ACL2020

Reasoning with Latent Structure Refinement for Document-Level Relation Extraction

Guoshun Nan[1*], **Zhijiang Guo**[1*], **Ivan Sekuli**[2†] and **Wei Lu**[1]
[1]StatNLP Research Group, Singapore University of Technology and Design
[2]Universit della Svizzera italiana
guoshun_nan@sutd.edu.sg, zhijiang_guo@mymail.sutd.edu.sg
ivan.sekulic@usi.ch, luwei@sutd.edu.sg

TACL2019 Q3

Densely Connected Graph Convolutional Networks for Graph-to-Sequence Learning

Zhijiang Guo[1*], **Yan Zhang**[1*], **Zhiyang Teng**[1,2], **Wei Lu**[1]

[1]Singapore University of Technology and Design
8 Somapah Road, Singapore, 487372
[2]School of Engineering, Westlake University, China
{zhijiang_guo,yan_zhang,zhiyang_teng}@mymail.sutd.edu.sg
tengzhiyang@westlake.edu.cn, luwei@sutd.edu.sg

ACL2019

Attention Guided Graph Convolutional Networks for Relation Extraction

Zhijiang Guo*, Yan Zhang* and **Wei Lu**
StatNLP Research Group
Singapore University of Technology and Design
{zhijiang_guo,yan_zhang}@mymail.sutd.edu.sg, luwei@sutd.edu.sg

TACL2019 Q3

Densely Connected Graph Convolutional Networks for Graph-to-Sequence Learning

Zhijiang Guo[1*], **Yan Zhang**[1*], **Zhiyang Teng**[1,2], **Wei Lu**[1]

[1]Singapore University of Technology and Design
8 Somapah Road, Singapore, 487372
[2]School of Engineering, Westlake University, China
{zhijiang_guo,yan_zhang,zhiyang_teng}@mymail.sutd.edu.sg
tengzhiyang@westlake.edu.cn, luwei@sutd.edu.sg

DCGCN

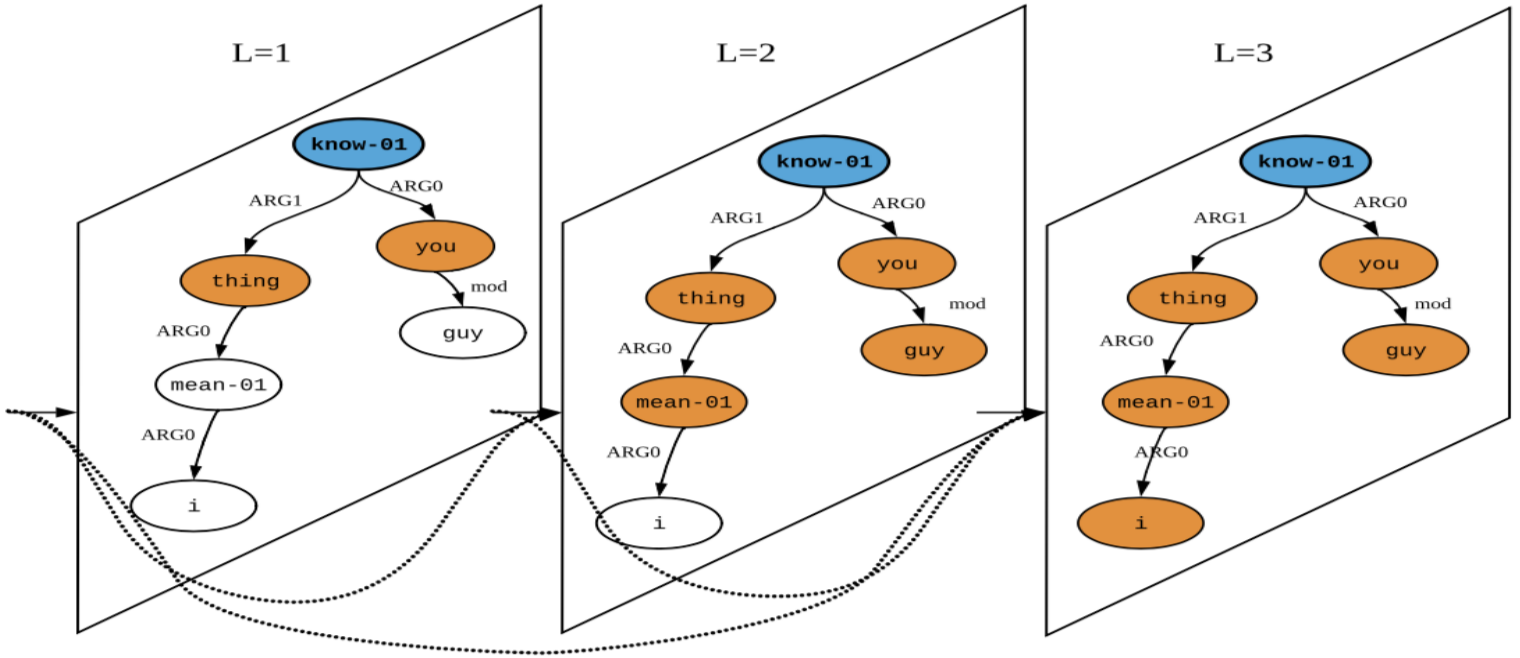


Figure 1: A 3-layer densely connected graph convolutional network. The example AMR graph here corresponds to the sentence ‘‘You guys know what I mean.’’ Every layer encodes information about immediate neighbors and 3 layers are needed to capture third-order neighborhood information (nodes that are 3 hops away from the current node). Each layer concatenates all preceding outputs as the input.

