Double Graph Based Reasoning for Document-level Relation Extraction

이봉석

- **Author**
  - Shuang Zeng
  - Runxin Xu
  - Baobao Chang and Lei Li

- **Title of Conference(Journal)**
  - EMNLP 2020

01. 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***.

Subject: ***Maryland***
Object: ***U.S.***
relation: **country**

Subject: ***Baltimore***；***Eldersburg***
Object: ***Maryland***
relation: **located in the administrative territorial entity**

Subject: ***Baltimore***；***Eldersburg***
Object: ***U.S.***
relation: **country**

Figure 1: An example document and its desired relations from DocRED (Yao et al., 2019). Entity mentions and relations involved in these relation instances are colored. Other mentions are underlined for clarity.

01. Introduction

기존의 RE에서는 sentence-level RE였다.

이런 기존 방식의 RE의 경우 여러 문장에서의 entity들 사이의 관계를 알 수 없다.

따라서 document-level의 RE는 텍스트 안에 있는 지식에 대한 전체적인 이해를 위해서 필요하다.

Document-level의 RE를 위해서는 몇가지 주요한 challenges가 있다.

- **관계의 entities들이 다른 문장에서 나타날 수 있다. 그러므로 한 문장만으로는 관계를 식별할 수 없다.**
- **동일한 entity가 다른 문장 들에서 여러 번 언급될 수 있다. 그래서 entity를 더 잘 표현하기 위해서는 문장 들간의 정보를 잘 모아야한다.**
- **많은 관계를 식별하는데 논리적 추론 기술이 필요하다. (기존에는 한 문장에 하나의 relation이였지만 지금은 한 document에 많은 relation이 존재)**

01. 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*.

Subject: *Maryland*
Object: *U.S.*
relation: **country**

Subject: *Baltimore*; *Eldersburg*
Object: *Maryland*
relation: **located in the administrative territorial entity**

Subject: *Baltimore*; *Eldersburg*
Object: *U.S.*
relation: **country**

Figure 1: An example document and its desired relations from DocRED (Yao et al., 2019). Entity mentions and relations involved in these relation instances are colored. Other mentions are underlined for clarity.

첫째와 둘째 relation의 경우 subject entity와 object entity가 한 문장 안에 있기때문에 intra-sentence relations 찾기가 쉽다.

하지만 마지막 relation은 같은 문장에서 subject entity와 object entity가 언급되지 않아 long-distance dependency를 가지기 때문에 inter-sentence relations 찾기 힘들다.
게다가 여기에는 Eldersburg는 미국에 속한 Maryland에 있기 때문에 미국에 속한다는 논리적 추론을 요구한다.

이런 문제를 풀 수 있는 Graph Aggregation and Inference Network(GAIN)을 제안한다.

02. Task Formulation

$\mathcal{D} = \{s_i\}_{i=1}^{N}$ N개의 sentence를 가진 document

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

$s_i = \{w_j\}_{j=1}^{M}$ i-th문장은 M개의 단어로 구성

$e_i = \{m_j\}_{j=1}^{Q}$ i-th entity의 mention span

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

$S_{e_i} \cap S_{e_j} \neq \varnothing$ inter - relation

$S_{e_i} \cap S_{e_j} = \varnothing$ intra - relation

03. Model

GAIN은 총 4개의 모듈로 구성되어 있다.

1. **Encoding module**
2. **Mention-level graph aggregation module**
3. **Entity-level inference module**
4. **Classification module**

