GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction

Tsu-Jui Fu, Peng-Hsuan Li, Wei-Yun Ma

Xuehan Chen
10/14/2021

LEHIGH UNIVERSITY

Background

LEHIGH UNIVERSITY

- Entity-Relation extraction:

Taking into account the interaction between relations, especially for overlapping relations

- Three categories of relation tripets:
    - *Normal pair:*

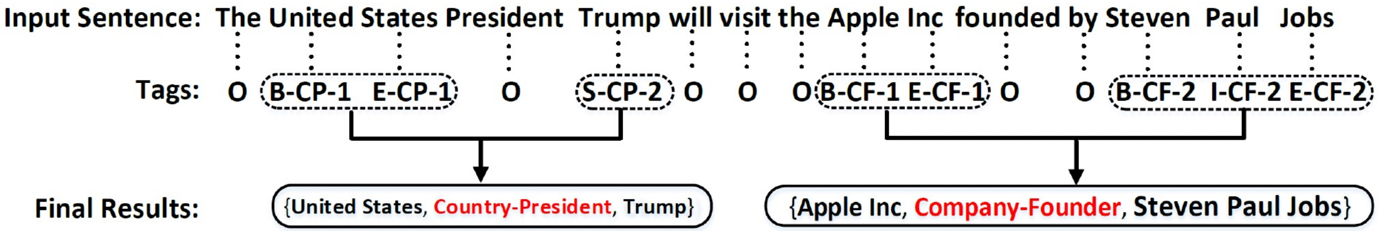
      (BarackObama, PresidentOf, UnitedStates)
    - *EntityPairOverlap(EPO):*

      (BarackObama, PresidentOf, UnitedStates)
      (BarackObama, Governance, UnitedStates)
    - *SingleEntityOverlap(SEO):*

      (BarackObama, LiveIn, WhiteHouse)
      (WhiteHouse, PresidentialPalace, UnitedStates)

Novel Tagging (Zheng et al., 2017)

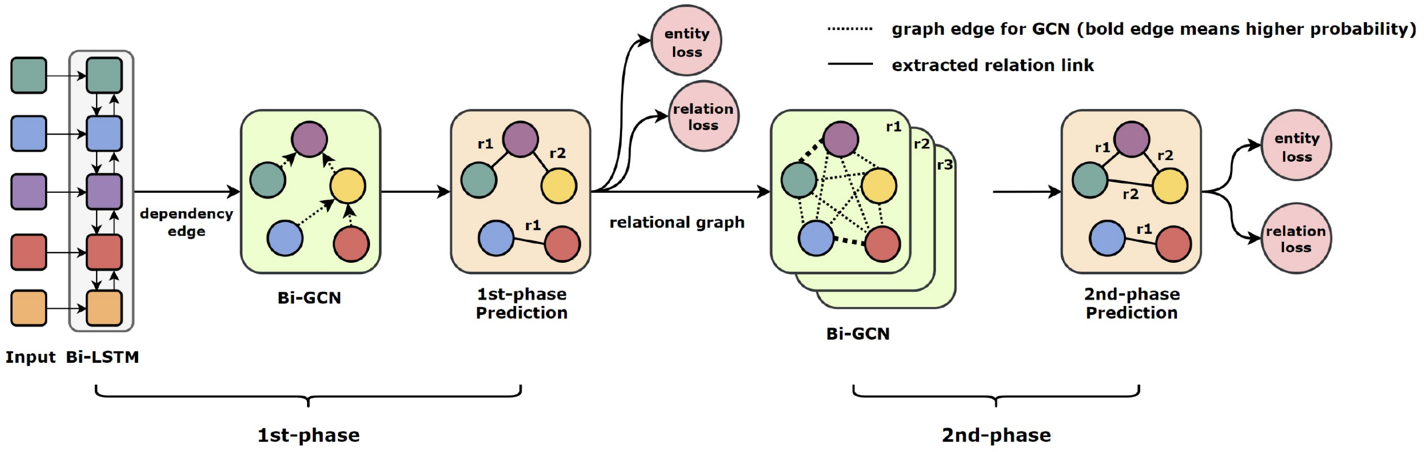
Proposed a strong neural end-to-end joint model of entities and relations based on an LSTM sequence tagger.