DCGCN

$$\mathbf{h}_v^{(l)} = \rho\Big(\sum_{u \in \mathcal{N}(v)} W^{(l)} \mathbf{h}_u^{(l-1)} + \mathbf{b}^{(l)} \Big)$$

$$\rightarrow \quad \mathbf{g}_u^{(l)} = [\mathbf{x}_u; \mathbf{h}_u^{(1)}; \ldots; \mathbf{h}_u^{(l-1)}]. \quad \mathbf{h}_v^{(l)} = \rho\Big(\sum_{u \in \mathcal{N}(v)} W^{(l)} \mathbf{g}_u^{(l)} + \mathbf{b}^{(l)} \Big)$$

$$\rightarrow \quad \alpha_{ij}^{(l)} = \frac{\exp\left(\phi\left(\mathbf{a}^\top [W_a \tilde{\mathbf{g}}_i^{(l)}; W_a \tilde{\mathbf{g}}_j^{(l)}]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\phi\left(\mathbf{a}^\top [W_a \tilde{\mathbf{g}}_i^{(l)}; W_a \tilde{\mathbf{g}}_k^{(l)}]\right)\right)}$$

$$\mathbf{h}_v^{(l)} = \rho\Big(\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(l)} W^{(l)} \mathbf{g}_u^{(l)} + \mathbf{b}^{(l)} \Big)$$

DCGCN

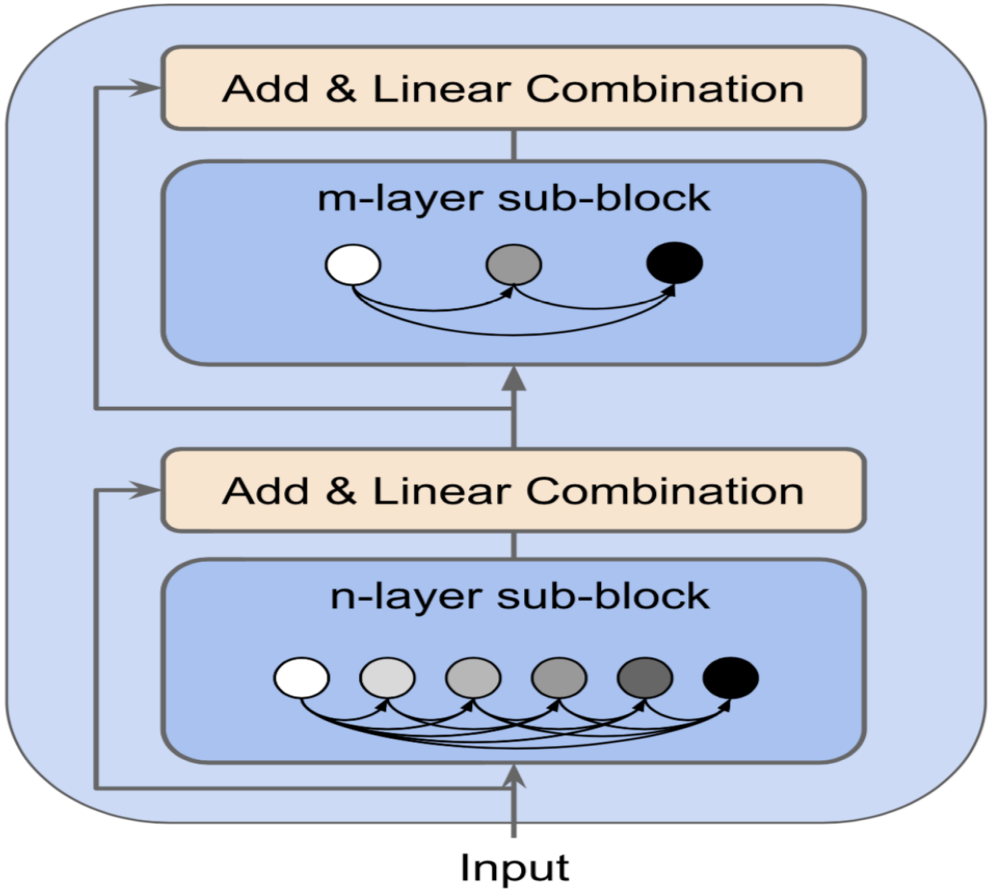


Figure 2: Each DCGCN block has two sub-blocks. Both of them are densely connected graph convolutional layers with different numbers of layers. A linear transformation is used between two sub-blocks, followed by a residual connection.

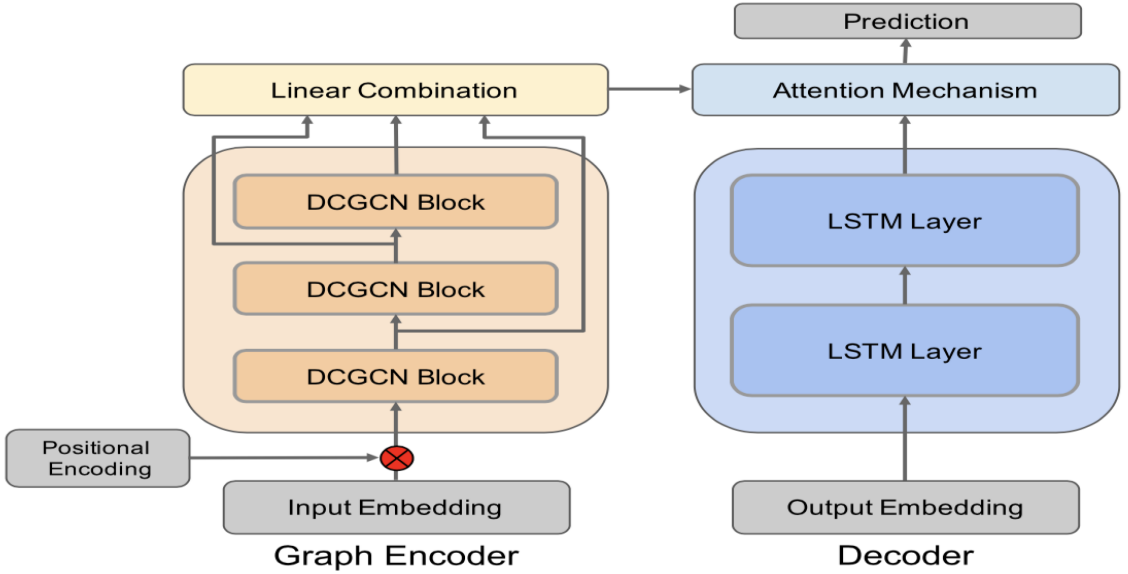


Figure 3: The model concatenates node embeddings and positional embeddings as inputs. The encoder contains a stack of N identical blocks. The linear transformation layer combines output of all blocks into hidden representations. These are fed into an attention mechanism, generating the context vector. The decoder, a 2-layer LSTM (Hochreiter and Schmidhuber, 1997), makes predictions based on hidden representations and the context vector.