03. Model

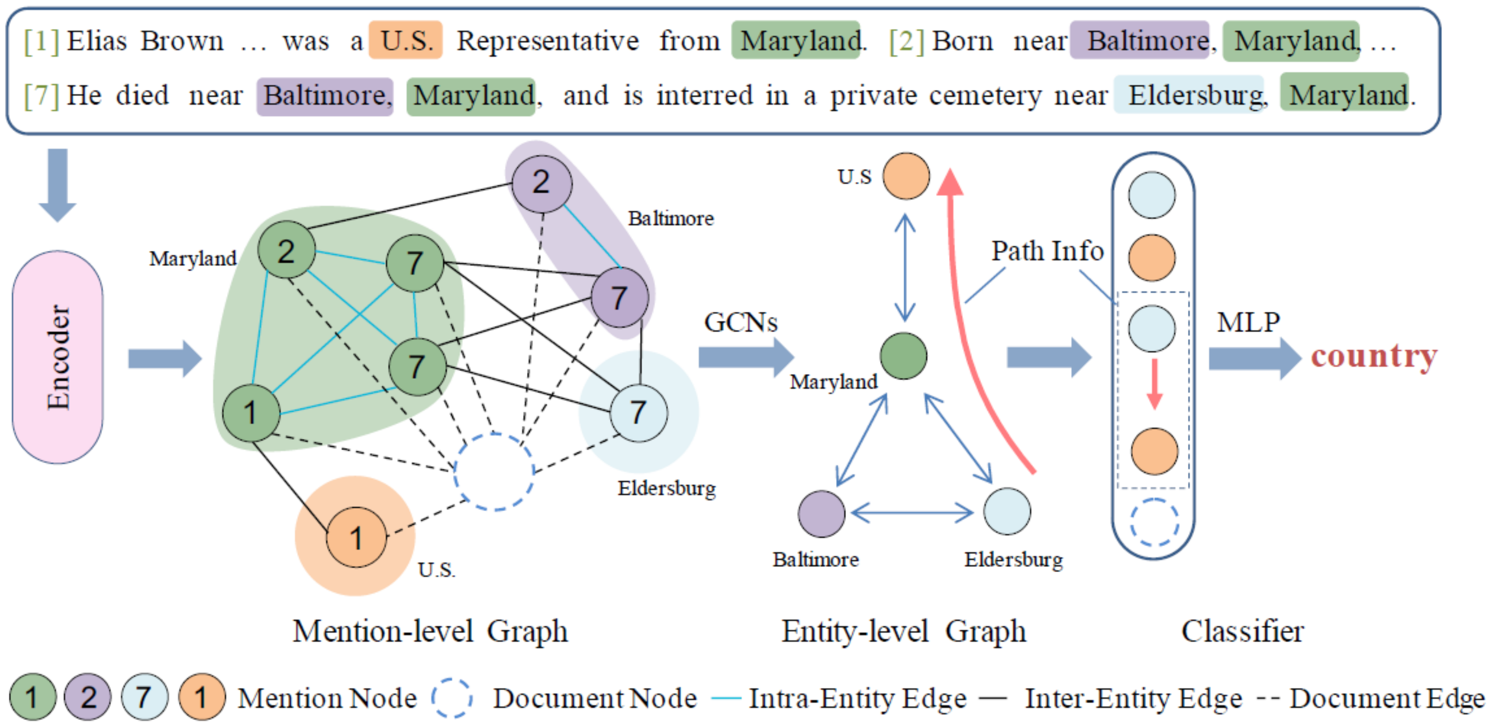


Figure 2: The overall architecture of GAIN. First, A context encoder consumes the input document to get a contextualized representation of each word. Then, the Mention-level Graph is constructed with mention nodes and a document node. After applying GCN, the graph is transformed into Entity-level Graph, where the paths between entities are identified for reasoning. Finally, the classification module predicts target relations based on the above information. Different entities are in different colors. The number i in the mention node denotes that it belongs to the i-th sentence.

03. Model(Encoding Module)

$$\mathcal{D} = \{w_i\}_{i=1}^{n}$$

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

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

- **$E_w()$, $E_t()$, $E_c()$ 는 각각 word embedding layer, entity type (e.g., PER, ORG) embedding layer, coreference embedding layer이다.**
- **t_i와 c_i는 각각 named entity type이고 entity id이다. 그리고 그 단어가 entity가 아니면 None entity type으로 한다.**

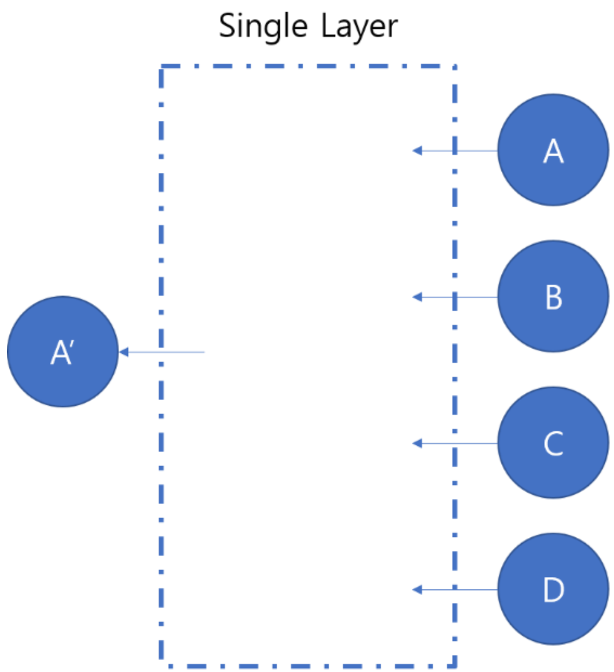
Related Work

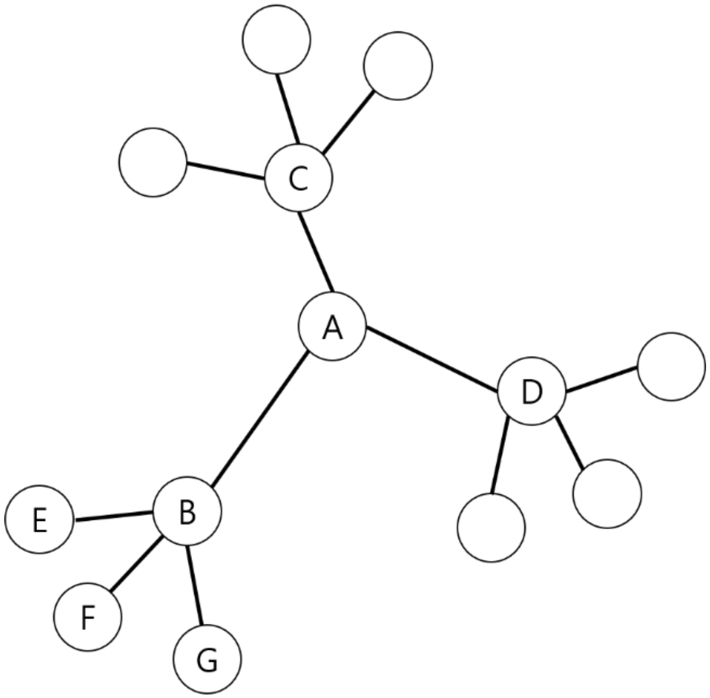
GNN (Graph Neural Network)는 그래프 구조에서 사용되는 인공 신경망을 말한다. 우리가 흔히 알고 있는 Fully-connected network, CNN, RNN은 보통 벡터나 행렬 형태로 input이 주어지는데 GNN의 경우 <u>input이 그래프 구조</u>라는 특징이 있습니다.

