EMNLP 2020

Double Graph Based Reasoning for Document-level Relation Extraction

Paper: https://arxiv.org/pdf/2009.13752.pdf
Github: https://github.com/DreamInvoker/GAIN

집현전 중급

손수현

Abstract

Document-level relation extraction

: sentence-level relation extraction 과는 달리, multiple sentence에서의 relation extraction

: 본 paper에서는 **G**raph **A**ggregation-and **I**nference **N**etwork **(GAIN)** featuring double graphs를 제안

→ 먼저, heterogeneous mention-level graph(hMG)를 구축하여 document내의 서로 다른 mention간의 relation을 modeling한다

→ Entity-level Graph(EG)를 구성하고, 이를 기반으로 entities간의 relation을 infer할 수 있는 새로운 mechanism을 제안한다.

: DocRED dataset에서 이전 SOTA보다 2.85 정도 향상된 F1 score

Introduction

- Text에서 entities간의 semantic relation을 identify하는 task인 Relation Extraction(RE)는 QA와 같은 knowledge-based applications의 중요한 task
- Previous method들은

  single sentence에서 entities간의 relation을 extraction하는 “sentence level relation extraction”에 focus

  → 여러 sentence에 걸쳐 있는 entities간의 relation은 extract하지 못하는 한계

- 그래서, Document-level relation extraction task에 대한 다양한 시도가 있었음

  → Firstly,

  relation을 가지는 subject entities와 object entities가 다른 sentence에 존재할 수 있기 때문에 sentence-level에서 처럼, 각 sentence에 대해 relation을 정의할 수 없음

  → Secondly,

  같은 entity가 여러 sentence에서 언급될 수 있기 때문에, entity를 잘 표현하기 위해서 cross-sentence context정보가 더해져야 함

  → Thirdly,

  다양한 relation을 identify하기 위해서는 logical reasoning technique이 필요

Introduction

> **Elias Brown**
> [1] *Elias Brown* (*May 9, 1793*– *July 7, 1857*) was a ***U.S.*** Representative from ***Maryland***. [2] Born near ***Baltimore***, ***Maryland***, *Brown* attended the common schools. … [7] He died near ***Baltimore***, ***Maryland***, and is interred in a private cemetery near ***Eldersburg***, ***Maryland***.

- 아래와 같은, Intra-sentence relations는 쉽게 recognize할 수 있다

  (Maryland, country, U.S.), (Baltimore, located in the administrative territorial entity, Maryland), (Eldersburg, located in the administrative territorial entity, Maryland)

- 하지만 아래와 같은, 한 sentence에 있지 않고 long-distance dependencies를 가지는 경우 앞서 말한 logical reasoning이 필요

  relations between Baltimore and U.S., relations between Eldersburg and U.S.

Introduction

Elias Brown
[1] Elias Brown (May 9, 1793– July 7, 1857) was a ***U.S.*** Representative from ***Maryland***. [2] Born near ***Baltimore***, ***Maryland***, Brown attended the common schools. … [7] He died near ***Baltimore***, ***Maryland***, and is interred in a private cemetery near ***Eldersburg***, ***Maryland***.

- 아래와 같은, Intra-sentence relations는 쉽게 recognize할 수 있다

  (Maryland, country, U.S.), (Baltimore, located in the administrative territorial entity, Maryland), (Eldersburg, located in the administrative territorial entity, Maryland)

- 하지만 아래와 같은, 한 sentence에 있지 않고 long-distance dependencies를 가지는 경우 앞서 말한 logical reasoning이 필요

  relations between Baltimore and U.S., relations between Eldersburg and U.S.

(Maryland, country, U.S.)이고, (Eldersburg, located in the administrative territorial entity, Maryland)이니까,

이 둘을 이용하여 (Eldersburg, country, U.S.)로 reasoning할 수 있다.

