EMNLP 2019

Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs

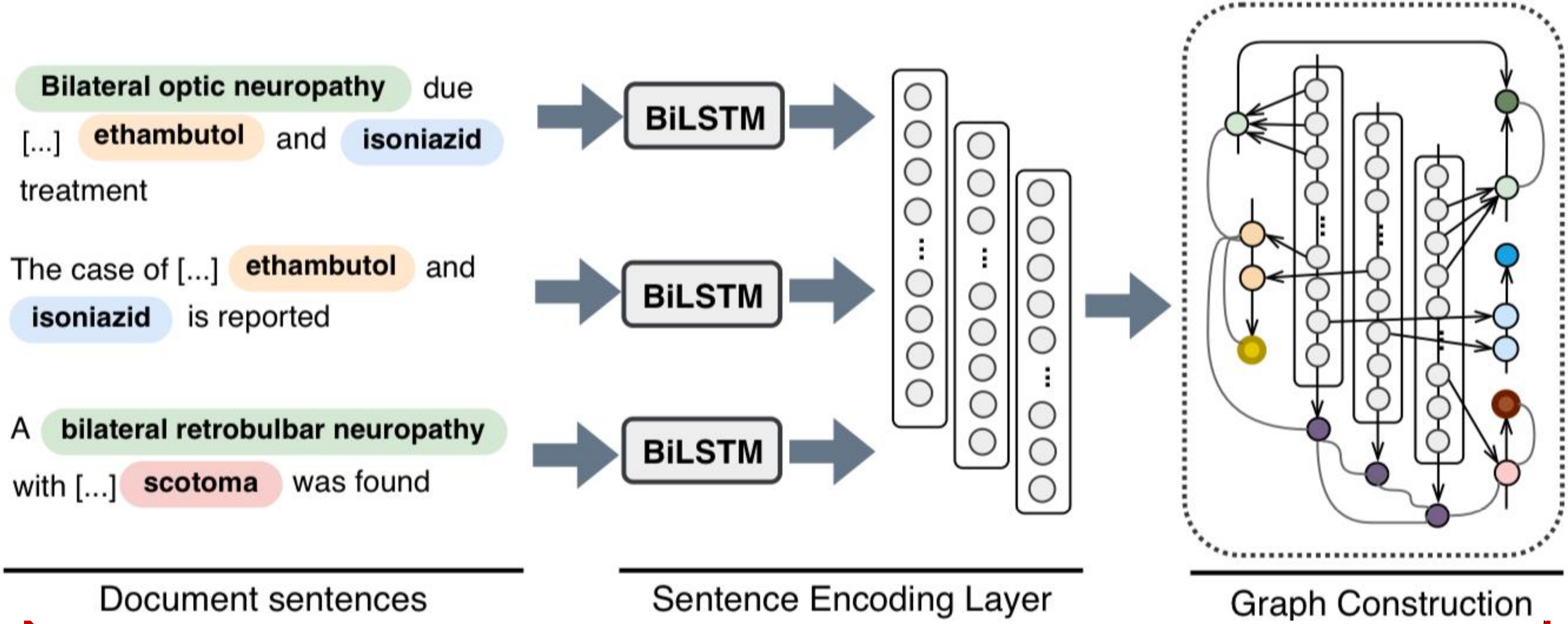
Fenia Christopoulou[1], **Makoto Miwa**[2,3], **Sophia Ananiadou**[1]

[1]National Centre for Text Mining,
School of Computer Science, The University of Manchester, United Kingdom
[2]Toyota Technological Institute, Nagoya, 468-8511, Japan
[3]Artificial Intelligence Research Center (AIRC),
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{efstathia.christopoulou, sophia.ananiadou}@manchester.ac.uk
makoto-miwa@toyota-ti.ac.jp