GNN은 입력으로 그래프 구조와 각 노드 별 feature 정보를 받습니다. 입력으로 받은 feature 정보와 그래프 내에서 나타내는 이웃 정보를 바탕으로 각 노드 별 feature를 출력 결과로 얻어낸다.

GNN은 하나의 레이어에서 각 노드들은 그래프 상의 이웃들의 정보와 자기 자신의 정보를 이용해 feature를 만든다.

Related Work

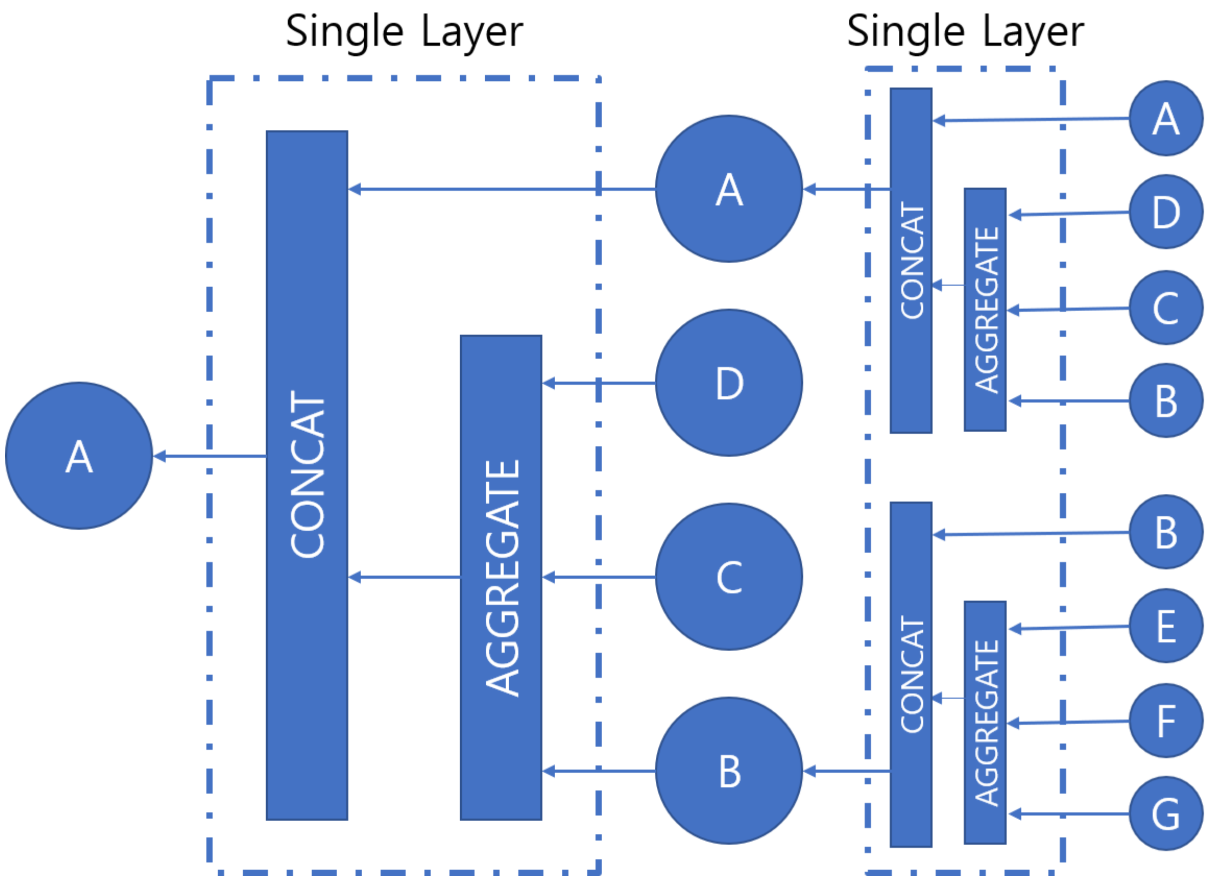




GNN에서는 어떻게 자기 자신의 정보와 이웃들의 정보를 합쳐서 feature를 구할까?

대부분의 GNN은 먼저 이웃들의 정보를 모으고, 모든 정보를 통해 얻은 새로운 값과 이전 상태의 자기 자신의 값을 이용해서 만든다.