Introduction

- DocRED dataset을 사용

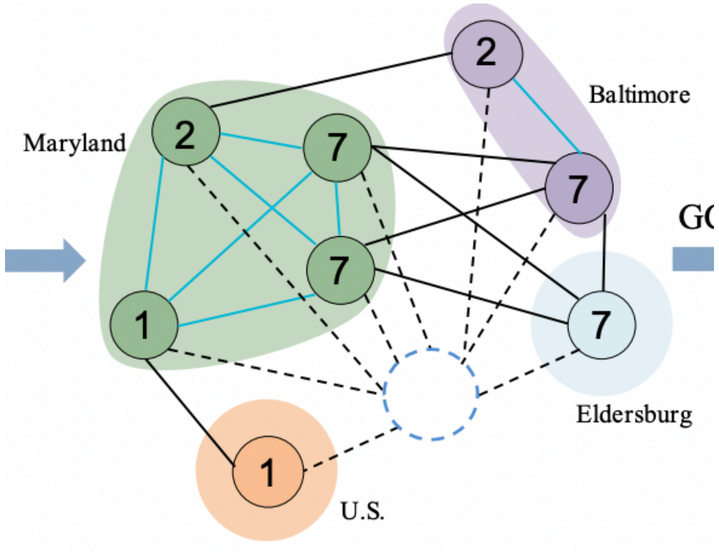
- **G**raph **A**ggregation-and **I**nference **N**etwork **(GAIN)** for document-level relation extraction 제안

  - GAIN은 (hMG)를 구축

  - 여기서, hMG는 2가지 type의 node와 3가지 타입의 edge로 구성되어 있다

nodes
- mention node
- document node

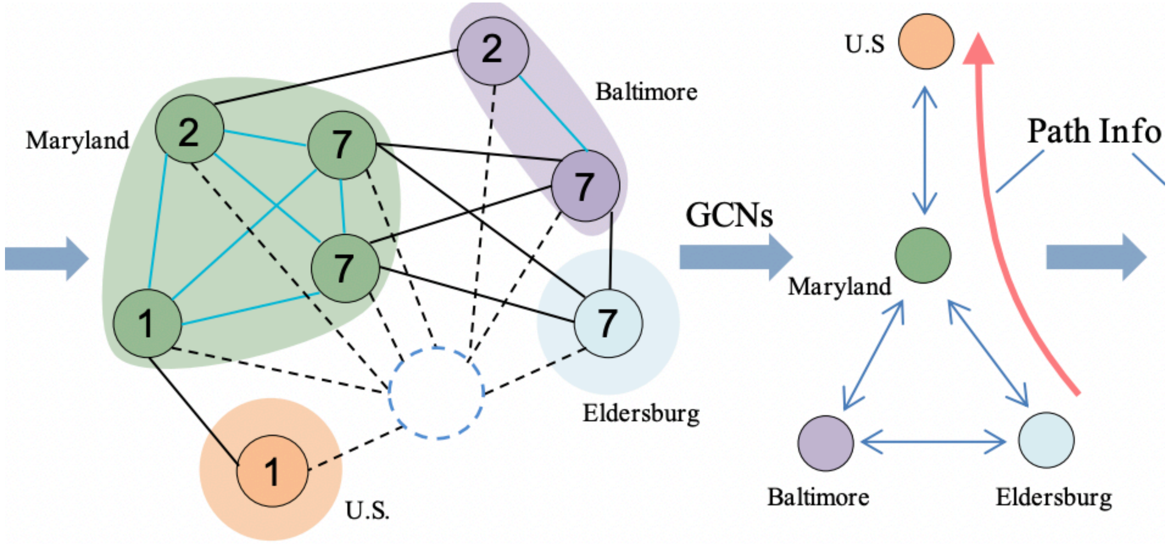
edges
- Intra-entity edge
- Inter-entity edge
- Document edge



Introduction

- hMG로 document내의 서로 다른 mention간의 relation을 modeling
- hMG에 Graph Convolution Network(GCN)을 적용하여 각 mention에 대한 document-aware representation을 얻음
- hMG에서 같은 entity를 참조한 mention들을 합쳐서 Entity-level Graph(EG) 구축

위 과정을 통해 entities 간의 relation을 reasoning하는 새로운 path reasoning mechanism 제안



Task formulation

- the task aims to extract the relations between different entities in $\mathcal{E}$

$$\boxed{\{(e_i, r_{ij}, e_j) | e_i, e_j \in \mathcal{E}, r_{ij} \in \mathcal{R}\}}$$