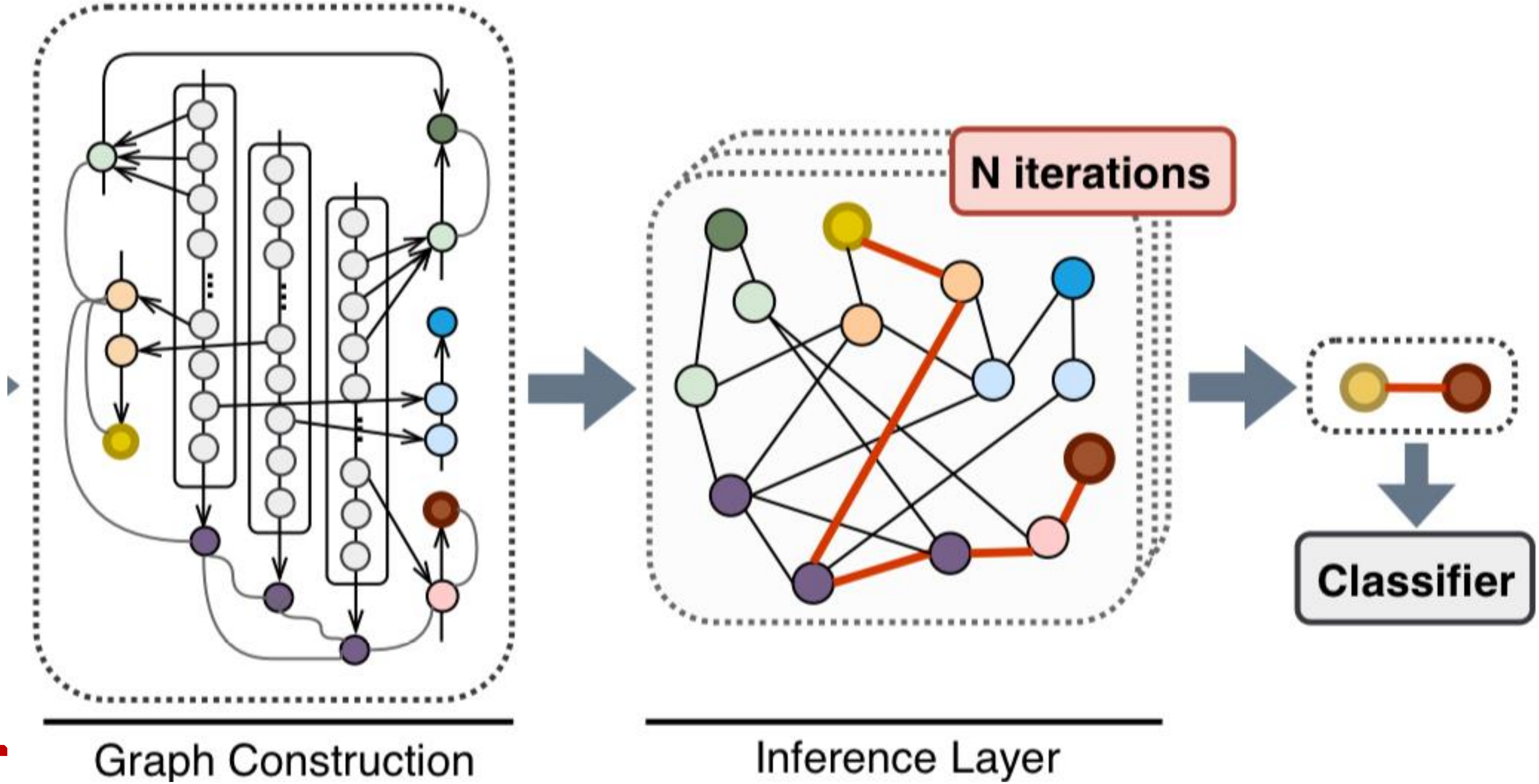
Task and Motivation

- Document-level Relation Extraction on Chemical-Disease Reaction and Gene-Disease Association datasets.
- Existing approaches use graph-based neural models with words as nodes and edges as relations between them, to encode relations across sentences.
- Node-based vs. Relation based

Architecture



Architecture



Node construction

- mention node (M): $\boldsymbol{n}_m = [\mathrm{avg}_{\omega_i \in m}(\boldsymbol{w}_i); \boldsymbol{t}_m]$
- entity node (E): $\boldsymbol{n}_e = [\mathrm{avg}_{m_i \in e}(\boldsymbol{m}_i); \boldsymbol{t}_e]$
- sentence node (S): $\boldsymbol{n}_s = [\mathrm{avg}_{\omega_i \in s}(\boldsymbol{w}_i); \boldsymbol{t}_s]$

