Learning Logic Rules for Document-level Relation Extraction

Dongyu Ru   Changzhi Sun   Jiangtao Feng   Lin Qiu

Hao Zhou   Weinan Zhang   Yong Yu   Lei Li

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY   ByteDance AI Lab 字节跳动人工智能实验室

Relation Extraction

Sentence-level RE

royalty of

engaged to

Britain ’s Prince Harry is engaged to his US partner Meghan Markle.

Relation Extraction

Sentence-level RE

royalty of

engaged to

Britain 's Prince Harry is engaged to his US partner Meghan Markle.

simple intra-sentence semantics (around 20 tokens per sentence)

fewer involved entities (2~3 on average) and relations (0~2 on average)

Relation Extraction

Document-level RE

royalty of

Britain 's Prince Harry is engaged to his US partner Meghan Markle. …

sibling of

Harry spent 10 years in the army and has this year, with his elderly brother William, …

royalty of

The last major royal wedding took place in 2011, when Kate Middleton and Prince William were married. …

spouse of

Relation Extraction

Document-level RE

royalty of

Britain 's Prince Harry is engaged to his US partner Meghan Markle. …

sibling of

Harry spent 10 years in the army and has this year, with his elderly brother William, …

royalty of

The last major royal wedding took place in 2011, when Kate Middleton and Prince William were married. …

spouse of

Longer context (hundreds of tokens) between entity mentions that may describe specific relations

More entities (around 20) and relations (>10 on average)

Main Challenges

- Long-range dependencies

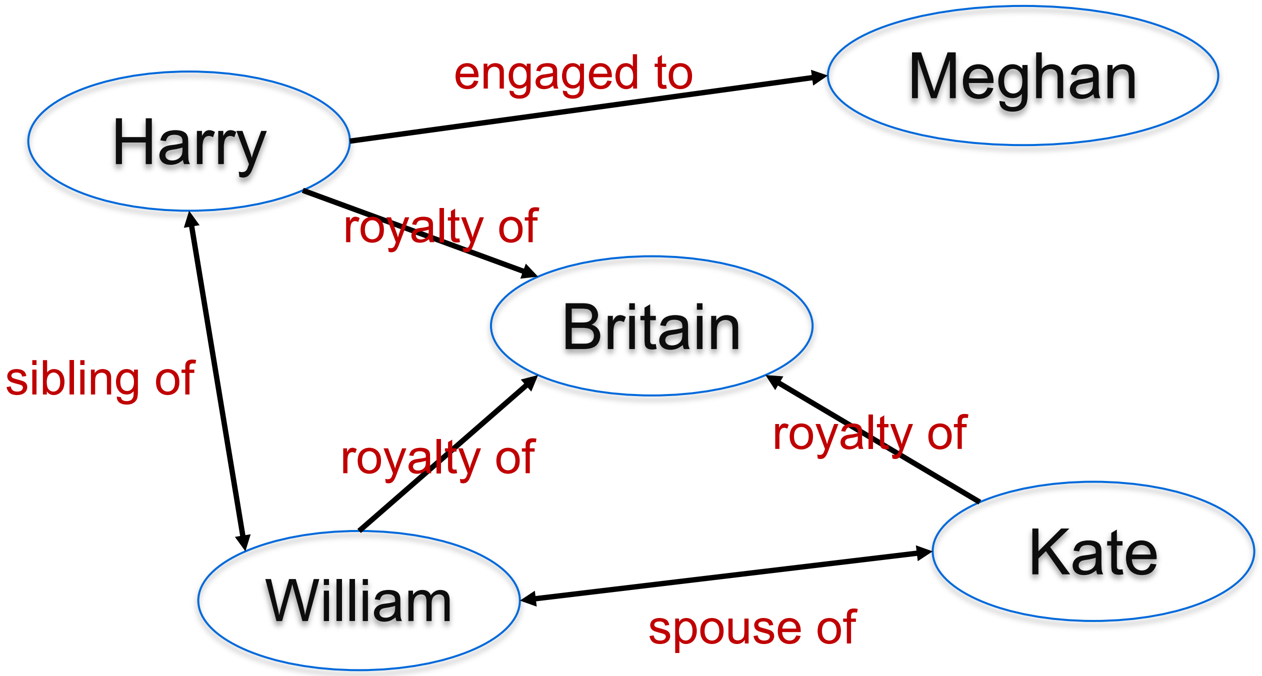
Britain's Prince Harry is engaged to his US partner Meghan Markle. …

Harry spent 10 years in the army and has this year, with his elderly brother William, …

The last major royal wedding took place in 2011, when Kate Middleton and Prince William were married. …

Main Challenges

- Complex interactions



An extracted subgraph from the previous document.

Existing Methods

- Sequence-based Approaches
  – Average pooling (e.g. DocRED (Yao et al., 2019)), Attentive pooling (e.g. ATLOP (Zhou et al., 2021)), …