$$\mathcal{E} = \{e_i\}_{i=1}^{P}$$

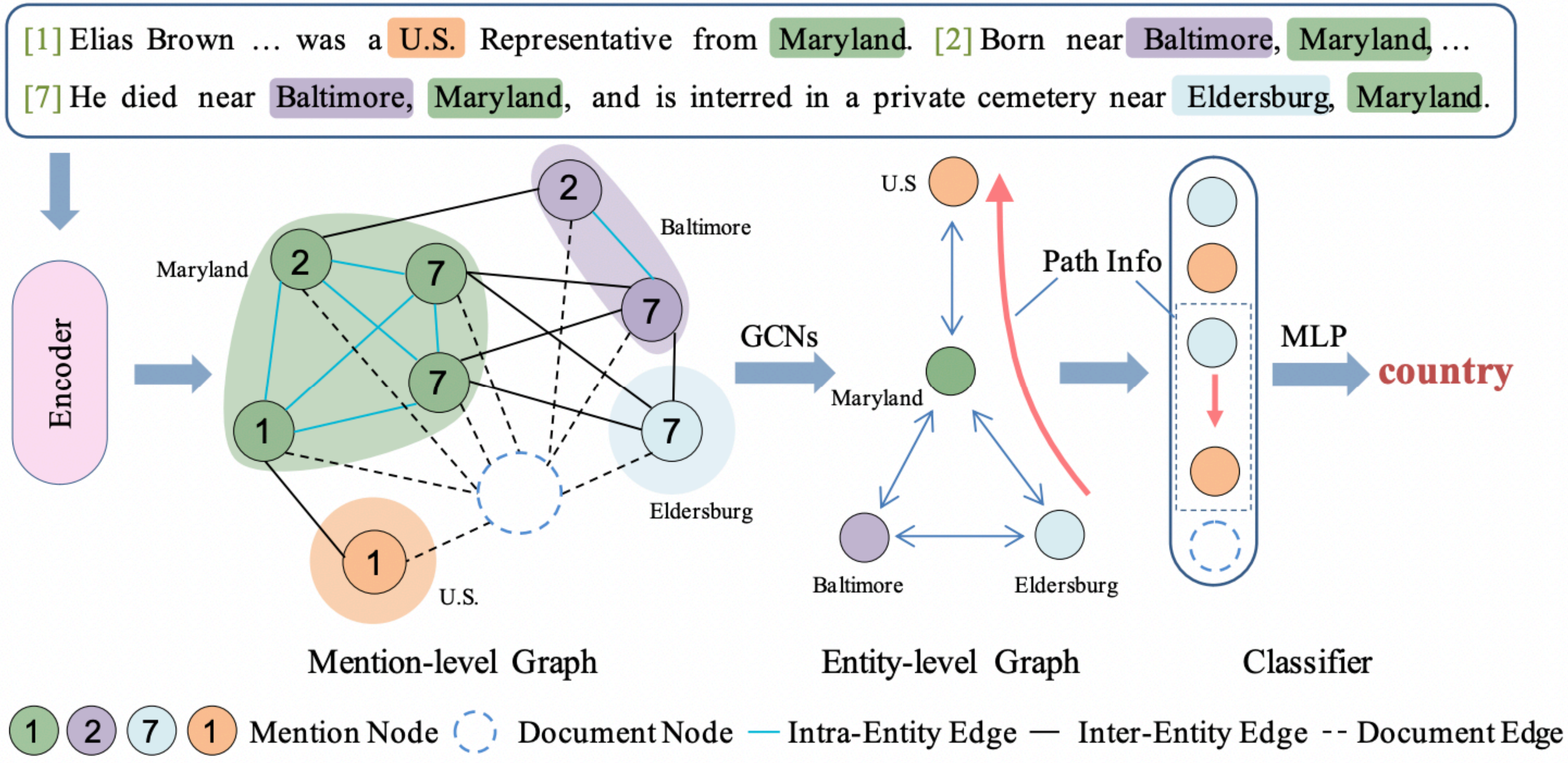
$$e_i = \{m_j\}_{j=1}^{Q}$$

m_j : span of words belonging to the j-th mention of the i-th entity

$\mathcal{R}$: a pre-defined relation type set.

Graph Aggregation and Inference Network (GAIN)

- mainly consists of 4 modules
  1. Encoding module
  2. Mention-level graph Aggregation Module
  3. Entity-level Graph Inference Module
  4. Classification Module



Graph Aggregation and Inference Network (GAIN)