Mention-Mention(MM)

- Co-occurrence of mentions in a sentence might be a weak indication of an interaction.
- $\boldsymbol{x}_{MM} = \left[\boldsymbol{n}_{m_i}; \boldsymbol{n}_{m_j}; \boldsymbol{c}_{m_i,m_j}; \boldsymbol{d}_{m_i,m_j}\right]$
- argument-based attention mechanism to measure the importance of other words in the sentence towards the mention, denoting $k \in \{1,2\}$ as the mention arguments.
- $H \in \mathbb{R}^{w \times d}$

$$\alpha_{k,i} = \mathbf{n}_{m_k}^{\intercal} \mathbf{w}_i,$$

$$\mathrm{a}_{k,i} = \frac{\exp(\alpha_{k,i})}{\sum_{j \in [1,n], j \notin m_k} \exp(\alpha_{k,j})},$$

$$\mathrm{a}_i = (\mathrm{a}_{1,i} + \mathrm{a}_{2,i})/2,$$

$$\mathbf{c}_{m_1,m_2} = \mathbf{H}^{\intercal}\,\mathbf{a},$$

edge construction

- **Mention-Sentence (MS):** if the mention resides in the sentence.

$$\boldsymbol{x}_{MS} = [\boldsymbol{n}_m; \boldsymbol{n}_s]$$

- **Mention-Entity (ME):** if the mention is associated with the entity.

$$\boldsymbol{x}_{ME} = [\boldsymbol{n}_m; \boldsymbol{n}_e]$$

- **Sentence-Sentence (SS):** to encode non-local information.

$$\boldsymbol{x}_{SS} = \left[\boldsymbol{n}_{s_i}; \boldsymbol{n}_{s_j}; \boldsymbol{d}_{s_i,s_j}\right]$$

- **Entity-Sentence (ES):** if at least one mention of the entity resides in this sentence.

$$\boldsymbol{x}_{ES} = [\boldsymbol{n}_e; \boldsymbol{n}_s]$$

In order to result in edge representations of equal dimensionality, we use different linear reduction layers for different edge representations,

$$\mathbf{e}_z^{(1)} = \mathbf{W}_z \, \mathbf{x}_z \, , \tag{2}$$

where $\mathbf{e}_z^{(1)}$ is an edge representation of length 1, $\mathbf{W}_z \in \mathbb{R}^{d_z \times d}$ corresponds to a learned matrix and $z \in [\mathrm{MM}, \mathrm{MS}, \mathrm{ME}, \mathrm{SS}, \mathrm{ES}]$.

Inference Layer

- We utilize an iterative algorithm to generate edges between different nodes in the graph, as well as to update existing edges.

- We initialize the graph only with the edges [MM, MS, ME, SS, ES].

- We can only generate EE edge representations by representing a path between their nodes:

- E-M-M-E (intra), E-M-S-E, E-S-M-E, E-S-S-E, E-S-E (intra) ….

Two-step inference

- At the first step, we aim to generate a path between two nodes i and j using intermediate nodes k.
- We thus combine the representations of two consecutive edges e_{ik} and e_{kj} , using a modified bilinear transformation:

$$f\left(\mathbf{e}_{ik}^{(l)}, \mathbf{e}_{kj}^{(l)}\right) = \sigma\left(\mathbf{e}_{ik}^{(l)} \odot \left(\mathbf{W}\ \mathbf{e}_{kj}^{(l)}\right)\right)$$

- This action generates an edge representation of double length.

Two-step inference

- During the second step, we aggregate the original (short) edge representation and the new (longer) edge representation resulted from the first step with linear interpolation as follows:

$$\mathbf{e}_{ij}^{(2l)} = \beta\, \mathbf{e}_{ij}^{(l)} + (1-\beta) \sum_{k \neq i,j} f\left(\mathbf{e}_{ik}^{(l)}, \mathbf{e}_{kj}^{(l)}\right)$$

- where $\beta \in [0,1]$ is a scalar that controls the contribution of the shorter edge presentation. In general β is larger for shorter edges as we expect that the relation between two nodes is better expressed through the shortest path between them.