Related Work



즉, GNN은 AGGREGATE와 CONCAT 함수를 정의하고 이 함수의 parameter를 학습한다.

Related Work

GCN

그래프 G = (A, X)와 같이 정의되며, $A \in R^{N \times N}$ 는 각 node의 연결을 나타내는 인접 행렬 (Adjacency matrix)이고 $X \in R^{N \times d}$는 node feature matrix이다. 이 때, N은 그래프에 포함된 node의 수, d는 node feature vector의 차원이다.

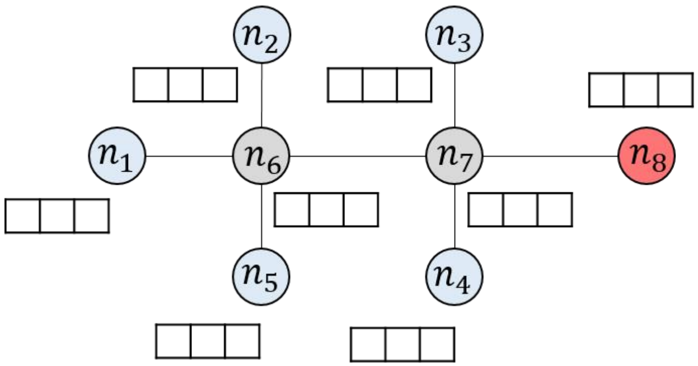
그래프에 대한 convolution ψ는 A와 X를 입력 받아 $H \in R^{N \times m}$라는 새로운 latent node feature matrix를 생성한다. 이 때, m은 latent feature vector의 차원이다. 가장 기본적인 convolution은 다음과 같다.

$$H = \psi(A, X) = \sigma(AXW)$$

$W \in R^{d \times m}$ 는 학습이 가능한 weight matrix, σ 는 sigmoid나 ReLU와 같이 비선형적인 출력을 생성하기 위한 non-linear activation function

Related Work

H = ψ(A, X) = σ(AXW)



$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

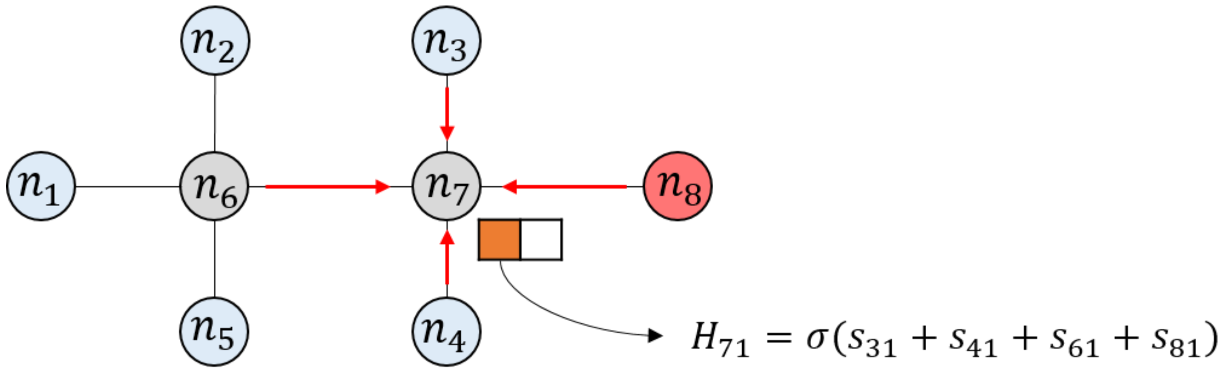
d = 3라고 가정

$$S = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{81} & x_{82} & x_{83} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} w_{i1}x_{1i} & \sum_{i=1}^{3} w_{i2}x_{1i} \\ \sum_{i=1}^{3} w_{i1}x_{2i} & \sum_{i=1}^{3} w_{i2}x_{2i} \\ \vdots & \vdots \\ \sum_{i=1}^{3} w_{i1}x_{8i} & \sum_{i=1}^{3} w_{i2}x_{8i} \end{bmatrix}$$

m = 2라고 가정

Related Work

H = ψ(A, X) = σ(AXW)



한계점

- **A에는 neighbor node와의 연결만 표현되어 있기 때문에 graph convolution 과정에서 해당 node 자체에 대한 정보는 latent feature vector를 만들 때 고려되지 않는다.**
- **일반적으로 A는 정규화 되어 있지 않기 때문에(그 해당 노드에 몇 개의 노드가 연결되어 있는지를 고려 X) feature vector와 A를 곱할 경우 feature vector의 크기가 불안정하게 변할 수 있다.**

Related Work

H = ψ(A, X) = σ(AXW)

두 가지 문제점을 해결하기 위해 GCN을 구현할 때는 A에 self-loop를 추가하고, A를 D−1/2AD−1/2로 정규화 한다.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

- $H^{(l)}$: l번째 layer의 Hidden state이며, $H^0 = X$(그래프 노드의 초기 feature) 입니다.
- $\tilde{A}$: $A + I_N$ 으로, 인접행렬(A)에 자기 자신으로의 연결(I_N)을 추가한 것입니다.
- $\tilde{D}$: $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ 로, 각 노드의 degree를 나타내는 대각 행렬입니다.
- $W^{(l)}$: l번째 layer의 학습가능한 parameter입니다.
- σ: 비선형 함수로 $ReLU(\cdot)$를 이용했습니다.