1. Encoding Module

$\mathcal{D} = \{w_i\}_{i=1}^{n}$ 인 document 의 각 단어에 대해 word embedding, entity type embedding, coreference embedding을 적용하고 각 embedding값을 concat

$$x_i = [E_w(w_i); E_t(t_i); E_c(c_i)]$$

이 벡터를 encoder에 넣어 context sensitive representation을 얻음

$$[g_1, g_2, \ldots, g_n] = Encoder([x_1, x_2, \ldots, x_n])$$

→ 이 때 이 encoder는 lstm과 같은 모델이 될 수 있다

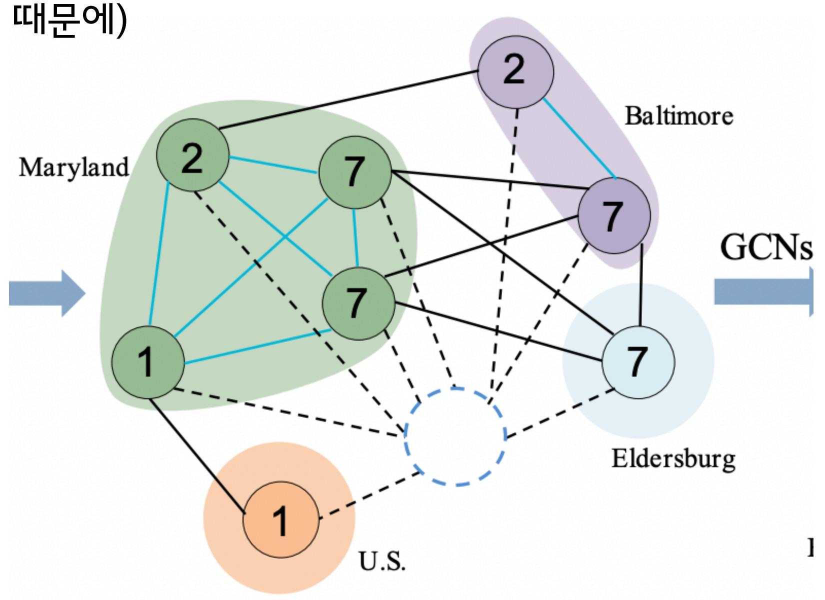
Graph Aggregation and Inference Network (GAIN)

2. Mention-level Graph Aggregation Module

: document-level information과 mentions와 entities 사이의 interactions를 modeling하기 위해 heterogeneous Mention-level Graph(hMG) 구축

nodes
- mention node : 각 mention node는 entity에 대한 particular mention을 의미
- document node : 전제 document 정보를 modeling하기 위한 node → 1개
  - → 다른 mention들과 interact하여 document안에서 mention들간의 long distance를 줄이는 역할 (Document node가 피봇일 때, mention node 간 거리에 최대 2이기 때문에)

edges
- Intra-entity edge
  : 동일한 entity를 언급하는 mention끼리 연결
- Inter-entity edge
  : 서로 다른 entity에 대한 mention이 한 문장 내에 발생할 경우 연결
- Document edge
  : 문서 내 모든 mention과 해당 document node간 연결



Mention-level Graph

Graph Aggregation and Inference Network (GAIN)

2. Mention-level Graph Aggregation Module

: neighbor들의 feature들을 합치기 위해 hMG에 Graph Convolution Network(GCN)을 적용

: Given node *u* at the *l*-th layer, the graph convolutional operation can be defined as:

$$h_u^{(l+1)} = \underline{\sigma}\left(\sum_{k\in\mathcal{K}}\sum_{v\in\mathcal{N}_k(u)} W_k^{(l)} h_v^{(l)} + b_k^{(l)}\right)$$

→ Activation function(e.g., ReLU)

$\mathcal{K}$: different types of edges

$W_k^{(l)} \in \mathbb{R}^{d\times d}$, $b_k^{(l)} \in \mathbb{R}^d$ → Trainable param

$\mathcal{N}_k(u)$ → neighbors for node u connected in k-th type edge

: GCN에서 서로 다른 layer는 서로 다른 level의 feature를 표현하므로, 각 node u에서 final representation을 표현하기 위해 각 layer에서 hidden states들을 concat한다.

$$\mathbf{m}_u = [h_u^{(0)}; h_u^{(1)}; \ldots; h_u^{(N)}]$$

→ 이를 통해 neighbors로부터의 feature를 통합할 수 있고, 각 mention에 대한 representation을 얻을 수 있음

Graph Aggregation and Inference Network (GAIN)

3. Entity-level Graph Inference Module

: 같은 entity를 언급하고 있는 mention은 하나의 entity node로 합쳐진다

: 여기서, document node는 고려하지 않음

: i-th entity node ei가 N번 mention됐다면, 아래와 같이 평균으로 ei를 표현한다.