The two steps are repeated a finite number of times N. The number of iterations is correlated with the final length of the edge representations. With initial edge length l equal to 1, the first iteration results in edges of length up-to 2. The second iteration results in edges of length up-to 4. Similarly, after N iterations, the length of edges will be up-to 2^N.

2.5 Classification Layer

To classify the concept-level entity pairs of interest, we incorporate a softmax classifier, using the entity-to-entity edges (EE) of the document graph that correspond to the concept-level entity pairs.

$$\mathbf{y} = \mathrm{softmax}\left(\mathbf{W}_c\, \mathrm{e}_{\mathrm{EE}} + \mathrm{b}_c\right), \tag{5}$$

where $\mathbf{W}_c \in \mathbb{R}^{r \times d_z}$ and $\mathrm{b}_c \in \mathbb{R}^{r}$ are learned parameters of the classification layer and r is the number of relation categories.

Datasets

CDR (BioCreative V): The Chemical-Disease Reactions dataset was created by Li et al. (2016a) for document-level RE. It consists of $1,500$ PubMed abstracts, which are split into three equally sized sets for training, development and testing. The dataset was manually annotated with binary interactions between Chemical and Disease concepts. For this dataset, we utilised PubMed pre-trained word embeddings (Chiu et al., 2016).

	Train	Dev	Test
Documents	500	500	500
Positive pairs	1,038	1,012	1,066
Intra	754	766	747
Inter	284	246	319
Negative pairs	4,202	4,075	4,138
Entities			
Chemical	1,467	1,507	1,434
Disease	1,965	1,864	1,988
Mentions			
Chemical	5,162	5,307	5,370
Disease	4,252	4,328	4,430

Table 7: CDR (BioCreative V) dataset statistics.

Datasets

GDA (DisGeNet): The Gene-Disease Associations dataset was introduced by Wu et al. (2019), containing $30,192$ MEDLINE abstracts, split into $29,192$ articles for training and $1,000$ for testing. The dataset was annotated with binary interactions between Gene and Disease concepts at the document-level, using distant supervision. Associations between concepts were generated by aligning the DisGeNet (Piñero et al., 2016) platform with PubMed[3] abstracts. We further split the training set into a 80/20 percentage split as training and development sets. For the GDA dataset, we used randomly initialized word embeddings.

	Train	Dev	Test
Documents	23,353	5,839	1,000
Positive pairs	36,079	8,762	1,502
Intra	30,905	7,558	1,305
Inter	5,174	1,204	197
Negative pairs	96,399	24,362	3,720
Entities			
Gene	46,151	11,406	1,903
Disease	67,257	16,703	2,778
Mentions			
Gene	205,457	51,410	8,404
Disease	226,015	56,318	9,524

Table 8: GDA (DisGeNet) dataset statistics.

Results

	Method	Overall (%)			Intra (%)			Inter (%)		
		P	R	F1	P	R	F1	P	R	F1
CNN	Gu et al. (2017)	55.7	68.1	61.3	59.7	55.0	57.2	51.9	7.0	11.7
Multi-head CNN	Verga et al. (2018)	55.6	70.8	62.1	-	-	-	-	-	-
char-CNN	Nguyen and Verspoor (2018)	57.0	68.6	62.3	-	-	-	-	-	-
	EoG	62.1	65.2	**63.6**	64.0	73.0	**68.2**	56.0	46.7	**50.9**
	EoG (`Full`)	59.1	56.2	57.6	71.2	62.3	66.5	37.1	42.0	39.4
	EoG (`NoInf`)	48.2	50.2	49.2	65.8	55.2	60.2	25.4	38.5	30.6
	EoG (`Sent`)	56.9	53.5	55.2	56.9	76.4	65.2	-	-	-
with additional data or tools or external knowledge	Zhou et al. (2016)	55.6	68.4	61.3	-	-	-	-	-	-
	Peng et al. (2016)	62.1	64.2	63.1	-	-	-	-	-	-
	Li et al. (2016b)	60.8	76.4	67.7	67.3	52.4	58.9	-	-	-
	Panyam et al. (2018)	53.2	69.7	60.3	54.7	80.6	65.1	47.8	43.8	45.7
	Zheng et al. (2018)	56.2	67.9	61.5	-	-	-	-	-	-