Strengths

- Deeper GCN layers to capture richer structural graph representation (both local and non-local) from large graph without suffering from performance degradation and optimization difficulties (Graph LSTM, GGNN)
- Better Encoder for graph data in some domain
- GCN layer (usually 2 is optimal), previous work up to 6 layers, this work up to 36 layers

ACL2019

Attention Guided Graph Convolutional Networks for Relation Extraction

Zhijiang Guo*, Yan Zhang* and **Wei Lu**
StatNLP Research Group
Singapore University of Technology and Design
{zhijiang_guo,yan_zhang}@mymail.sutd.edu.sg, luwei@sutd.edu.sg

Relation Extraction

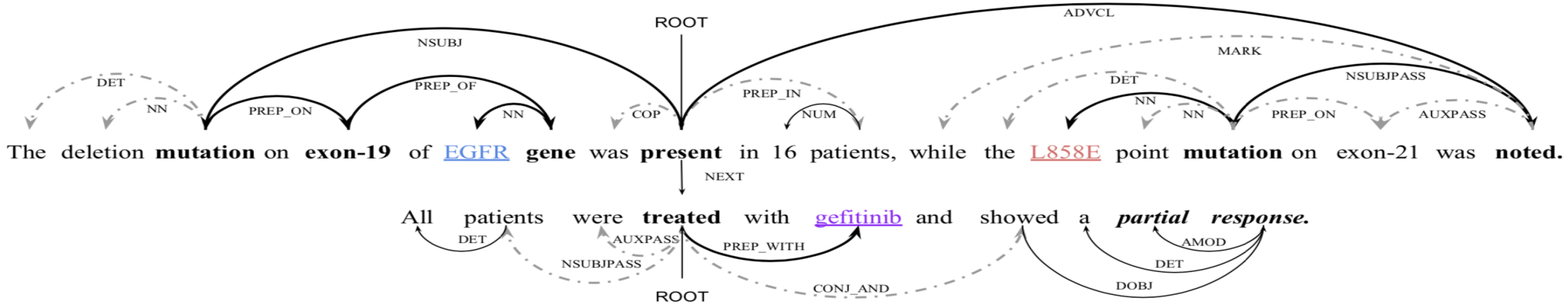
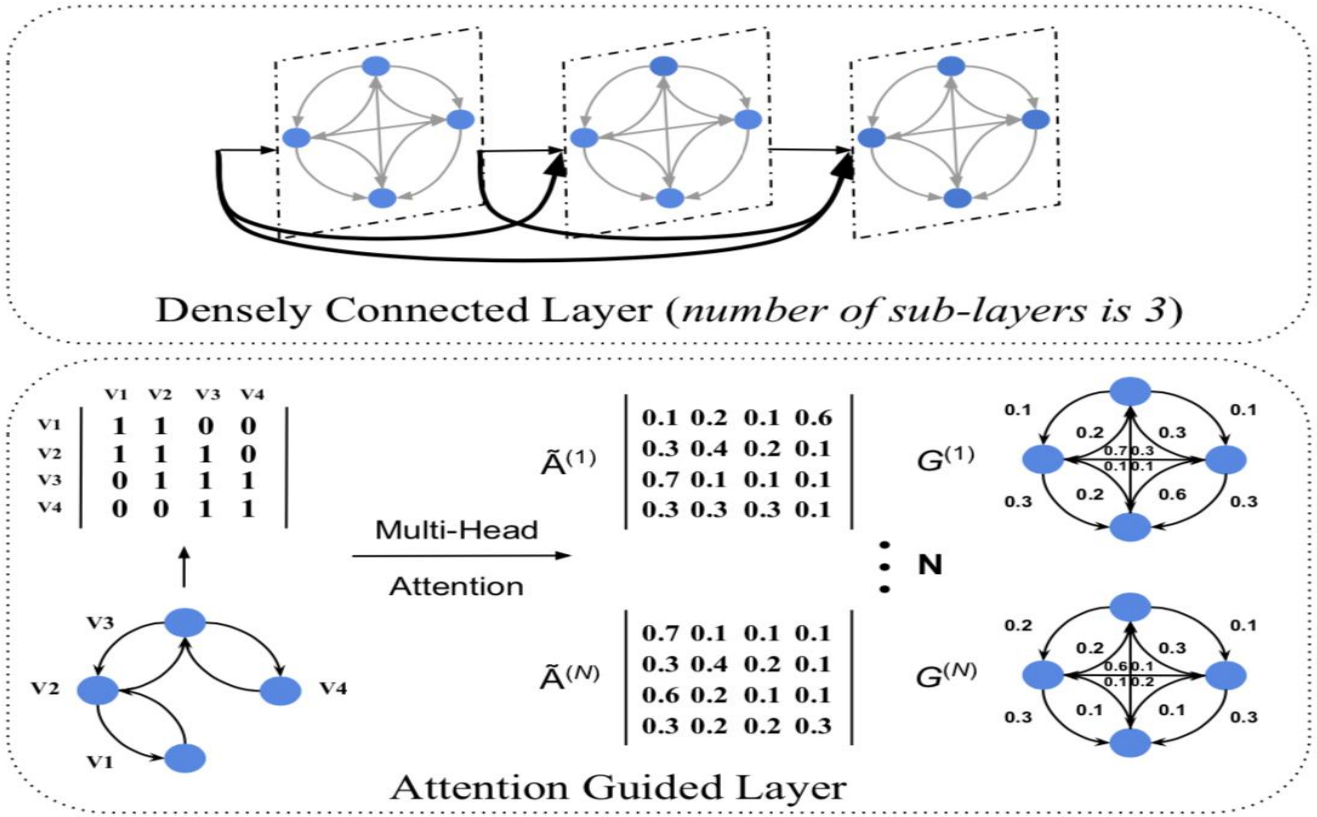
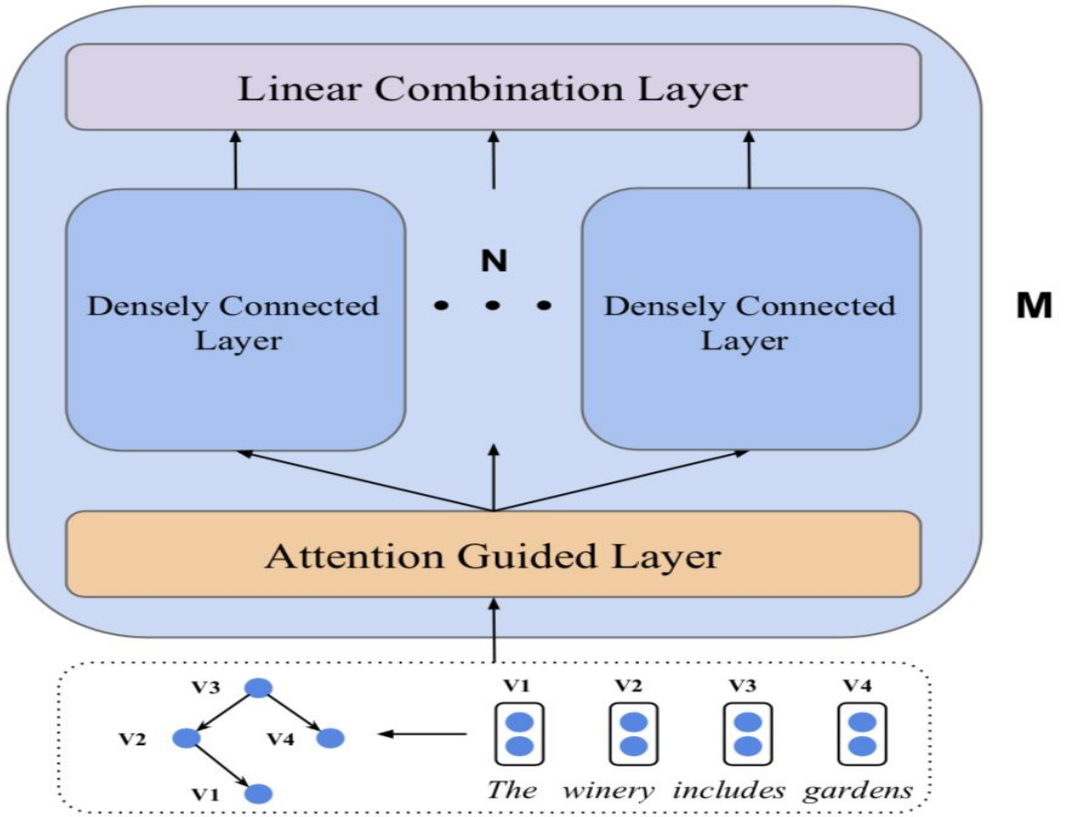