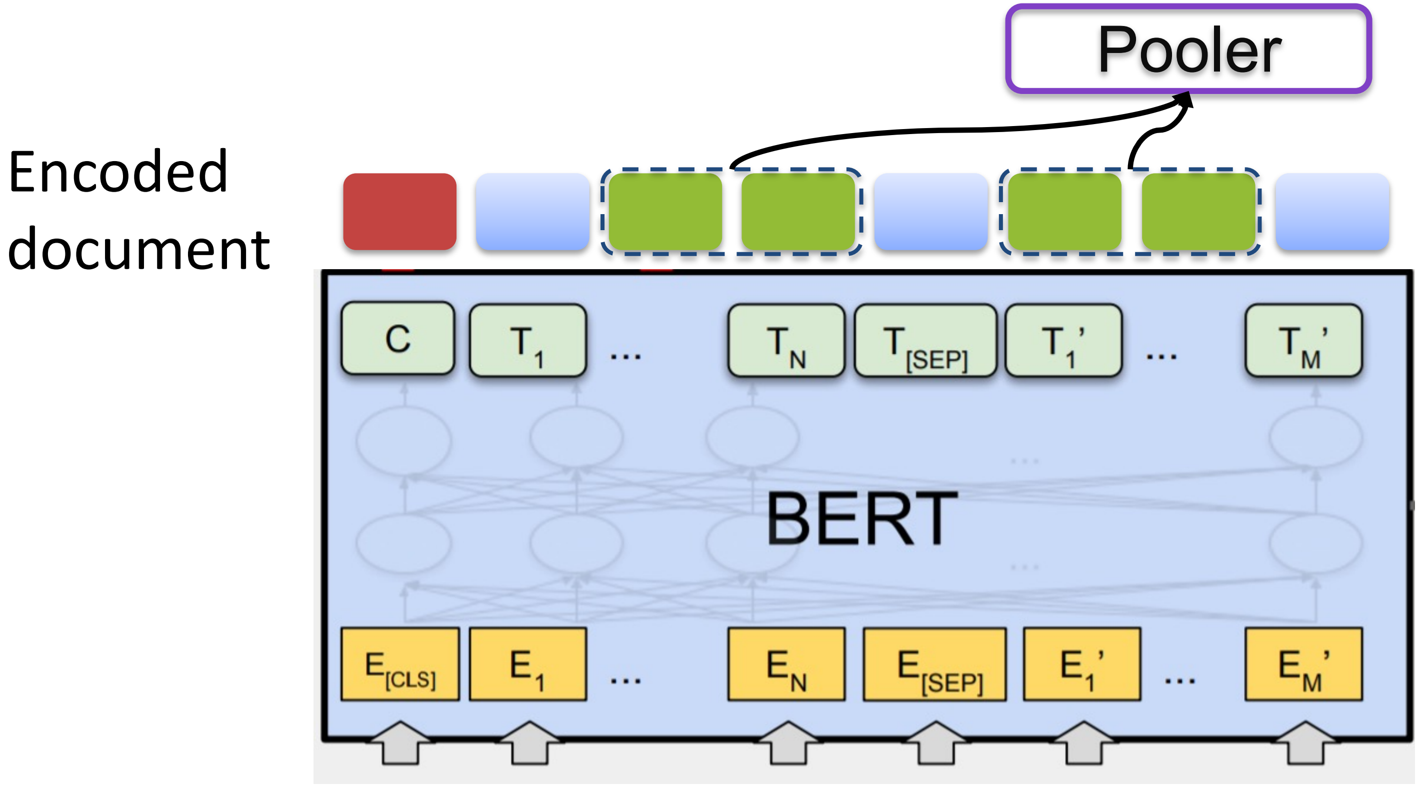


Figure from BERT (Devlin et al. 2018)

Existing Methods

- Graph-based Approaches
  - e.g. EoG (Christopoulou et al. 2019), GLRE (Wang et al., 2020), GAIN (Zeng et al., 2020)

Graphs connecting sentences, mentions and entities

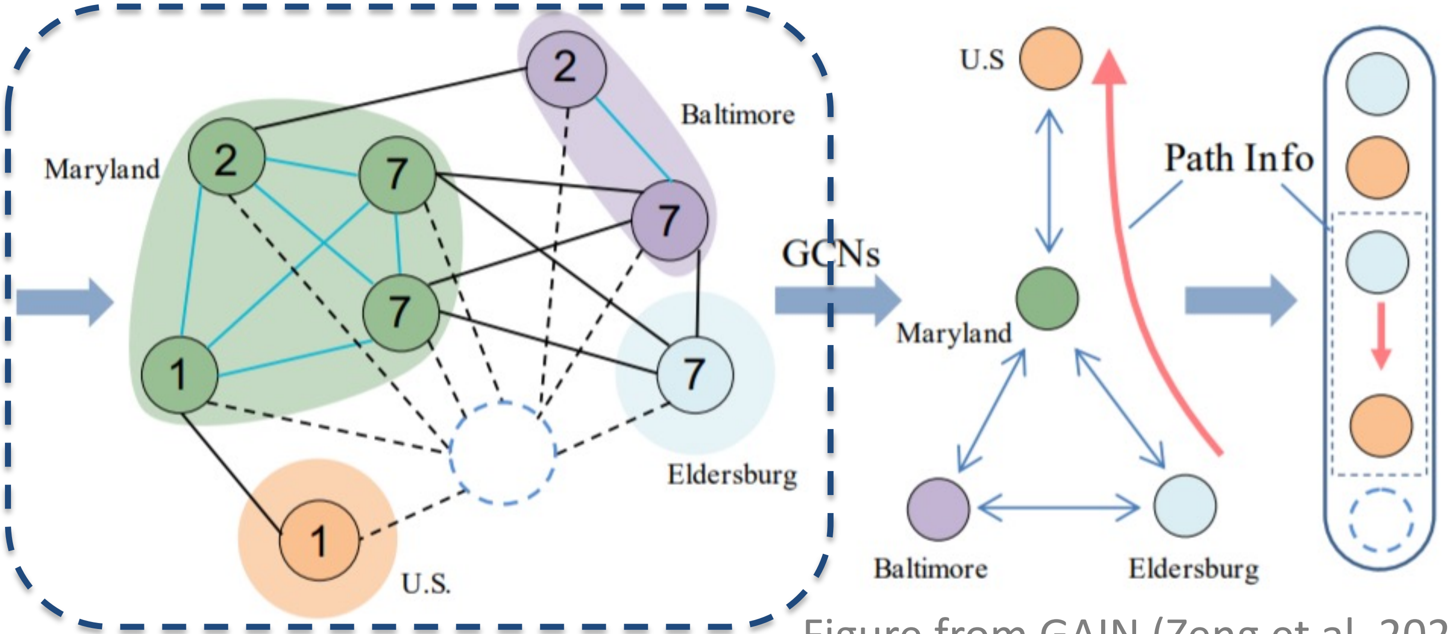


Figure from GAIN (Zeng et al. 2020)

Weaknesses of Prior Works

- Implicit long-range dependencies
  - PLMs encoding (suffers the difficulty on capturing long-range semantics)
  - Graph construction (depends on hand-crafted rules and contains only coarse granularity low-level information.)

