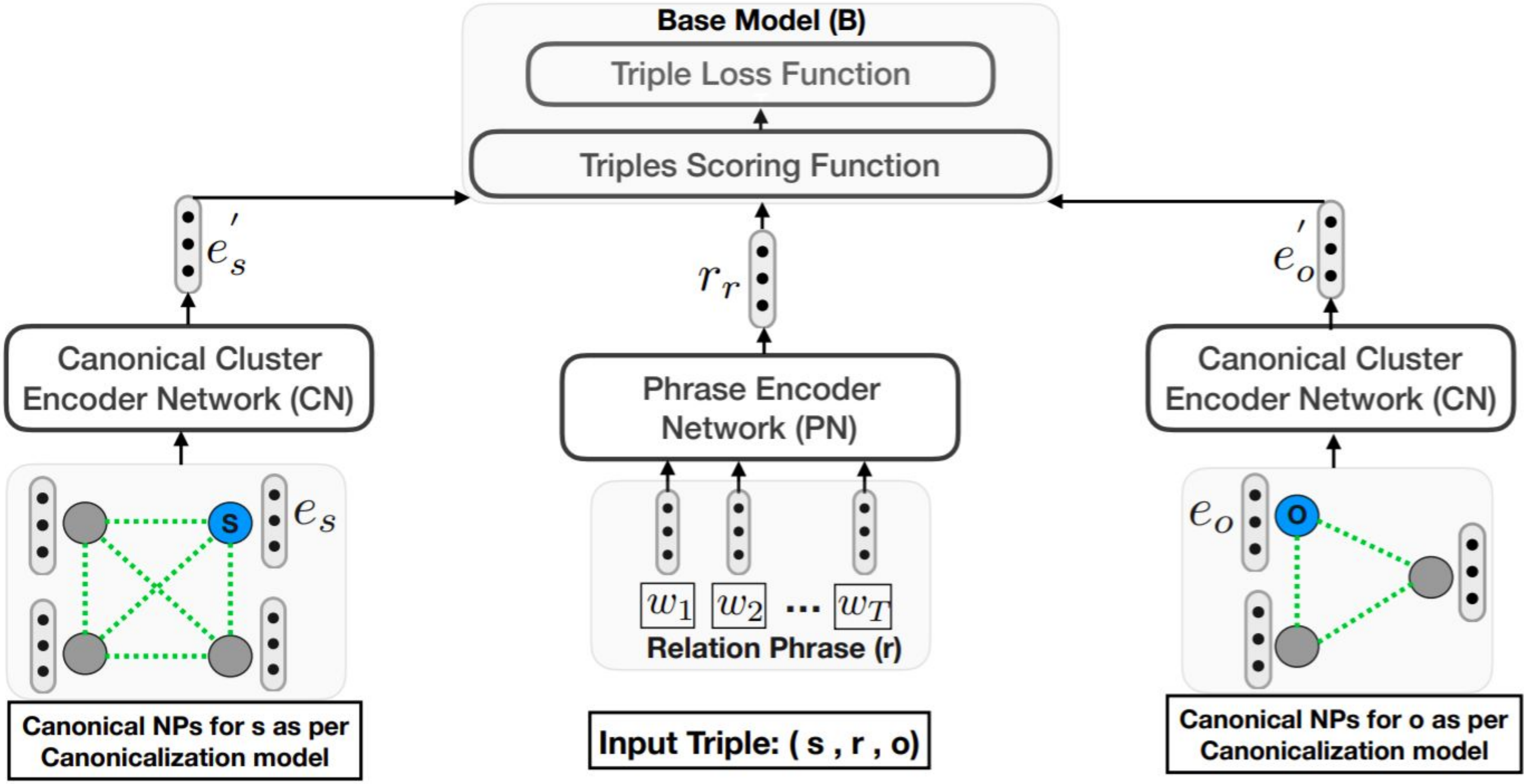
EMNLP 2019

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

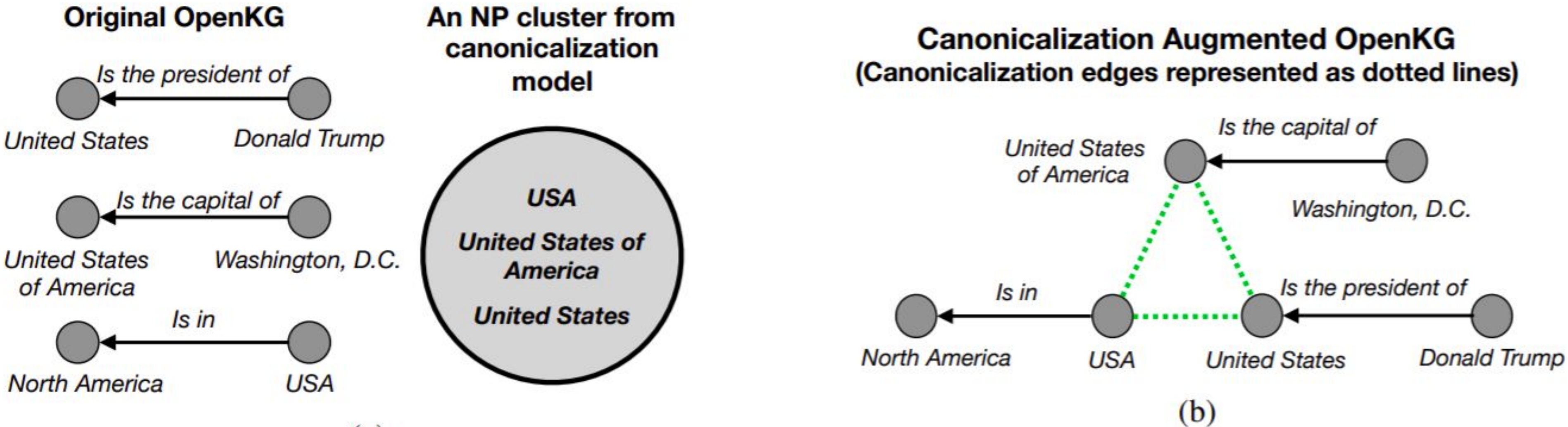
Model Overview

- 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


Base Model (B)
Triple Loss Function
Triples Scoring Function
e_s'
r_r
e_o'
Canonical Cluster Encoder Network (CN)
Phrase Encoder Network (PN)
Canonical Cluster Encoder Network (CN)
e_s
s
w_1 w_2 ... w_T
Relation Phrase (r)
e_o
o
Canonical NPs for s as per Canonicalization model
Input Triple: (s , r , o)
Canonical NPs for o as per Canonicalization model

Augment with canonical edges

- Nodes that refer to the same entity are connected with new bidirectional edges:



(a)

(b)

Relation and Node Representation

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

$$(\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_T}) = \overrightarrow{GRU}(x_1, x_2, ..., x_T)$$
$$(\overleftarrow{h_1}, \overleftarrow{h_2}, ..., \overleftarrow{h_T}) = \overleftarrow{GRU}(x_1, x_2, ..., x_T)$$

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