Figure 1: An example dependency tree for two sentences expressing a relation (sensitivity) among three entities. The shortest dependency path between these entities is highlighted in bold (edges and tokens). The root node of the LCA subtree of entities is *present*. The dotted edges indicate tokens K=1 away from the subtree. Note that tokens *partial response* off these paths (shortest dependency path, LCA subtree, pruned tree when K=1).

AGGCN for RE (Encoder)





Contribution

- operate directly on the full syntactic tree to distill the useful information from it in an end-to-end fashion

Definition

- Graph Laplacian Matrix
  - Combinatorial Laplacian $\boldsymbol{L} = D - A$
  - Symmetric normalized Laplacian $\boldsymbol{\mathcal{L}} = D^{-1/2}\boldsymbol{L}D^{-1/2}$
  - Asymmetric normalized (Random Walk) Laplacian $\boldsymbol{\mathcal{L}} = D^{-1}\boldsymbol{L}$

ACL2020

Reasoning with Latent Structure Refinement for Document-Level Relation Extraction

Guoshun Nan[1*], **Zhijiang Guo**[1*], **Ivan Sekuli**[2†] and **Wei Lu**[1]
[1]StatNLP Research Group, Singapore University of Technology and Design
[2]Universit della Svizzera italiana
guoshun_nan@sutd.edu.sg, zhijiang_guo@mymail.sutd.edu.sg
ivan.sekulic@usi.ch, luwei@sutd.edu.sg

Task and Motivation

- Document-level RE on DocRED
- Existing approaches construct static document-level graphs based on syntactic trees, co-references or heuristics from the unstructured text to model the dependencies. They may not be able to capture rich non-local interactions for inference at doc-level.

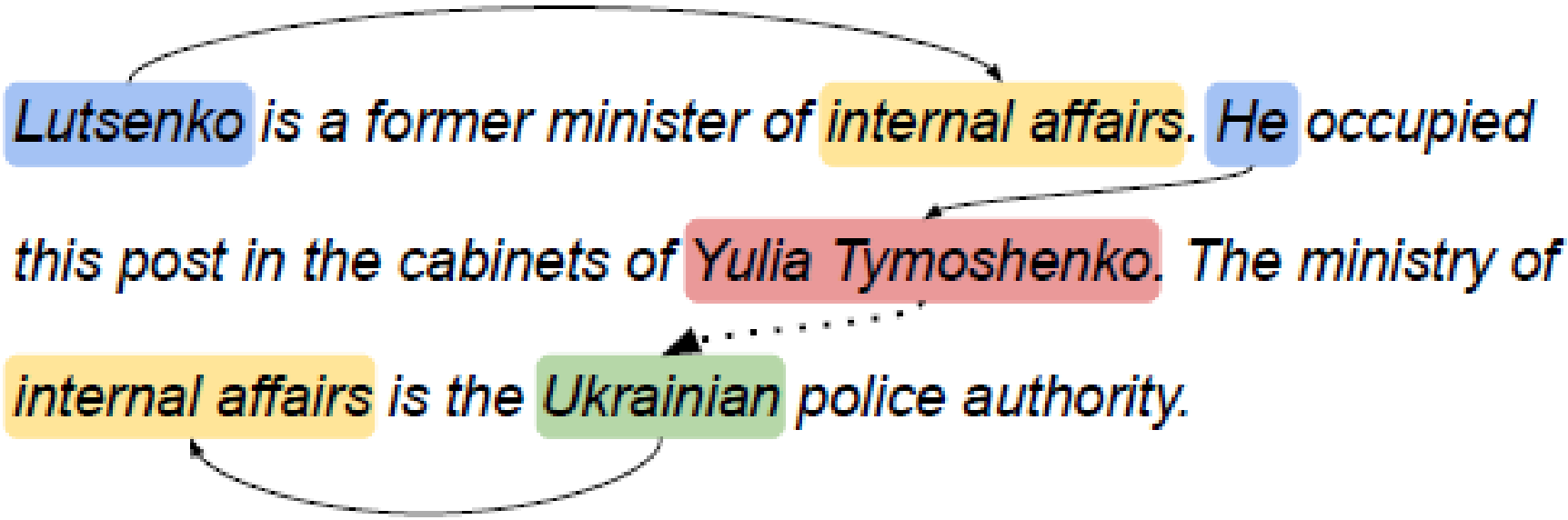
Task and Motivation

- we propose a novel model that empowers the relational reasoning across sentences by automatically inducing the latent document-level graph.

- We further develop a refinement strategy, which enables the model to incrementally aggregate relevant information for multi-hop reasoning.