- Implicit interactions modeling
  - features for interactions modeling, ignoring the explicit logical constraints among relations

- Poor Interpretability

Weaknesses of Prior Works

- Implicit long-range dependencies
  - PLMs encoding (suffers the difficulty on capturing long-range semantics)
  - Graph construction (depends on hand-crafted rules and contains only coarse granularity low-level information.)
- Implicit interactions modeling
  - features for interactions modeling, ignoring the explicit logical constraints among relations
- Poor Interpretability

Weaknesses of Prior Works

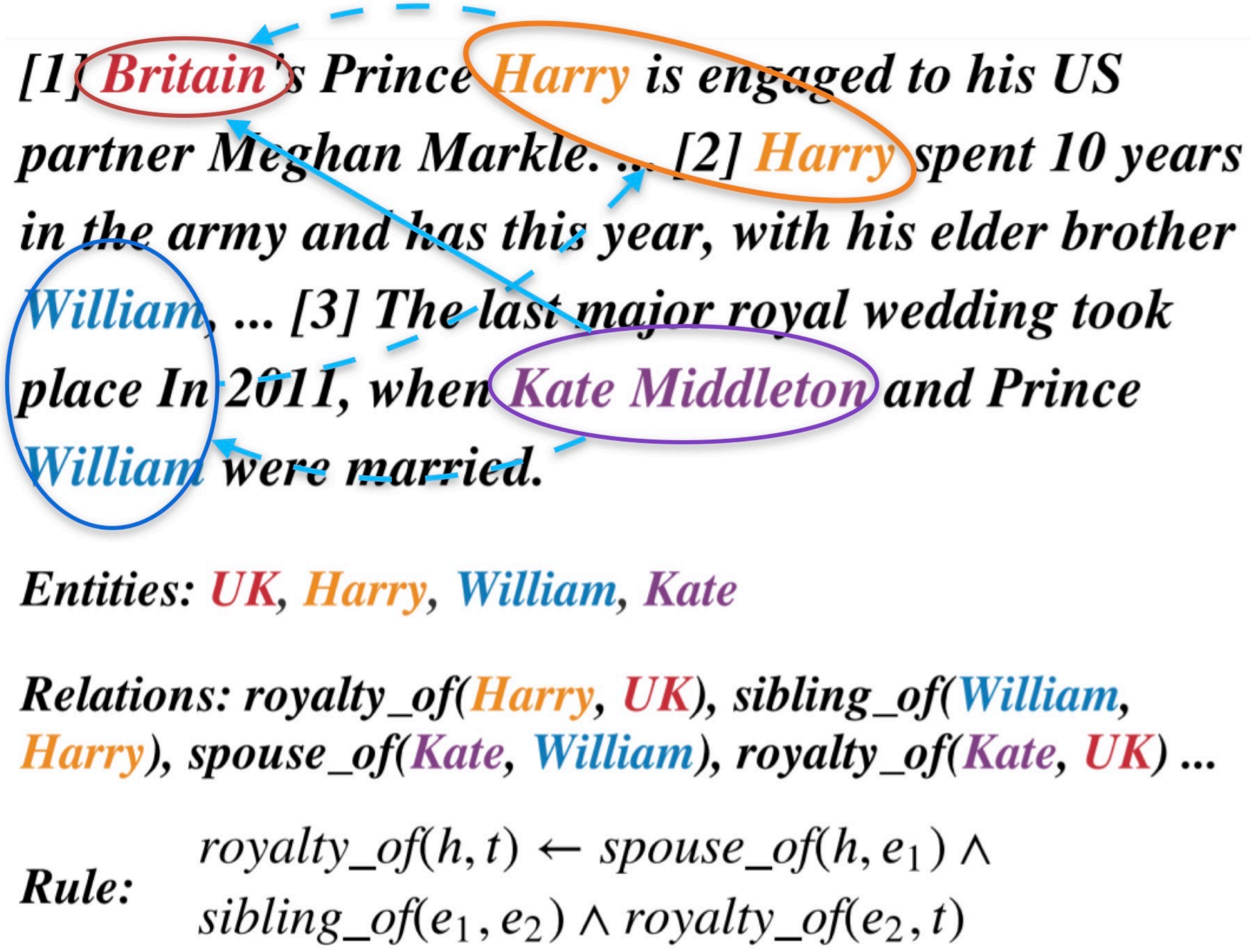
- Implicit long-range dependencies
  - PLMs encoding (suffers the difficulty on capturing long-range semantics)
  - Graph construction (depends on hand-crafted rules and contains only coarse granularity low-level information.)
- Implicit interactions modeling
  - features for interactions modeling, ignoring the explicit logical constraints among relations
- Poor Interpretability

LogiRE

- Strength of logic rules
  - Explicitly capturing long-range dependencies
  - Exhibiting interpretability
- Combining logic rules with neural network

LogiRE

- Strength of logic rules
  - Explicitly capturing long-range dependencies
  - Exhibiting interpretability
- Combining logic rules with neural network



LogiRE

- Strength of logic rules
  - Explicitly capturing long-range dependencies
  - Exhibiting interpretability
- Combining logic rules with neural network

The solid arc relation can be deducted from the other three dashed arc relations in concept level.

Logic Rules

Logic Rule

$$\forall \{e_i\}_{i=0}^{l}\, r(e_0, e_l) \leftarrow r_1(e_0, e_1) \wedge \cdots \wedge r_l(e_{l-1}, e_l)$$

$e_i \in \mathcal{E}$ entity set

$r_i \in \mathcal{T}_r$ predefined relations

Logic Rules

Logic Rule

$$\forall \{e_i\}_{i=0}^{l}\, r(e_0, e_l) \leftarrow r_1(e_0, e_1) \wedge \cdots \wedge r_l(e_{l-1}, e_l)$$

$e_i \in \mathcal{E}$ entity set

$r_i \in \mathcal{T}_r$ predefined relations

e.g. transferability: $\forall \{e_0, e_1, e_2\}\, r(e_0, e_2) \leftarrow r(e_0, e_1) \wedge r(e_1, e_2)$

Logic Rules

Logic Rule

$$\forall \{e_i\}_{i=0}^{l}\, r(e_0, e_l) \leftarrow r_1(e_0, e_1) \wedge \cdots \wedge r_l(e_{l-1}, e_l)$$

$e_i \in \mathcal{E}$ entity set $r_i \in \mathcal{T}_r$ predefined relations

e.g. transferability: $\forall \{e_0, e_1, e_2\}\, r(e_0, e_2) \leftarrow r(e_0, e_1) \wedge r(e_1, e_2)$