03. Model(Mention-level Graph Aggregation Module)

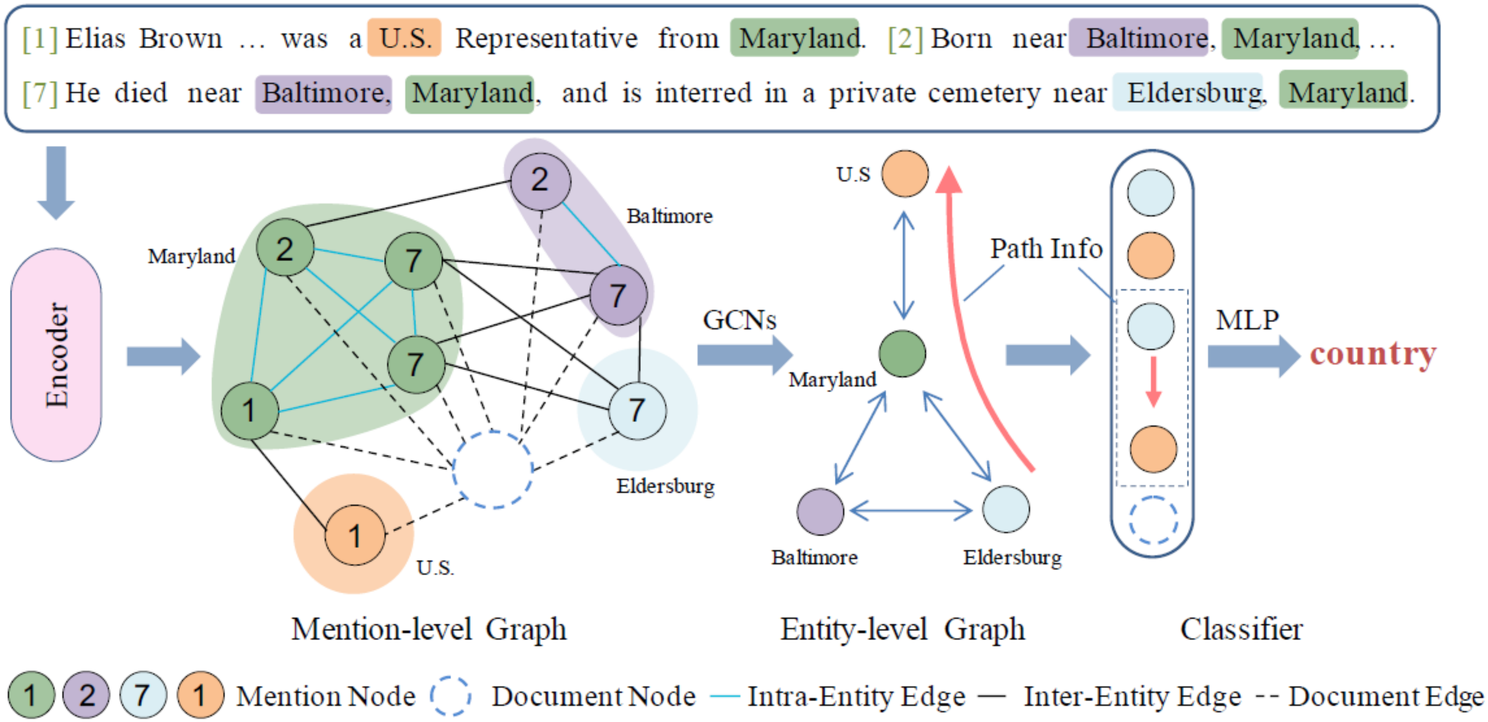


Figure 2: The overall architecture of GAIN. First, A context encoder consumes the input document to get a contextualized representation of each word. Then, the Mention-level Graph is constructed with mention nodes and a document node. After applying GCN, the graph is transformed into Entity-level Graph, where the paths between entities are identified for reasoning. Finally, the classification module predicts target relations based on the above information. Different entities are in different colors. The number i in the mention node denotes that it belongs to the i-th sentence.

03. Model(Mention-level Graph Aggregation Module)

- **heterogeneous Mention-level Graph (hMG)**
- **nodes :**

  - Mention node: 각 엔티티들에 대한 멘션 노드

  - Document node: document의 전체 정보를 담고있는 노드

- **edges:**

  - Intra-Entity Edge: 같은 엔티티를 언급하는 노드끼리 연결

  - Inter-Entity Edge: 한 문장안에 같이 있는 다른 엔티티들 연결

  - Document Edge: 모든 멘션 노드를 document node와 연결

03. Model(Mention-level Graph Aggregation Module)

$$h_u^{(0)} = \frac{1}{t-s+1} \sum_{j=s}^{t} g_j$$

For a mention ranging from the s-th word to the t-th word in the document.