Results

Model	Dev | Test F1 (%)					
	Overall		Intra		Inter	
EoG	78.7	81.5	82.5	85.2	48.8	50.0
EoG (`Full`)	78.6	80.8	82.4	84.1	52.3	54.7
EoG (`NoInf`)	71.8	74.6	76.8	79.1	45.5	49.3
EoG (`Sent`)	73.8	73.8	78.1	78.8	-	-

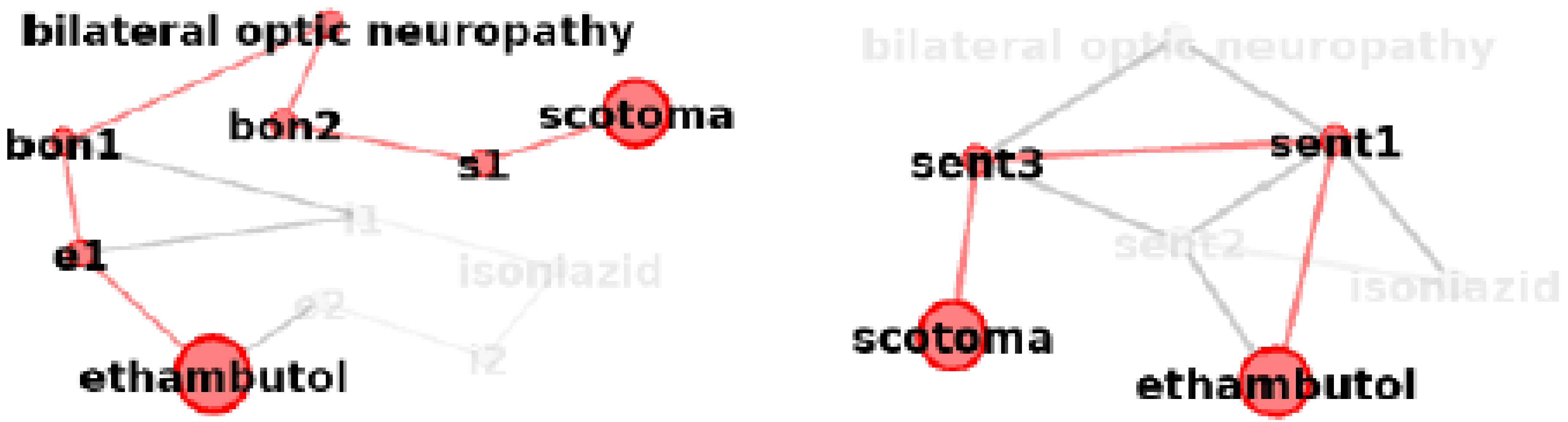
Table 2: Performance comparison on the GDA development and test sets.

Analysis

Edge Types	F1 (%)		
	Overall	Intra	Inter
EE	55.14	61.31	40.34
EoG	63.57	68.25	46.68
−MM	62.77	67.93	46.65
−ME	61.57	66.39	45.40
−MS	62.92	67.55	44.74
−ES	61.41	66.44	43.04
−$SS_{indirect}$	59.70	67.09	28.00
−SS	57.41	65.45	1.59
−MM,ME,MS	60.46	66.07	39.56
−ES,MS,SS	56.86	64.63	0.00

Table 4: Ablation analysis for different edge and node types on the CDR development set.

Analysis



(a) MM, ME edges (b) ES, SS edges

Figure 4: Relation paths with different types of edges.