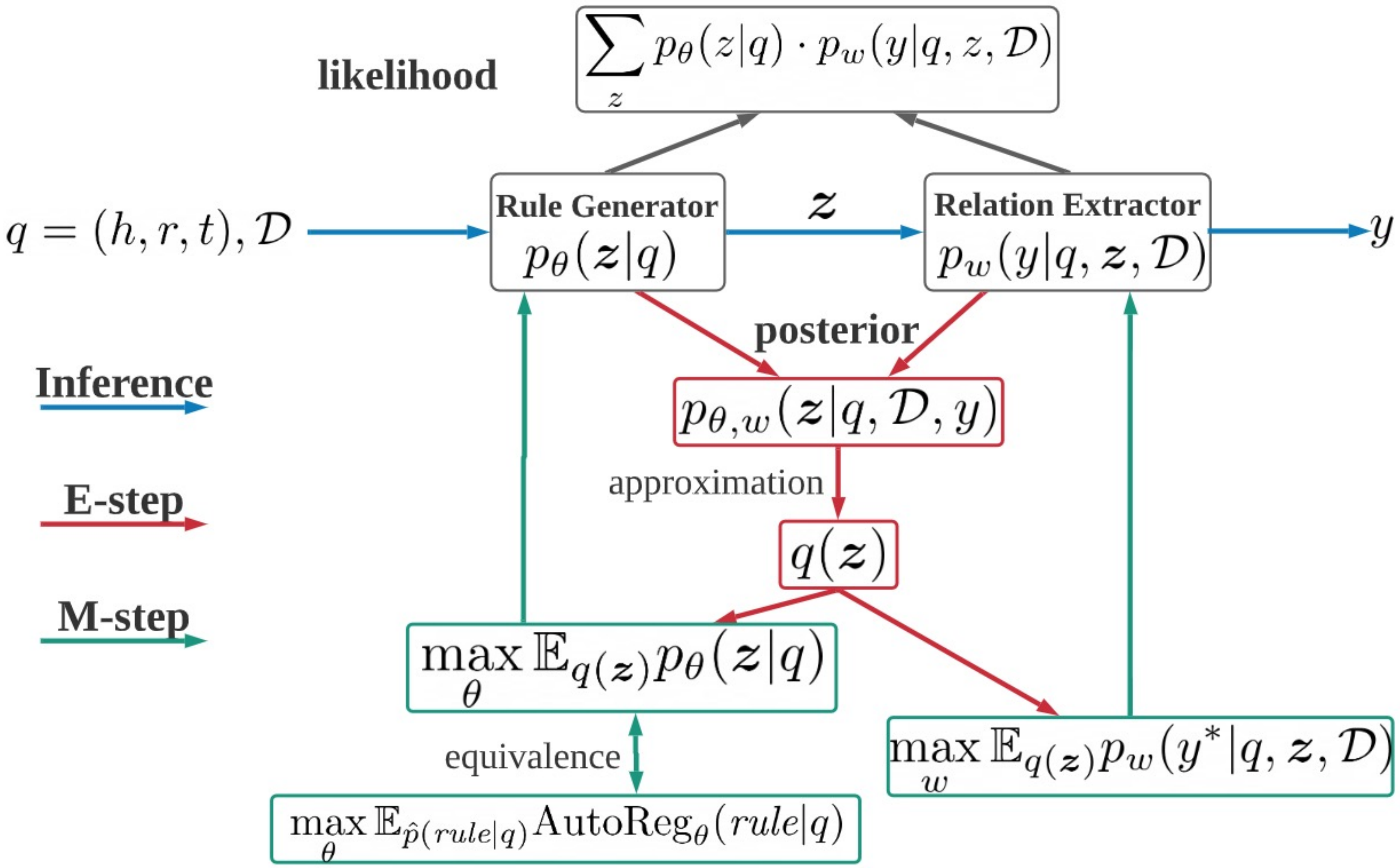
Instantiated Path

$$\begin{tikzcd} \underset{(h)}{e_0} \arrow[r, "r_1"] \arrow[rrrr, bend right, "r"'] & e_1 \arrow[r, "r_2"] & e_2 \arrow[r] & \cdots \arrow[r, "r_l"] & \underset{(t)}{e_l} \end{tikzcd}$$

Note: the direct relation r from h to t can be depicted by a path with intermediate entities and relations included.