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

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

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

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

03. Model(Entity-level Graph Inference Module)

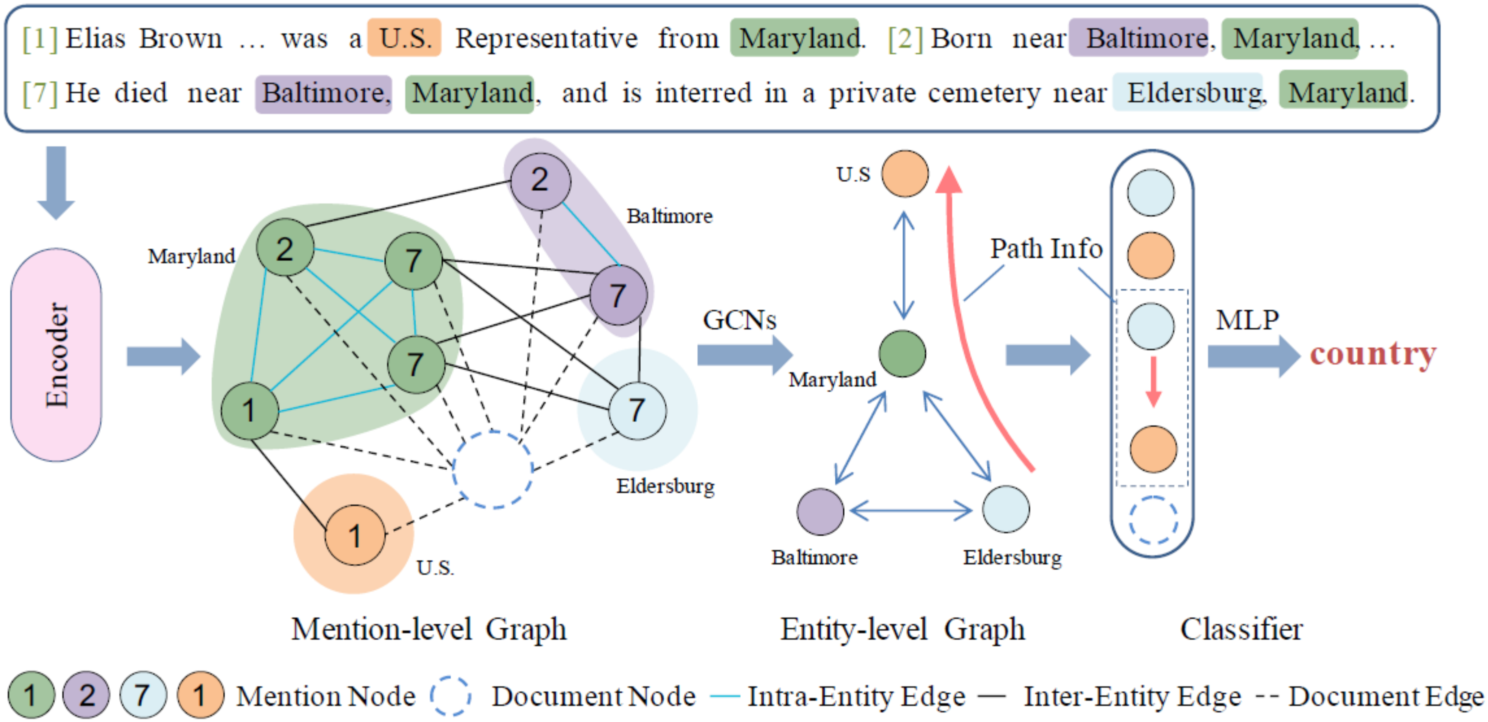


Figure 2: The overall architecture of GAIN. First, A context encoder consumes the input document to get a contextualized representation of each word. Then, the Mention-level Graph is constructed with mention nodes and a document node. After applying GCN, the graph is transformed into Entity-level Graph, where the paths between entities are identified for reasoning. Finally, the classification module predicts target relations based on the above information. Different entities are in different colors. The number i in the mention node denotes that it belongs to the i-th sentence.

03. Model(Entity-level Graph Inference Module)

➢ Entity-level Graph 생성

- ➢ **nodes** **같은 엔티티들을 통합해서 1개의 노드로 만든다** $$\mathbf{e}_i = \frac{1}{N}\sum_n \mathbf{m}_n$$

- ➢ **edges** **그전 그래프에서 inter-entity edges를 통합**

03. Model(Entity-level Graph Inference Module)

directed edge(e_i에서 e_j까지)

$$\mathbf{e}_{ij} = \sigma\left(W_q[\mathbf{e}_i; \mathbf{e}_j] + b_q\right)$$

헤드 엔티티 e_h와 테일 엔티티 e_t사이의 i 번째 경로(여기서는 two-hop path)