- Build upon sturcture attention and Matrix-Tree Theorem

Running Example



Subject: *Yulia Tymoshenko*

Object: *Ukrainian*

Relation: country of citizenship

Method

- Node Constructor
- Dynamic Reasoner
- Classifier

Node Constructor

$$\overleftarrow{\mathbf{h}}^i_j = \mathbf{LSTM}_l(\overleftarrow{\mathbf{h}}^i_{j+1}, \gamma^i_j)$$
$$\overrightarrow{\mathbf{h}}^i_j = \mathbf{LSTM}_r(\overrightarrow{\mathbf{h}}^i_{j-1}, \gamma^i_j)$$
$$\mathbf{h}^i_j = [\overleftarrow{\mathbf{h}}^i_j; \overrightarrow{\mathbf{h}}^i_j]$$

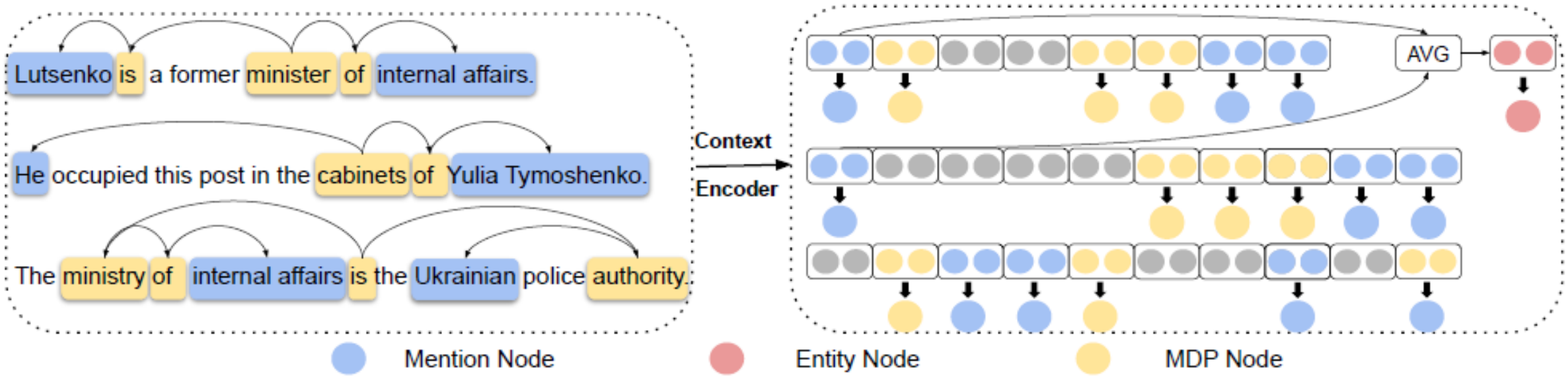


Figure 2: Overview of the Node Constructor: A context encoder is applied to get the contextualized representations of sentences. The representations of mentions and words in the meta dependency paths are extracted as mention nodes and MDP nodes. An average pooling is used to construct the entity node from the mention nodes. For example, the entity node *Lutsenko* is constructed by averaging representations of its mentions *Lutsenko* and *He*. All figures best viewed in color.

Dynamic Reasoner

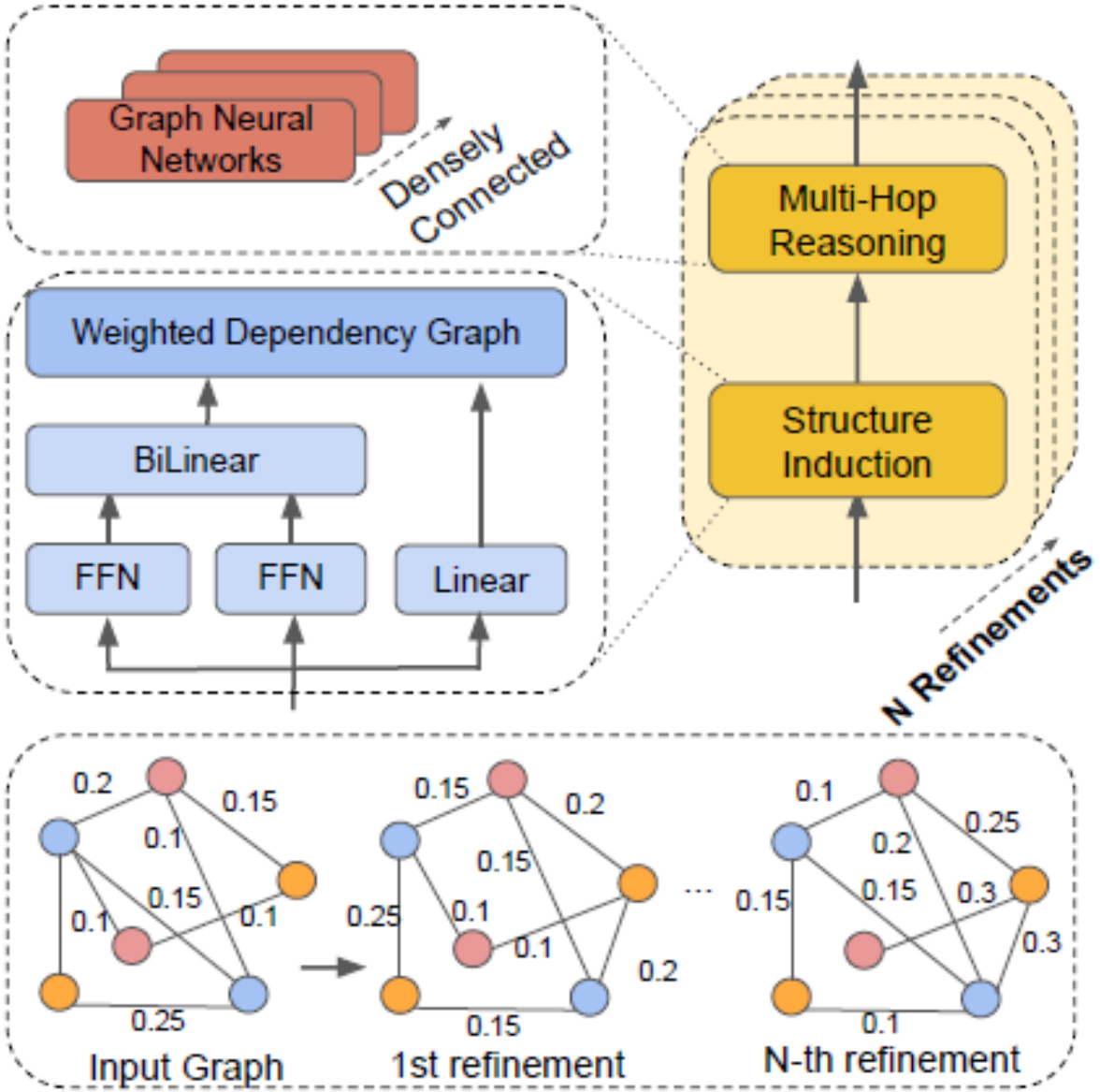
Structure Attention

$$\mathbf{s}_{ij} = (\tanh(\mathbf{W}_p \mathbf{u}_i))^T \mathbf{W}_b (\tanh(\mathbf{W}_c \mathbf{u}_j)) \quad (3)$$

root score

$$\mathbf{s}_i^r = \mathbf{W}_r \mathbf{u}_i \quad (4)$$

unnormalized probability of the i-th node to be selected as the root node of the structure



Structure Indcution

Edge Weight:

$$\mathbf{P}_{ij} = \begin{cases} 0 & \text{if } i = j \\ \exp\left(\mathbf{s}_{ij}\right) & \text{otherwise} \end{cases} \tag{5}$$

Terry K Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. **Structured prediction models via the matrix-tree theorem**. In Proc. of EMNLP-CoNLL.