Not consider overlapping relations

Word Position
B: begin
S: single
E: end
I : inside
O: others

GraphRel



- GraphRel learns to automatically extract hidden features for each word by stacking a Bi-LSTM sentence encoder and a GCN dependency tree encoder.
- Novel 2nd-phase relation-weighted GCN for further extraction

GraphRel 1st-phase

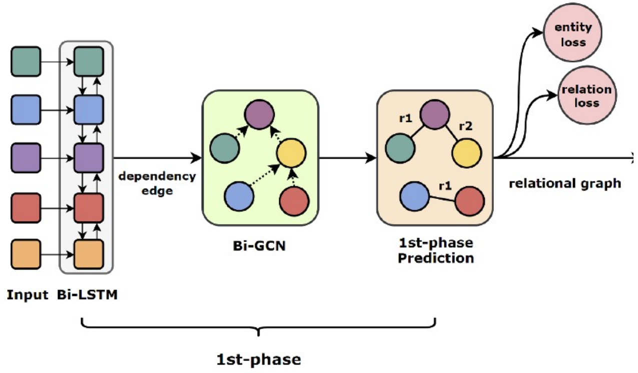
1st-phase Bi-LSTM +(usual) GCN:

- Bi-LSTM: extract sequential dependency word features

$$h_u^0 = \mathrm{Word}(u) \oplus POS(u)$$

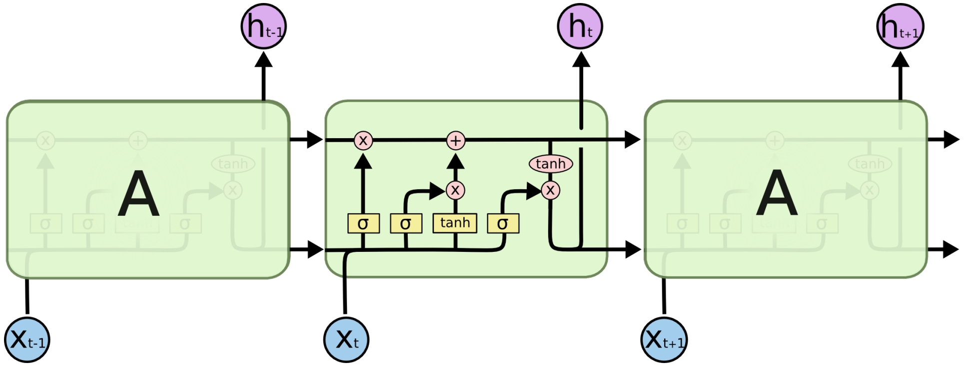
- GCN: extract regional dependency word features

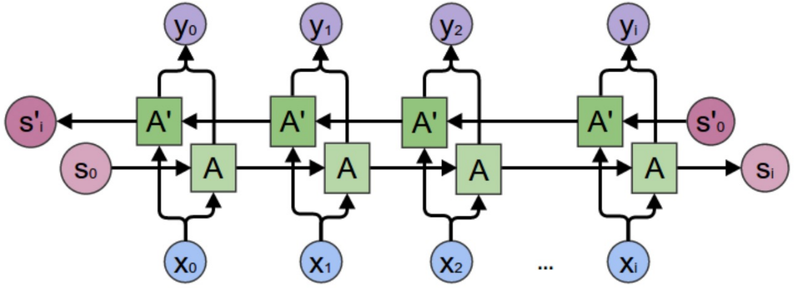
$$h_u^{l+1} = \mathrm{Re}\,LU\left(\sum_{v \in N(u)} \left(W^l h_v^l + b^l\right)\right)$$



- Given the word features, predict relations for each word pair and the entities for all words.