Contributions and Thoughts

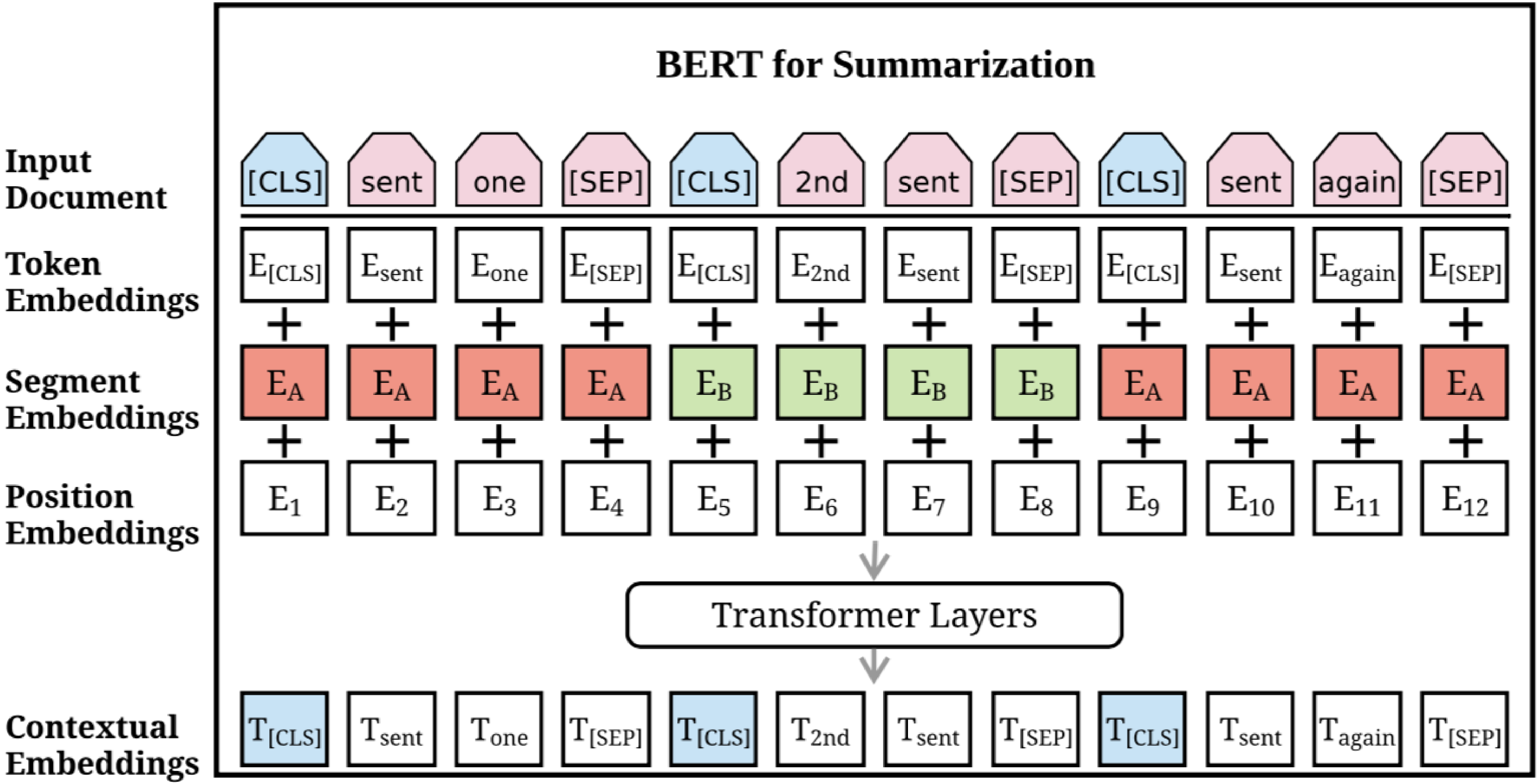
- New inference mechanism on edge-oriented graph.

- Representation of each word only cover the contextual information of the sentence it reside in. (document-level encoder?)
- It is intuitively unreliable to form node representation (entity, mention, sentence) by averaging the elements of them. (mask-based or structure-aware representation learning?)

This week

- Survey papers in the field of "GNN for complex reasoning"
- DocRED baseline code debugging

Model—Summarization Encoder



Thanks!