Two Laplacian variants:

$$\mathbf{L}_{ij} = \begin{cases} \sum_{i'=1}^{n} \mathbf{P}_{i'j} & \text{if } i = j \\ -\mathbf{P}_{ij} & \text{otherwise} \end{cases} \tag{6}$$

$$\hat{\mathbf{L}}_{ij} = \begin{cases} \exp(\mathbf{s}_i^r) & \text{if } i = 1 \\ \mathbf{L}_{ij} & \text{if } i > 1 \end{cases} \tag{7}$$

We use $\mathbf{A}_{ij}$ to denote the marginal probability of the dependency edge between the i-th and the j-th node. Then, $\mathbf{A}_{ij}$ can be derived based on Equation (8), where δ is the Kronecker delta (Koo et al., 2007).

$$\begin{aligned} \mathbf{A}_{ij} = (1 - \delta_{1,j})\mathbf{P}_{ij}[\hat{\mathbf{L}}^{-1}]_{ij} \\ -(1 - \delta_{i,1})\mathbf{P}_{ij}[\hat{\mathbf{L}}^{-1}]_{ji} \end{aligned} \tag{8}$$

Multi-hop Reasoning

$$\mathbf{g}_u^{(l)} = [\mathbf{x}_u; \mathbf{h}_u^{(1)}; \ldots; \mathbf{h}_u^{(l-1)}]. \quad \mathbf{h}_v^{(l)} = \rho\Big(\sum_{u \in \mathcal{N}(v)} W^{(l)} \mathbf{g}_u^{(l)} + \mathbf{b}^{(l)} \Big)$$

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\phi\left(\mathbf{a}^\top [W_a \tilde{\mathbf{g}}_i^{(l)}; W_a \tilde{\mathbf{g}}_j^{(l)}]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\phi\left(\mathbf{a}^\top [W_a \tilde{\mathbf{g}}_i^{(l)}; W_a \tilde{\mathbf{g}}_k^{(l)}]\right)\right)},$$

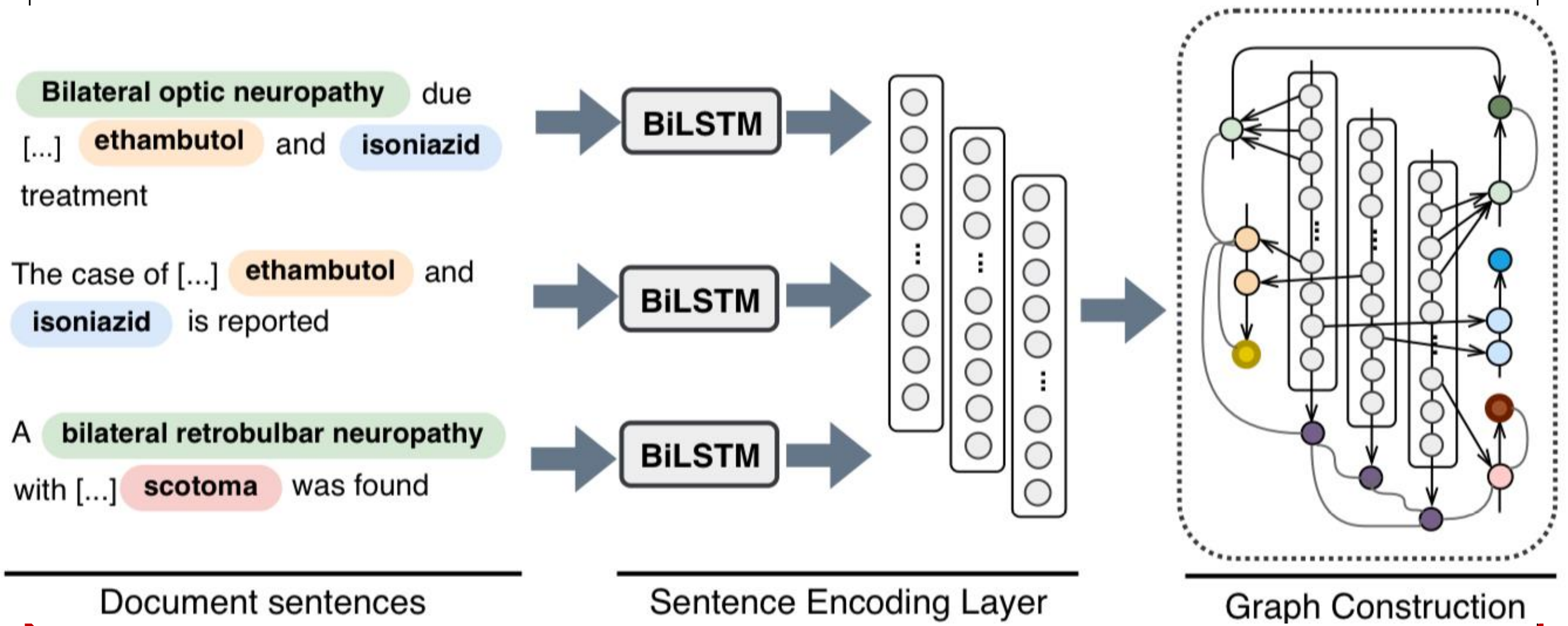
$$\mathbf{h}_v^{(l)} = \rho\Big(\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(l)} W^{(l)} \mathbf{g}_u^{(l)} + \mathbf{b}^{(l)} \Big)$$