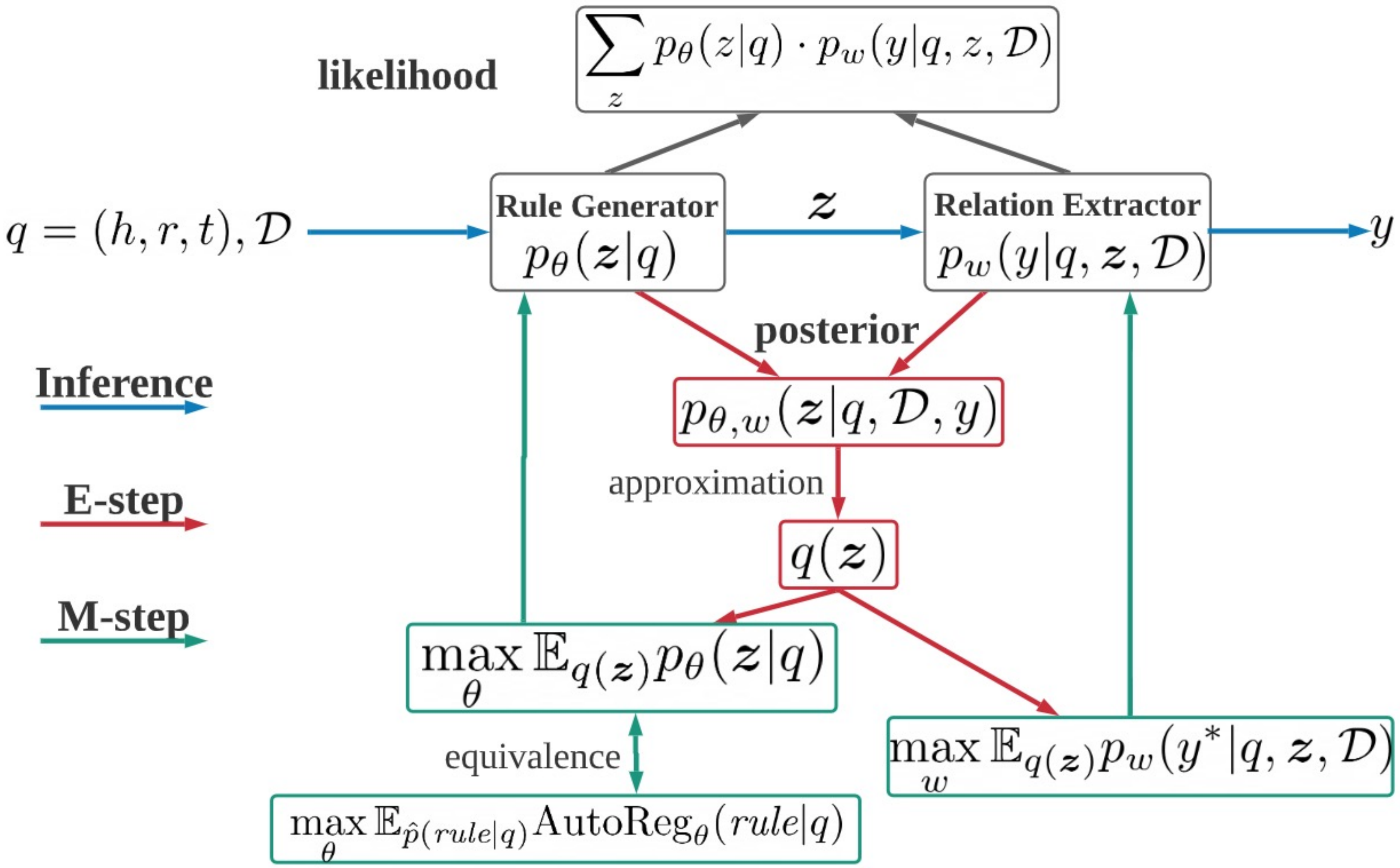
Framework Overview

- Treat logic rules as latent variable z
- Rule Generator
  - generates rules for the relation extractor
- Relation Extractor
  - predicts relations give queries and logic rules
  - provides supervision signals for the rule generator to produce high-quality rules
- EM Optimization



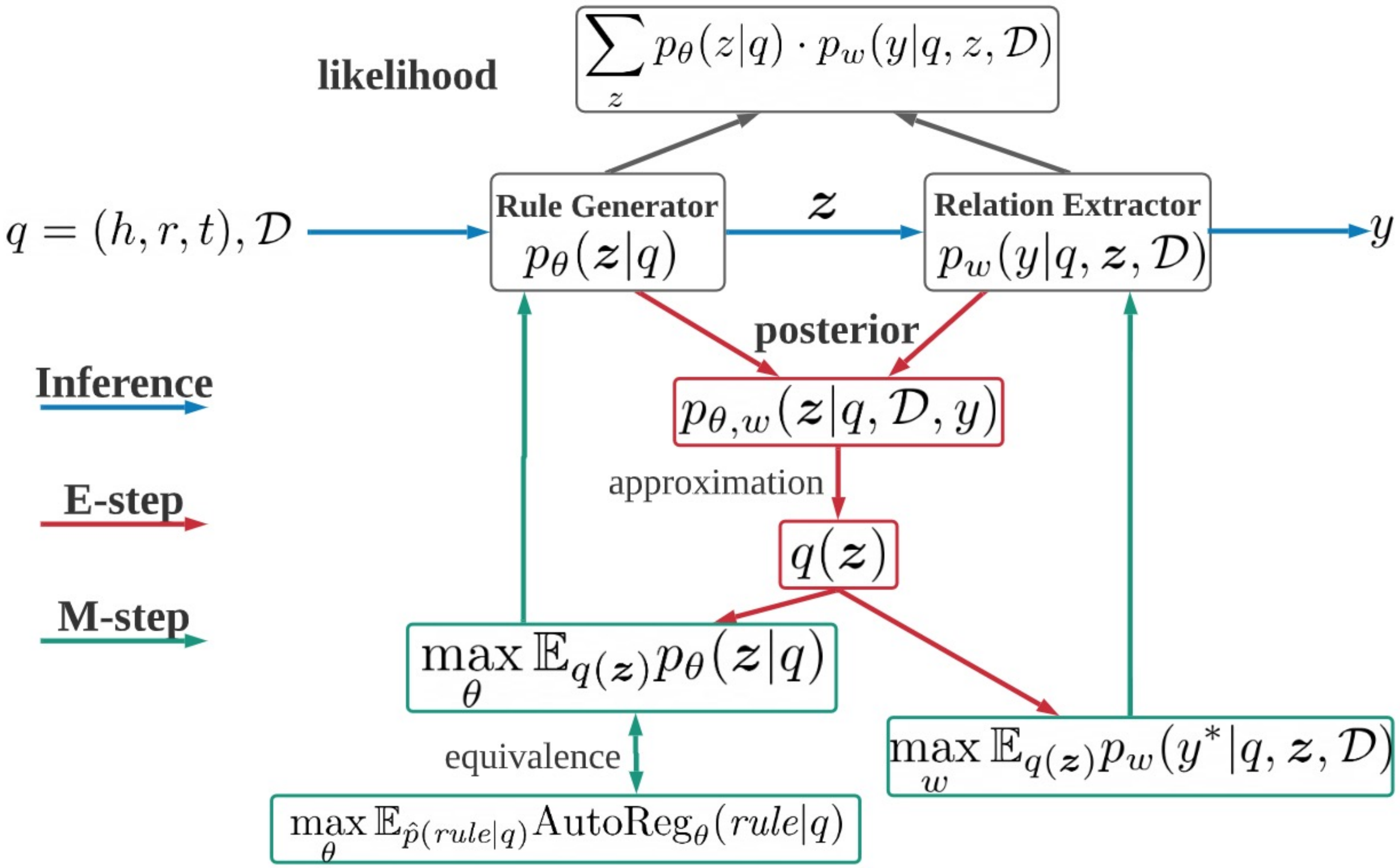
Framework Overview

- Treat logic rules as latent variable z
- Rule Generator
  - generates rules for the relation extractor
- Relation Extractor
  - predicts relations give queries and logic rules
  - provides supervision signals for the rule generator to produce high-quality rules
- EM Optimization