$$e_n^c = \left(\sum_{i \in \mathcal{N}(n)} \frac{1}{|\mathcal{N}(n)|} e_i \right), \forall n \in \mathbf{N}.$$

$$e_n^{'} = \frac{e_n}{2} + \frac{e_n^c}{2}$$

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:

$$\psi(s, r, o) = f(vec(f([\bar{e_s}; \bar{r_r}] * w))W)e_o$$

- GCN:

$$e_n^{l+1} = f\left(\sum_{i \in \mathcal{N}(n)} \left(W^l e_i^l + b^l\right)\right), \forall n \in \mathbf{N}$$

- GAT:

$$e_n^{l+1} = ||_{k=1}^{K} f\left(\sum_{i \in \mathcal{N}(n)} \left(\alpha_k(e_n^l, e_i^l) W_k^l e_i^l\right)\right)$$

Results

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

Method	ReVerb45K					ReVerb20K				
	MR	MRR	Hits@10	Hits@30	Hits@50	MR	MRR	Hits@10	Hits@30	Hits@50
TransE	2955.8	.193	.361	.446	.478	1425.8	.126	.299	.411	.468
TransH	2998.2	.194	.362	.442	.478	1464.4	.129	.303	.409	.467
DistMult	8988.8	.051	.051	.052	.065	6260.0	.033	.044	.055	.060
ComlEx	7786.5	.047	.047	.048	.073	5502.2	.037	.058	.075	.085
R-GCN	2866.8	.042	.046	.091	.113	1204.3	.122	.187	.263	.305
ConvE	2650.8	.233	.338	.401	.429	1014.5	.294	.402	.491	.541
CaRe(B=ConvE)	**1308.0**	**.324**	**.456**	**.543**	**.579**	**973.2**	**.318**	**.439**	**.525**	**.566**

Questions?
Thanks