Bi-LSTM (Long Short-Term Memory)





LSTM has three gates

- forget gate layer

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- input gate layer

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- update the old cell state

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- output gate layer

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

Bi-GCN

- use the dependency tree as the input sentence's adjacency matrix
- use GCN to extract regional dependency features

$$\overrightarrow{h}_u^{l+1} = ReLU\left(\sum_{v \in \overrightarrow{N}(u)} \left(\overrightarrow{W}^l h_v^l + \overrightarrow{b}^l\right)\right)$$

$$\overleftarrow{h}_u^{l+1} = ReLU\left(\sum_{v \in \overleftarrow{N}(u)} \left(\overleftarrow{W}^l h_v^l + \overleftarrow{b}^l\right)\right)$$

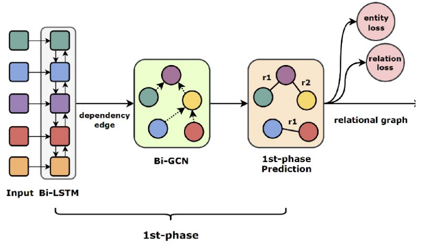
$$h_u^{l+1} = \overrightarrow{h}_u^{l+1} \oplus \overleftarrow{h}_u^{l+1},$$

GraphRel 1st-phase

Training:

- word entity categorical loss : eloss1p

$$-\Sigma_{c=1}^{M} y_{i,c} \log(p_{i,c})$$



Prediction:

- Predict word entity; why doing this?
- remove the dependency edges and do relation prediction for all word pairs (w1, w2)

Calculate relation tendency score(including non-relation):

- for all word pairs (w1, w2) and relation r:

$$S_{(w1,r,w2)} = W_r^3 \, \mathrm{Re}\, LU\left(W_r^1 h_{w1} \oplus W_r^2 h_{w2}\right)$$

- apply softmax function to $S_{(w1,r,w2)}$
- get probability of each relation r for (w1, w2): $P_r(w1, w2)$
- calculate relation categorical loss: rloss1p

GraphRel 2nd-phase

1st-phase not consider interaction between named entities and relations.

- generate complete relation-weighted graph for each relation r
- edge of (w1, w2): Pr(w1,w2)

2nd-phase:

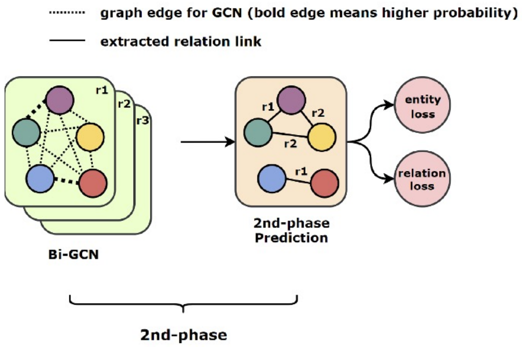
- adopts bi-GCN on each relation graph
- aggregate relations to generate comprehensive word feature

$$h_u^{l+1} = \mathrm{ReLU}\left(\sum_{v \in V} \sum_{r \in R} P_r(u,v) \times \left(W_r^l h_v^l + b_r^l\right)\right) + h_u^l$$

- the edge weight:

  $P_r(u,v)$,the probability of word u to word v under relation r

- Run named entity and relation classification



the texts in these figures are too small

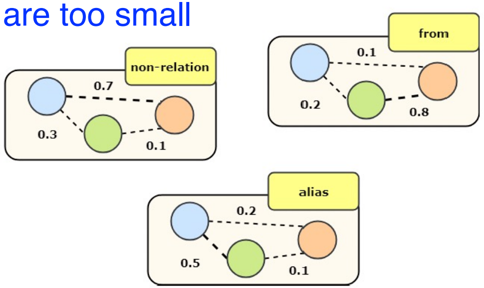


Figure 3: Relation-weighted graph for each relation.