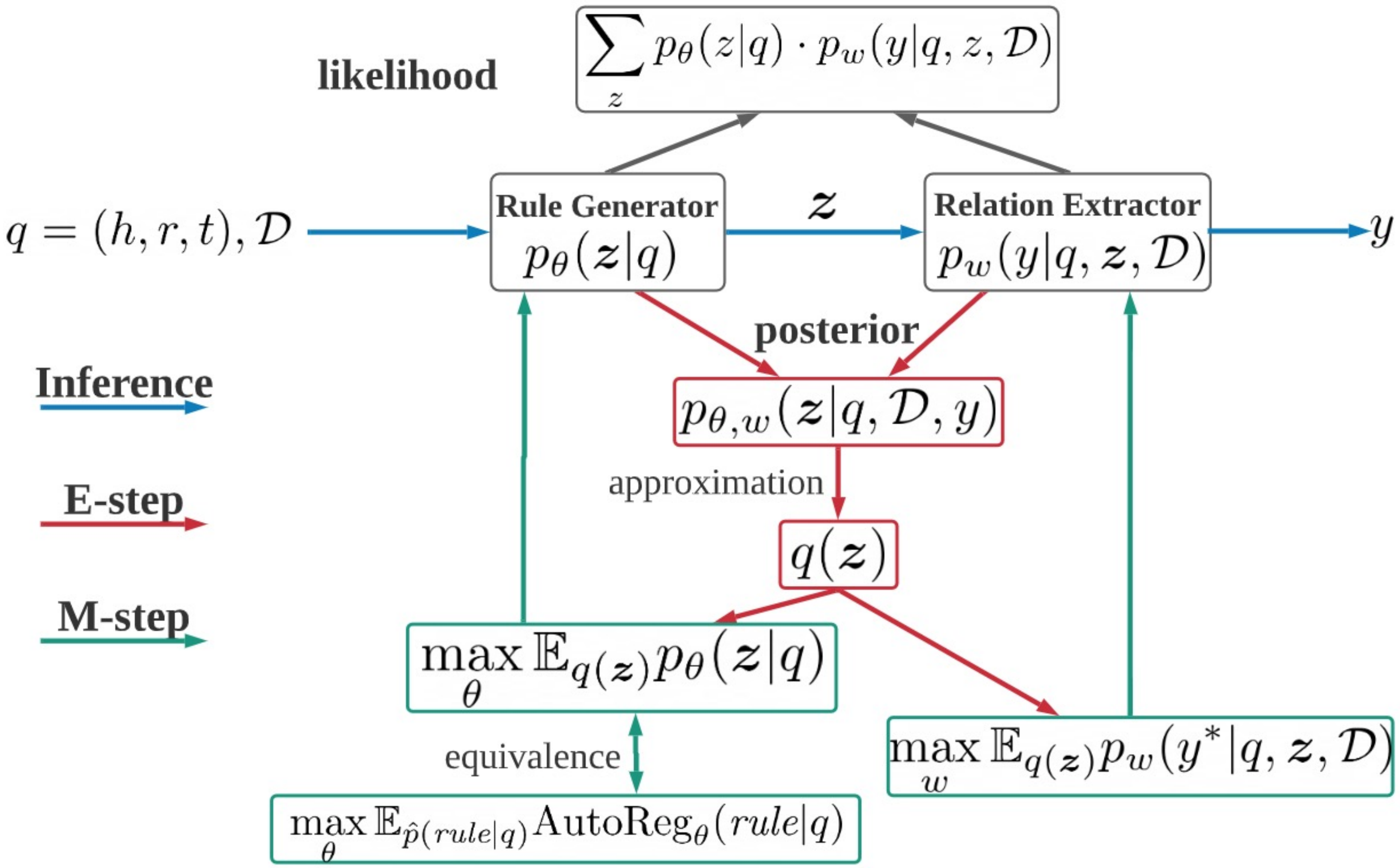
Framework Overview

- Treat logic rules as latent variable z
- Rule Generator
  - generates rules for the relation extractor
- Relation Extractor
  - predicts relations give queries and logic rules
  - provides supervision signals for the rule generator to produce high-quality rules
- EM Optimization



Framework Overview

- Treat logic rules as latent variable z
- Rule Generator
  - generates rules for the relation extractor
- Relation Extractor
  - predicts relations give queries and logic rules
  - provides supervision signals for the rule generator to produce high-quality rules
- EM Optimization



Rule Generator

- Following RNNLogic (Qu et al. 2021), the latent variable z is defined as a multi-set containing multiple rules

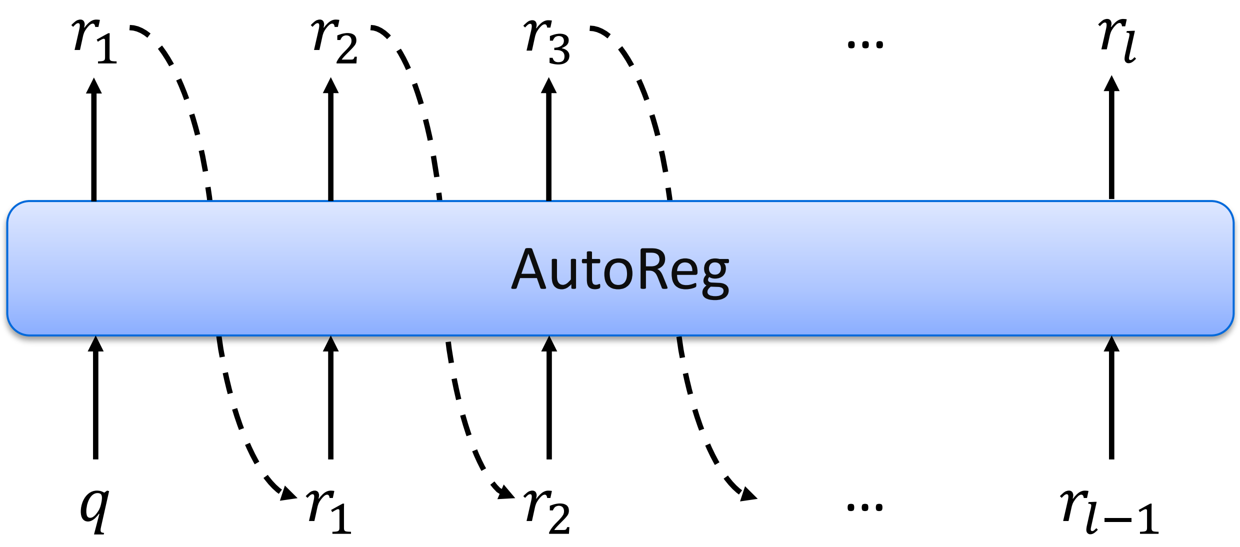
$$p_\theta(\boldsymbol{z}|q) \sim \mathrm{Multi}(\boldsymbol{z}|N, \mathrm{AutoReg}_\theta(rule|q))$$