$$\mathbf{e}_i = \frac{1}{N}\sum_{n}\mathbf{m}_n$$

: inter-entity edge들을 EG내의 edge로

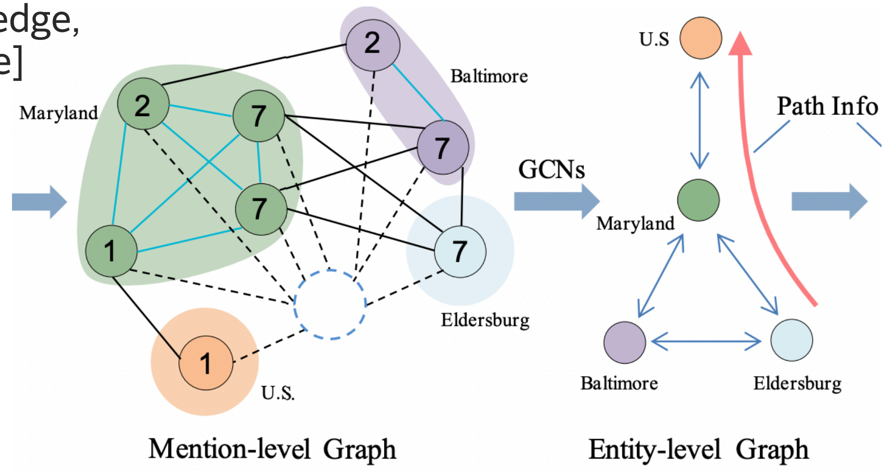
: EG에서 head entity와 tail entity간의 path는 중간에 거치는 through entity를 포함하여 아래와 같이 표현

$$\mathbf{p}^i_{h,t} = [\mathbf{e}_{ho}; \mathbf{e}_{ot}; \mathbf{e}_{to}; \mathbf{e}_{oh}]$$

→ [head와 through 간 edge, through와 tail 간 edge, tail과 through 간 edge, through와 head 간 edge]

: 많은 path중에서 어떤 path가 중요한지에 대해 attention을 이용하여 학습

(entity pair $(\mathbf{e}_h, \mathbf{e}_t)$를 query로 하여서)

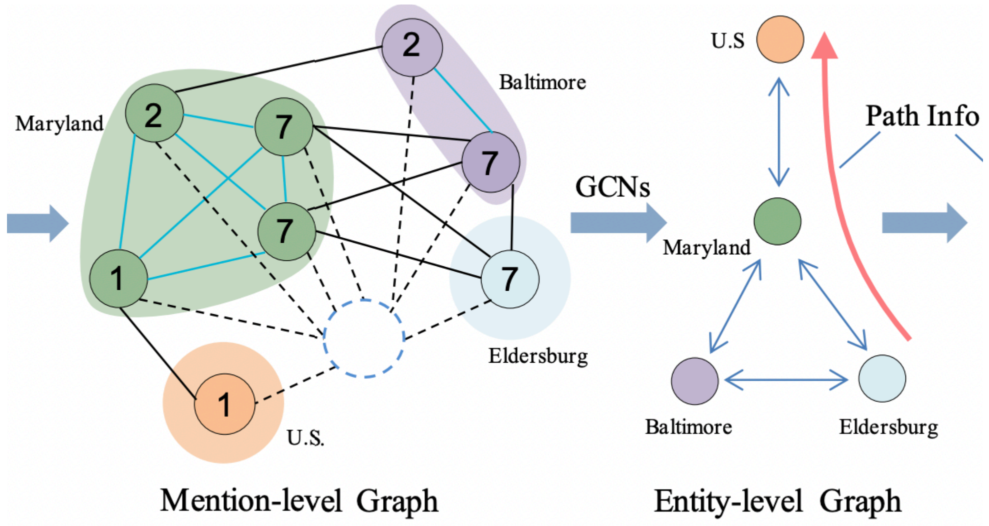


Graph Aggregation and Inference Network (GAIN)

3. Entity-level Graph Inference Module

   : 이 module을 통해

   - Multiple sentences에 퍼져있는 mention들의 정보를 잘 융합하여 entity를 나타낼 수 있다.
   - Attention mechanism을 통해 relation 예측 시 logical reasoning chains를 고려할 수 있다.