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

$$s_i = \sigma([\mathbf{e}_h; \mathbf{e}_t] \cdot W_l \cdot \mathbf{p}^i_{h,t})$$

$$\alpha_i = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

$$\mathbf{p}_{h,t} = \sum_i \alpha_i \mathbf{p}^i_{h,t}$$

03. Model(Classification Module)

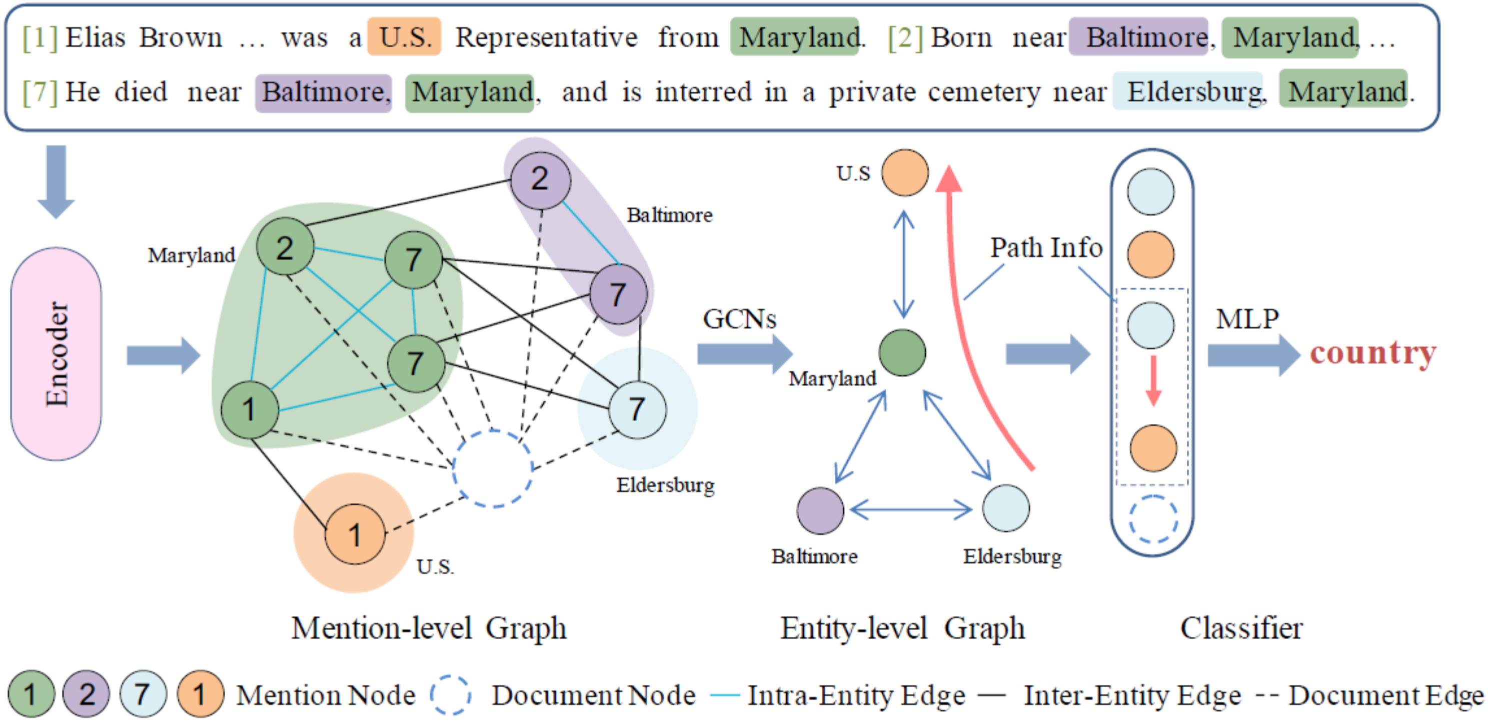


Figure 2: The overall architecture of GAIN. First, A context encoder consumes the input document to get a contextualized representation of each word. Then, the Mention-level Graph is constructed with mention nodes and a document node. After applying GCN, the graph is transformed into Entity-level Graph, where the paths between entities are identified for reasoning. Finally, the classification module predicts target relations based on the above information. Different entities are in different colors. The number i in the mention node denotes that it belongs to the i-th sentence.

03. Model(Classification Module)

- **Classification**

$$I_{h,t} = [\mathbf{e}_h; \mathbf{e}_t; |\mathbf{e}_h - \mathbf{e}_t|; \mathbf{e}_h \odot \mathbf{e}_t; \mathbf{m}_{doc}; \mathbf{p}_{h,t}]$$

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

- **Loss function**

(binary cross entropy)

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

where $\mathcal{S}$ denotes the whole corpus, and $\mathbb{I}(\cdot)$ refers to indication function.

04. Experiment

➢ Relation Extraction

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**
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**

Table 2: Performance on DocRED. Models above the first double line do not use pre-trained model. Results with * are reported in their original papers. Results with ‡ are performances of graph-based state-of-the-art RE models implemented in (Nan et al., 2020). Results with † are based on our implementation.