Rule Generator

- Following RNNLogic (Qu et al. 2021), the latent variable z is defined as a multi-set containing multiple rules

$$p_\theta(z|q) \sim \text{Multi}(z|N, \text{AutoReg}_\theta(rule|q))$$

$$\text{AutoReg}_\theta(rule|q) = \prod_{i=1}^{l} \text{AutoReg}_\theta(r_i|q, r_1, r_2, \ldots, r_{i-1})$$



Relation Extractor

- Backbone model: initial probabilistic assessment on a triple
- Rule scorer: logic fusion over all instantiated paths of rules

$$\phi_w(rule) = \max_{path \in \mathcal{P}(rule)} \phi_w(path)$$

$$path:\ \underset{(h)}{e_0} \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \rightarrow \cdots \xrightarrow{r_l} \underset{(t)}{e_l}$$

$$\phi_w(path) = \prod_{i=1}^{l} \phi_w(e_{i-1}, r_i, e_i)$$

$$p_w(\boldsymbol{y}|q, \boldsymbol{z}) = \text{Sigmoid}(\boldsymbol{y} \cdot \texttt{score}_w(q, \boldsymbol{z}))$$
$$\texttt{score}_w(q, \boldsymbol{z}) = \phi_w(q) + \sum_{rule \in \boldsymbol{z}} \phi_w(q, rule)\phi_w(rule)$$

Relation Extractor

- Backbone model: initial probabilistic assessment on a triple
- Rule scorer: logic fusion over all instantiated paths of rules

dynamic programming

$$\phi_w(rule) = \max_{path \in \mathcal{P}(rule)} \phi_w(path)$$

$$path\colon \underset{(h)}{e_0} \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \rightarrow \cdots \xrightarrow{r_l} \underset{(t)}{e_l}$$

$$\phi_w(path) = \prod_{i=1}^{l} \phi_w(e_{i-1}, r_i, e_i)$$

$$p_w(\boldsymbol{y}|q, \boldsymbol{z}) = \text{Sigmoid}(\boldsymbol{y} \cdot \texttt{score}_w(q, \boldsymbol{z}))$$

$$\texttt{score}_w(q, \boldsymbol{z}) = \phi_w(q) + \sum_{rule \in \boldsymbol{z}} \phi_w(q, rule)\phi_w(rule)$$

Relation Extractor

- Backbone model: initial probabilistic assessment on a triple
- Rule scorer: logic fusion over all instantiated paths of rules

dynamic programming

$$\phi_w(rule) = \max_{path \in \mathcal{P}(rule)} \phi_w(path)$$

$$path:\ \underset{(h)}{e_0} \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \rightarrow \cdots \xrightarrow{r_l} \underset{(t)}{e_l}$$

$$\phi_w(path) = \prod_{i=1}^{l} \phi_w(e_{i-1}, r_i, e_i)$$

probability assessment on the triple (e_{i-1}, r_i, e_i) by the backbone model

$$p_w(\boldsymbol{y}|q, z) = \mathrm{Sigmoid}(\boldsymbol{y} \cdot \mathtt{score}_w(q, z))$$

$$\mathtt{score}_w(q, z) = \phi_w(q) + \sum_{rule \in z} \phi_w(q, rule)\phi_w(rule)$$

Relation Extractor

- Backbone model: initial probabilistic assessment on a triple
- Rule scorer: logic fusion over all instantiated paths of rules

dynamic programming

$$\phi_w(rule) = \max_{path \in \mathcal{P}(rule)} \phi_w(path)$$

$$path:\ \underset{(h)}{e_0} \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \rightarrow \cdots \xrightarrow{r_l} \underset{(t)}{e_l}$$

$$\phi_w(path) = \prod_{i=1}^{l} \phi_w(e_{i-1}, r_i, e_i)$$

probability assessment on the triple (e_{i-1}, r_i, e_i) by the backbone model

$$p_w(\boldsymbol{y}|q, \boldsymbol{z}) = \text{Sigmoid}(\boldsymbol{y} \cdot \texttt{score}_w(q, \boldsymbol{z}))$$

$$\texttt{score}_w(q, \boldsymbol{z}) = \phi_w(q) + \sum_{rule \in \boldsymbol{z}} \phi_w(q, rule)\phi_w(rule)$$

fuzzy logic for disjunction over all conjunctive rules

Optimization (EM)

- E-step: exact posterior inference → approximated posterior
- M-step: maximize lower bound

$$\overbrace{\mathop{\mathbb{E}}_{q(\boldsymbol{z})}\left[\log \frac{p_{w,\theta}(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{q})}{p_{w,\theta}(\boldsymbol{z}|\boldsymbol{q},\boldsymbol{y})}\right]}^{\log p_{w,\theta}(\boldsymbol{y}|\boldsymbol{q})} - \overbrace{\mathop{\mathbb{E}}_{q(\boldsymbol{z})}\left[\log \frac{p_{w,\theta}(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{q})}{q(\boldsymbol{z})}\right]}^{\text{lower bound}}$$

$$= \mathrm{KL}\left(q(\boldsymbol{z})||p_{w,\theta}(\boldsymbol{z}|\boldsymbol{q},\boldsymbol{y})\right) \geq 0$$

$$\mathcal{L}_{\text{lower}} = \overbrace{\mathop{\mathbb{E}}_{q(\boldsymbol{z})}[\log p_\theta(\boldsymbol{z}|q)]}^{\mathcal{L}_{\mathrm{G}}} + \overbrace{\mathop{\mathbb{E}}_{q(\boldsymbol{z})}[\log p_{w,\theta}(y^*|\boldsymbol{z},q)]}^{\mathcal{L}_{\mathrm{R}}}$$

$q(\boldsymbol{z})$ is the approximated posterior of the latent rule set

(details)

Posterior Approximation

- Prior distribution: $p_\theta(z|q) \sim \text{Multi}(z|N, \text{AutoReg}_\theta(rule|q))$