Training

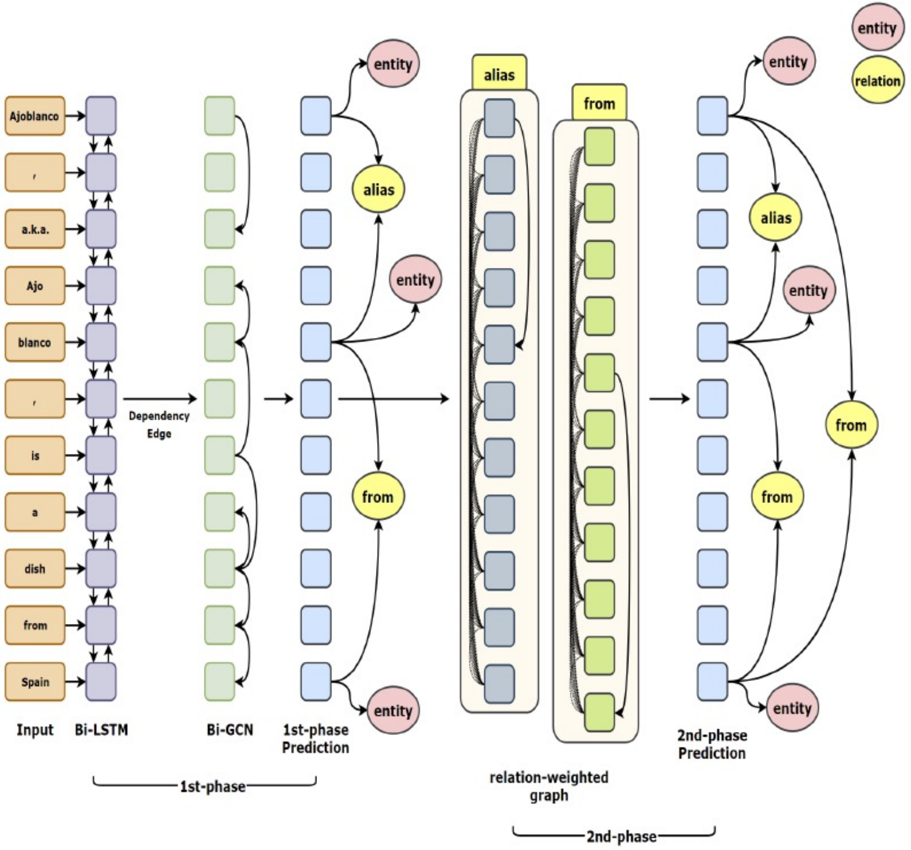
Ground-truth label:

- entity loss: conventional (Begin, Inside, End, Single, Out) tagging
- relation loss: a one-hot relation vector of Pr(w1,w2) for each word pair (w1, w2).

- Calculate categorical cross entropy loss

$$\text{loss}_{all} = (e\,\text{loss}_{1p} + \text{rloss}_{1p}) + \alpha(e\,\text{loss}_{2p} + r\,\text{loss}_{2p})$$

GraphRel



need to add some texts to explain this figure.

Experiment

Datasets: NYT (Riedel et al., 2010),WebNLG (Gardent et al., 2017)
Setting: word embedding(Glove 300d), POS embedding(15d)
The POS tag and the dependency tree were retrieved from spaCy

Method	NYT			WebNLG		
	Precision	Recall	F1	Precision	Recall	F1
NovelTagging	62.4%	31.7%	42.0%	**52.5%**	19.3%	28.3%
OneDecoder	59.4%	53.1%	56.0%	32.2%	28.9%	30.5%
MultiDecoder	61.0%	56.6%	58.7%	37.7%	36.4%	37.1%
GraphRel_{1p}	62.9%	57.3%	60.0%	42.3%	39.2%	40.7%
GraphRel_{2p}	**63.9%**	**60.0%**	**61.9%**	44.7%	**41.1%**	**42.9%**

Table 1: Results for both NYT and WebNLG datasets.

why two-phase is better than one-phase?

GraphRel maintains both high precision and high recall

Experiment

Sentence	GraphRel_{1p}	GrapRel_{2p}
Agra Airport is in India where one of its leaders is Thakur.	(Agra Airport, location, India) (India, leader_name, Thakur)	(Agra Airport, location, India) (India, leader_name, Thakur)
In Italy, the capital is Rome and A.S. Gubbio 1910 is located there.	(Italy, captical, Rome)	(Italy, captical, Rome) (A.S. Gubbio 1910, ground, Italy)
Asam pedas (aka Asam padeh) is from the Sumatra and Malay Peninsula regions of Malaysia.	(Asam pedas, alias, Asam padeh) (Asam pedas, region, Malay Peninsula) (Asam pedas, country, Malaysia)	(Asam pedas, alias, Asam padeh) (Asam pedas, region, Malay Peninsula) (Asam padeh, region, Malay Peninsula) (Asam pedas, country, Malaysia) (Asam padeh, country, Malaysia)

Table 3: Case Study for Graph_{1p} and GraphRel_{2p}.

Thank you

LEHIGH
UNIVERSITY