04. Experiment

➢ Ablation Study

Model	Dev				Test	
	Ign F1	Ign AUC	F1	AUC	Ign F1	F1
GAIN-GloVe	**53.05**	**52.57**	**55.29**	**55.44**	**52.66**	**55.08**
- *hMG*	50.97	48.84	53.10	51.73	50.76	53.06
- *Inference Module*	50.84	48.68	53.02	51.58	50.32	52.66
- *Document Node*	50.86	48.68	53.01	52.46	50.32	52.67
GAIN-BERT$_{base}$	**59.14**	**57.76**	**61.22**	**60.96**	**59.00**	**61.24**
- *hMG*	57.12	51.54	59.17	54.61	57.31	59.56
- *Inference Module*	56.97	54.29	59.28	57.25	57.01	59.34
- *Document Node*	57.26	52.07	59.62	55.51	57.01	59.63

Table 3: Performance of GAIN with different embeddings and submodules.

04. Experiment

➢ Ablation Study

Model	Intra-F1	Inter-F1
CNN*	51.87	37.58
LSTM*	56.57	41.47
BiLSTM*	57.05	43.49
Context-Aware*	56.74	42.26
LSR-GloVe*	60.83	48.35
GAIN-GloVe	**61.67**	**48.77**
- *hMG*	59.72	46.49
BERT-RE$^*_{base}$	61.61	47.15
RoBERTa-RE$_{base}$	65.65	50.09
BERT-Two-Step$^*_{base}$	61.80	47.28
LSR-BERT$^*_{base}$	65.26	52.05
GAIN-BERT$_{base}$	**67.10**	**53.90**
- *hMG*	66.15	51.42

Table 4: Intra- and Inter-F1 results on dev set of DocRED. Results with * are reported in (Nan et al., 2020).

Model	Infer-F1	P	R
CNN	37.11	32.81	42.72
LSTM	39.03	33.16	47.44
BiLSTM	38.73	31.60	50.01
Context-Aware	39.73	**33.97**	47.85
GAIN-GloVe	**40.82**	32.76	**54.14**
- *Inference Module*	39.76	32.26	51.80
BERT-RE$_{base}$	39.62	34.12	47.23
RoBERTa-RE$_{base}$	41.78	37.97	46.45
GAIN-BERT$_{base}$	**46.89**	**38.71**	**59.45**
- *Inference Module*	45.11	36.91	57.99

Table 5: Infer-F1 results on dev set of DocRED. P: Precision, R: Recall.

04. Experiment

➢ Case Study

[1] *The Eminem Show* is the fourth studio album by American rapper *Eminem*, released on *May 26, 2002* by Aftermath Entertainment, Shady Records, and Interscope Records.
[2] *The Eminem Show* includes the commercially successful singles "*Without Me*", "Cleanin' Out My Closet", "Superman", and "Sing for the Moment"....

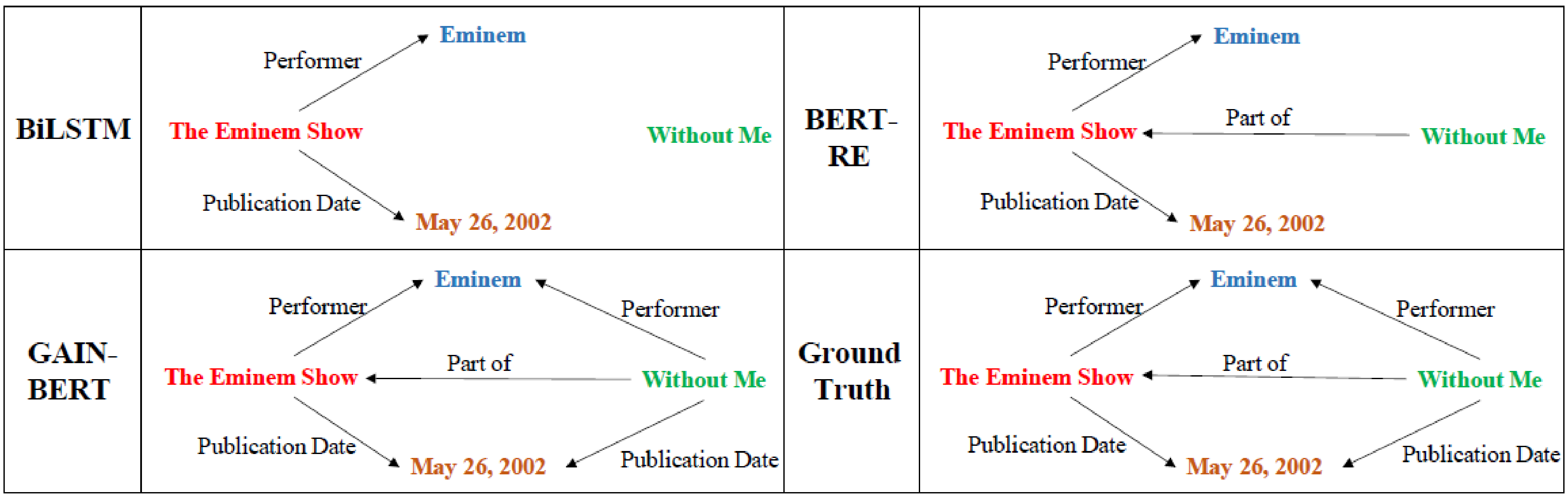


Figure 3: The case study of our proposed GAIN and baseline models. The models take the document as input and predict relations among different entities in different colors. We only show a part of entities within the documents and the according sentences due to the space limitation.

감사합니다