Dataset

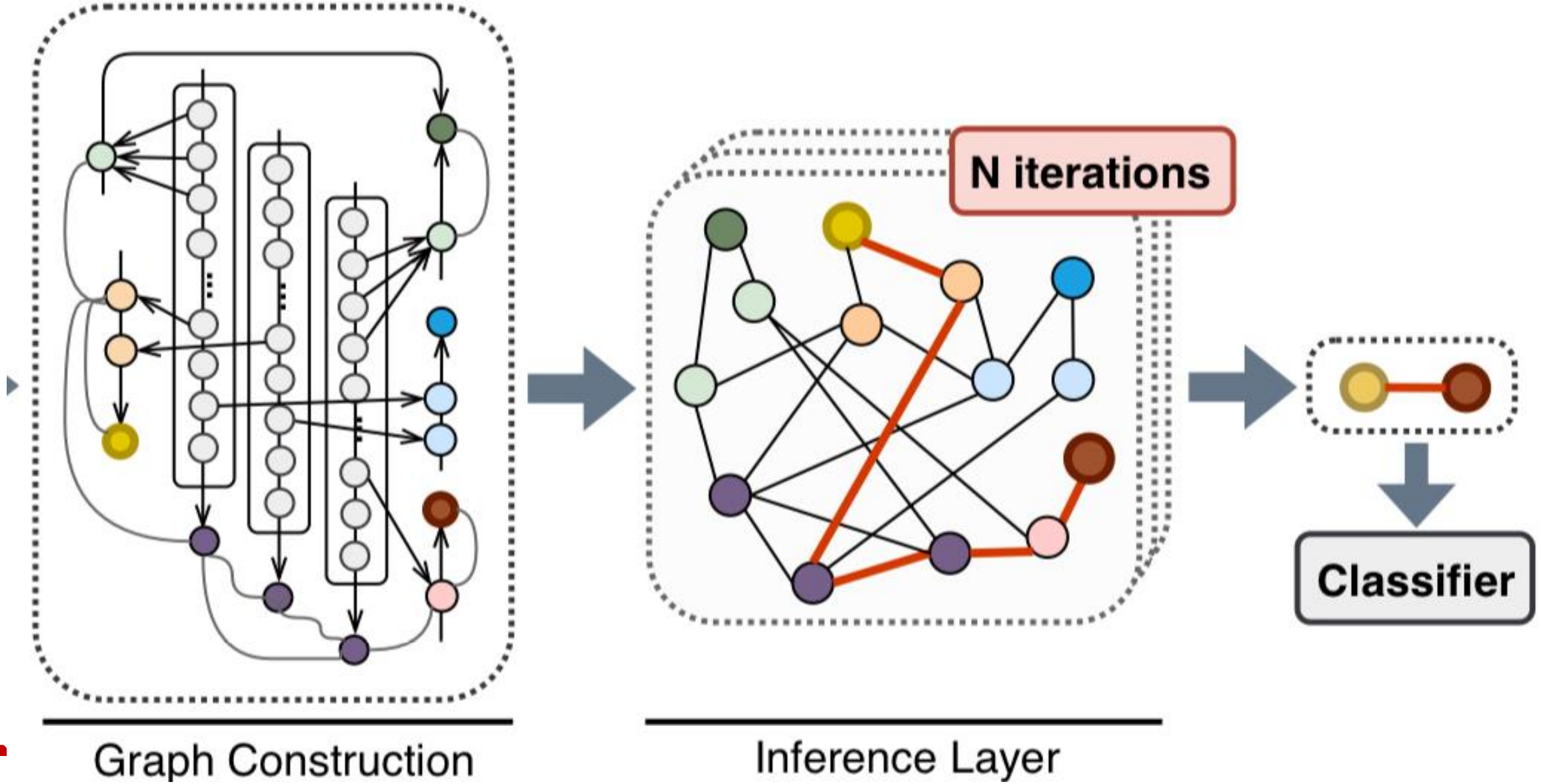
- DocRED, a large-scale human-annotated dataset for document-level RE
- 3053 documents for training, 1000 for development and 1000 for test
- 96 relations
- 8 sentences on average and more than 40.7% relation facts can only be extracted from multiple sentences. And 61.1% relation instances require a variety of inference skills such as logical inference,

Model	Dev				Test	
	Ign $F1$	$F1$	Intra-$F1$	Inter-$F1$	Ign $F1$	$F1$
CNN (Yao et al., 2019)	41.58	43.45	51.87*	37.58*	40.33	42.26
LSTM (Yao et al., 2019)	48.44	50.68	56.57*	41.47*	47.71	50.07
BiLSTM (Yao et al., 2019)	48.87	50.94	57.05*	43.49*	48.78	51.06
ContextAware (Yao et al., 2019)	**48.94**	51.09	56.74*	42.26*	48.40	50.70
GCNN ♣ (Sahu et al., 2019)	46.22	51.52	57.78	44.11	49.59	51.62
EoG ♣ (Christopoulou et al., 2019)	45.94	52.15	58.90	44.60	49.48	51.82
GAT ♣ (Veličković et al., 2018)	45.17	51.44	58.14	43.94	47.36	49.51
AGGCN ♣ (Guo et al., 2019a)	46.29	52.47	58.76	45.45	48.89	51.45
GloVe+LSR	48.82	**55.17**	**60.83**	**48.35**	**52.15**	**54.18**
BERT (Wang et al., 2019)	-	54.16	61.61*	47.15*	-	53.20
Two-Phase BERT (Wang et al., 2019)	-	54.42	61.80*	47.28*	-	53.92
BERT+LSR	**52.43**	**59.00**	**65.26**	**52.05**	**56.97**	**59.05**

Architecture



Architecture



Does Latent Structure and Refinement Matter?

We adapt rules from De Cao et al. (2019) for multi-hop question answering, i.e., each mention node is connected to its entity node and to the same mention nodes across sentences, while mention nodes and MDP nodes which reside in the same sentence are fully connected. The model is termed QAGCN.