Graph Aggregation and Inference Network (GAIN)

4. Classification Module

: task as multi-label classification task and predict relations between entities:

$$P(r|\mathbf{e}_h, \mathbf{e}_t) = sigmoid\left(W_b \sigma(W_a \underline{I_{h,t}} + b_a) + b_b\right)$$

$$I_{h,t} = [\mathbf{e}_h; \mathbf{e}_t; |\mathbf{e}_h - \mathbf{e}_t|; \underbrace{\mathbf{e}_h \odot \mathbf{e}_t}_{\text{head와 tail의 element-wise 곱}}; \overbrace{\mathbf{m}_{doc}}^{\text{document node}}; \underbrace{\mathbf{p}_{h,t}}_{\text{head와 tail간의 attention이 적용된 경로}}]$$

: binary cross entropy 사용

$$\mathcal{L} = -\sum_{\mathcal{D}\in\mathcal{S}} \sum_{h \neq t} \sum_{r_i \in \mathcal{R}} \mathbb{I}(r_i = 1) \log P(r_i|\mathbf{e}_h, \mathbf{e}_t) + \mathbb{I}(r_i = 0) \log(1 - P(r_i|\mathbf{e}_h, \mathbf{e}_t))$$

Experiments

- Dateset

: DocRED(a large-scale human-annotated dataset for document-level RE constructed from Wikipedia and Wikidata.)

graph-based

Model	Dev				Test	
	Ign F1	**Ign AUC**	**F1**	**AUC**	**Ign F1**	**F1**
CNN* (Yao et al., 2019)	41.58	36.85	43.45	39.39	40.33	42.26
LSTM* (Yao et al., 2019)	48.44	46.62	50.68	49.48	47.71	50.07
BiLSTM* (Yao et al., 2019)	48.87	47.61	50.94	50.26	48.78	51.06
Context-Aware* (Yao et al., 2019)	48.94	47.22	51.09	50.17	48.40	50.70
HIN-GloVe* (Tang et al., 2020)	51.06	-	52.95	-	51.15	53.30
GAT‡ (Velickovic et al., 2017)	45.17	-	51.44	-	47.36	49.51
GCNN‡ (Sahu et al., 2019)	46.22	-	51.52	-	49.59	51.62
EoG‡ (Christopoulou et al., 2019)	45.94	-	52.15	-	49.48	51.82
AGGCN‡ (Guo et al., 2019)	46.29	-	52.47	-	48.89	51.45
LSR-GloVe* (Nan et al., 2020)	48.82	-	55.17	-	52.15	54.18
GAIN-GloVe	**53.05**	**52.57**	**55.29**	**55.44**	**52.66**	**55.08**

Experiments

- Dateset

: DocRED(a large-scale human-annotated dataset for document-level RE constructed from Wikipedia and Wikidata.)

BERT-RE$^{*}_{base}$ (Wang et al., 2019a)	-	-	54.16	-	-	53.20
RoBERTa-RE$^{\dagger}_{base}$	53.85	48.27	56.05	51.35	53.52	55.77
BERT-Two-Step$^{*}_{base}$ (Wang et al., 2019a)	-	-	54.42	-	-	53.92
HIN-BERT$^{*}_{base}$ (Tang et al., 2020)	54.29	-	56.31	-	53.70	55.60
CorefBERT-RE$^{*}_{base}$ (Ye et al., 2020)	55.32	-	57.51	-	54.54	56.96
LSR-BERT$^{*}_{base}$ (Nan et al., 2020)	52.43	-	59.00	-	56.97	59.05
GAIN-BERT$_{base}$	**59.14**	**57.76**	**61.22**	**60.96**	**59.00**	**61.24**
BERT-RE$^{*}_{large}$ (Ye et al., 2020)	56.67	-	58.83	-	56.47	58.69
CorefBERT-RE$^{*}_{large}$ (Ye et al., 2020)	56.73	-	58.88	-	56.48	58.70
RoBERTa-RE$^{*}_{large}$ (Ye et al., 2020)	57.14	-	59.22	-	57.51	59.62
CorefRoBERTa-RE$^{*}_{large}$ (Ye et al., 2020)	57.84	-	59.93	-	57.68	59.91
GAIN-BERT$_{large}$	**60.87**	**61.79**	**63.09**	**64.75**	**60.31**	**62.76**

참고 블로그
: https://kabbi159.medium.com/emnlp-2020-double-graph-based-reasoning-for-document-level-relation-extraction-%EB%85%BC%EB%AC%B8-%EC%A0%95%EB%A6%AC-7bb0dd5f65f3