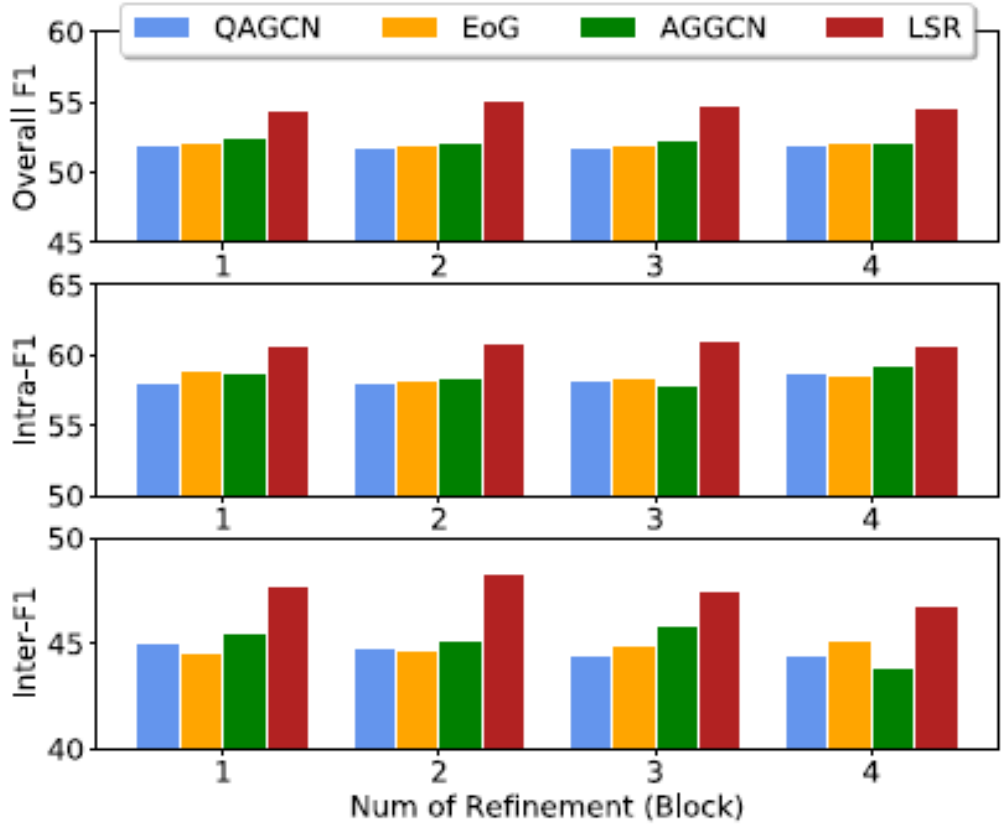


Figure 4: Intra- and inter-sentence $F1$ for different graph structures in QAGCN, EoG, AGGCN and LSR. The number of refinements is ranging from 1 to 4.

Model	$F1$	Intra-$F1$	Inter-$F1$
Full model	55.17	60.83	48.35
- 1 Refinement	54.42	60.46	47.67
- 2 Structure Induction	51.91	58.08	45.04
- 1 Multi-hop Reasoning	54.49	59.75	47.49
- 2 Multi-hop Reasoning	54.24	60.58	47.15
- MDP nodes	54.20	60.54	47.12

Table 5: Ablation study of LSR on DocRED.

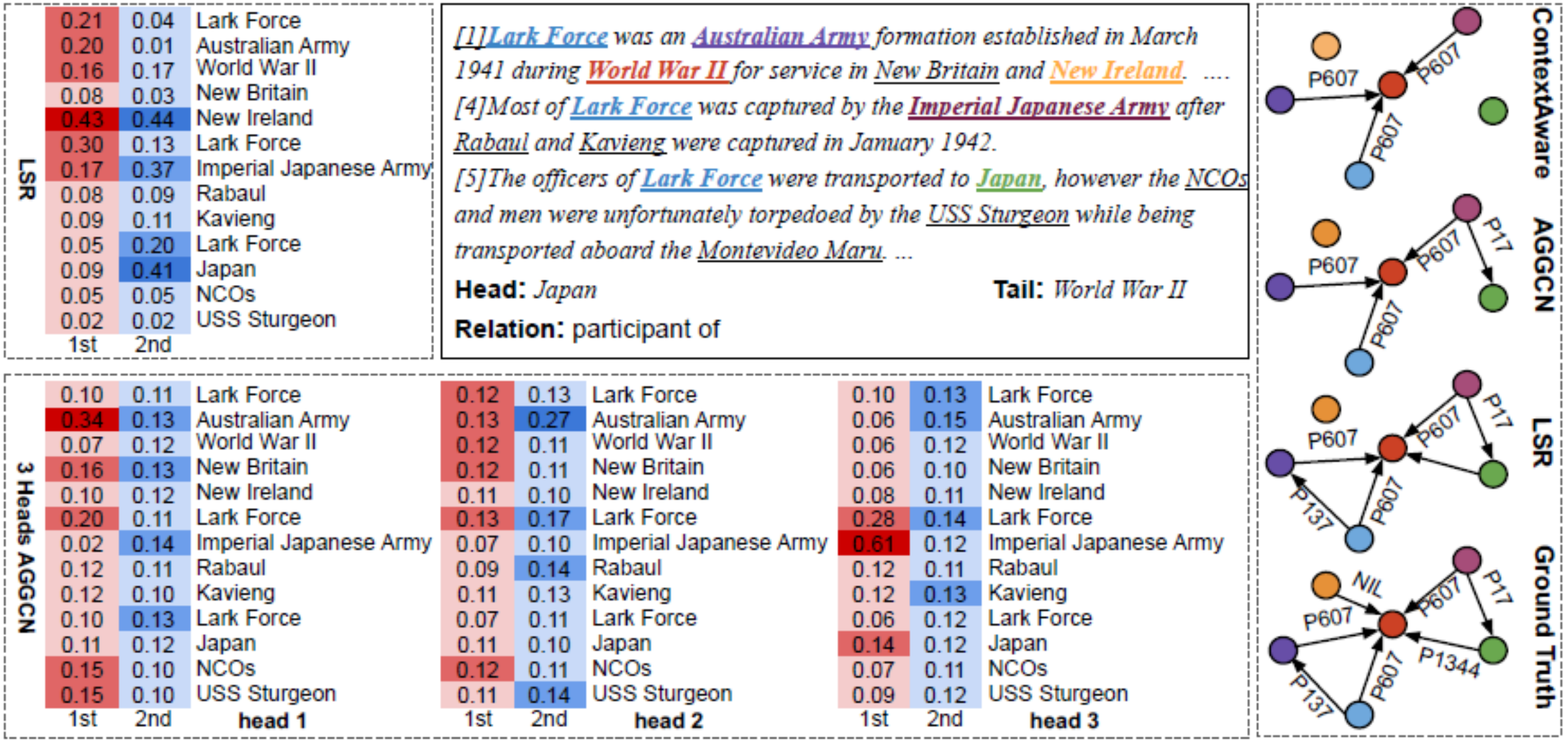


Figure 5: Case study of an example from the development set of DocRED. We visualize the reasoning process for predicting the relation of an entity pair ⟨*Japan*, *World War II*⟩ by LSR and AGGCN in two refinement steps, using the attention scores of the mention *World War II* in each step. We scale all attention scores by 1000 to illustrate them more clearly. Some sentences are omitted due to space limitation.

Thanks!