- Posterior

$$
\begin{aligned}
&\log p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{q}, \mathcal{D}) \\
&= \log p_w(\boldsymbol{y}|\boldsymbol{q}, z, \mathcal{D}) + \log p_\theta(\boldsymbol{z}|\boldsymbol{q}) + C \\
&= \log \frac{1}{1 + e^{-\boldsymbol{y} \cdot \texttt{score}(\mathbf{q}, \boldsymbol{z})}} + \sum_{rule \in \boldsymbol{z}} \log \text{AutoReg}_\theta(rule|\boldsymbol{q}) + C \\
&\approx \frac{1}{2}\boldsymbol{y} \cdot \texttt{score}_w(\boldsymbol{q}, \boldsymbol{z}) + \sum_{rule \in \boldsymbol{z}} \log \text{AutoReg}_\theta(rule|\boldsymbol{q}) + C \\
&= \sum_{rule \in \boldsymbol{z}} (\frac{1}{2}\boldsymbol{y} \cdot (\frac{1}{N}(\phi_w(\boldsymbol{q}) + \phi_w(\boldsymbol{q}, rule)\phi_w(rule)) \\
&\quad + \log \text{AutoReg}_\theta(rule|\boldsymbol{q})) + C
\end{aligned}
$$

(details)

Posterior Approximation

- Prior distribution: $p_\theta(z|q) \sim \text{Multi}(z|N, \text{AutoReg}_\theta(rule|q))$

- Posterior

$$
\begin{aligned}
&\log p(z|\boldsymbol{y}, \boldsymbol{q}, \mathcal{D}) \\
&= \log p_w(\boldsymbol{y}|\boldsymbol{q}, z, \mathcal{D}) + \log p_\theta(z|\boldsymbol{q}) + C \\
&= \log \frac{1}{1 + e^{-\boldsymbol{y}\cdot \mathtt{score}(\mathbf{q},z)}} + \sum_{rule \in z} \log \text{AutoReg}_\theta(rule|\boldsymbol{q}) + C \\
&\boxed{\approx \frac{1}{2}\boldsymbol{y} \cdot \mathtt{score}_w(\boldsymbol{q}, z) + \sum_{rule \in z} \log \text{AutoReg}_\theta(rule|\boldsymbol{q}) + C} \\
&= \sum_{rule \in z} (\frac{1}{2}\boldsymbol{y} \cdot (\frac{1}{N}(\phi_w(\boldsymbol{q}) + \phi_w(\boldsymbol{q}, rule)\phi_w(rule)) \\
&\quad + \log \text{AutoReg}_\theta(rule|\boldsymbol{q})) + C
\end{aligned}
$$

order-2 Taylor expansion

$$\boxed{-\log(1 + e^{-x}) = -\log 2 + \frac{x}{2} + O(x^2)}$$

(details)

Posterior Approximation

- Prior distribution: $p_\theta(z|q) \sim \mathrm{Multi}(z|N, \mathrm{AutoReg}_\theta(rule|q))$

- Approximated Posterior

$$H(rule) = \log \mathrm{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

$$q(z) \sim \mathrm{Multi}(N, \frac{1}{Z}\exp(H(rule)))$$

conjugate distributions

(details)

Posterior Approximation

- Prior distribution: $p_\theta(z|q) \sim \mathrm{Multi}(z|N, \mathrm{AutoReg}_\theta(rule|q))$

- Approximated Posterior

$$H(rule) = \log \mathrm{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

$$q(z) \sim \mathrm{Multi}(N, \frac{1}{Z}\exp(H(rule)))$$

conjugate distributions

🤗 conjugate property → easier optimization

$$\max_\theta \mathbb{E}_{q(\boldsymbol{z})} p_\theta(\boldsymbol{z}|q) \longrightarrow \max_\theta \mathbb{E}_{\hat{p}(rule|q)} \mathrm{AutoReg}_\theta(rule|q)$$

(details)

Iterative Optimization

- E-step
  - Obtain the approximated posterior

Approximated posterior of the rule set

$$H(rule) = \log \mathrm{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

$$q(\boldsymbol{z}) \sim \mathrm{Multi}(N, \frac{1}{Z}\exp(H(rule)))$$

Iterative Optimization

- E-step
  - Obtain the approximated posterior

Approximated posterior of the rule set

$$H(rule) = \log \text{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

$$q(\boldsymbol{z}) \sim \text{Multi}(N, \frac{1}{Z}\exp(H(rule)))$$

Approximated posterior of each rule

$$\hat{p}(rule|q) \propto \exp\left(H(rule)\right)$$

Iterative Optimization

(details)

- E-step
  - Obtain the approximated posterior

Approximated posterior of the rule set

$$H(rule) = \log \mathrm{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

$$q(z) \sim \mathrm{Multi}(N, \frac{1}{Z}\exp(H(rule)))$$

Approximated posterior of each rule

$$\hat{p}(rule|q) \propto \exp\left(H(rule)\right)$$

- M-step
  - Optimize the parameters

Rule Generator

Relation Extractor

$$\mathcal{L}_{\text{lower}} = \underbrace{\mathop{\mathbb{E}}_{q(z)}[\log p_\theta(z|q)]}_{\mathcal{L}_{\text{G}}} + \underbrace{\mathop{\mathbb{E}}_{q(z)}[\log p_{w,\theta}(y^*|z, q)]}_{\mathcal{L}_{\text{R}}}$$

(details)

Iterative Optimization

- E-step
  - Obtain the approximated posterior

Approximated posterior of the rule set

$$H(rule) = \log \text{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

$$q(z) \sim \text{Multi}(N, \frac{1}{Z}\exp(H(rule)))$$

Approximated posterior of each rule

$$\hat{p}(rule|q) \propto \exp\left(H(rule)\right)$$

- M-step
  - Optimize the parameters

Rule Generator

Relation Extractor

$$\mathcal{L}_{\text{lower}} = \underbrace{\mathop{\mathbb{E}}_{q(z)}[\log p_\theta(z|q)]}_{\mathcal{L}_{\text{G}}} + \underbrace{\mathop{\mathbb{E}}_{q(z)}[\log p_{w,\theta}(y^*|z,q)]}_{\mathcal{L}_{\text{R}}}$$

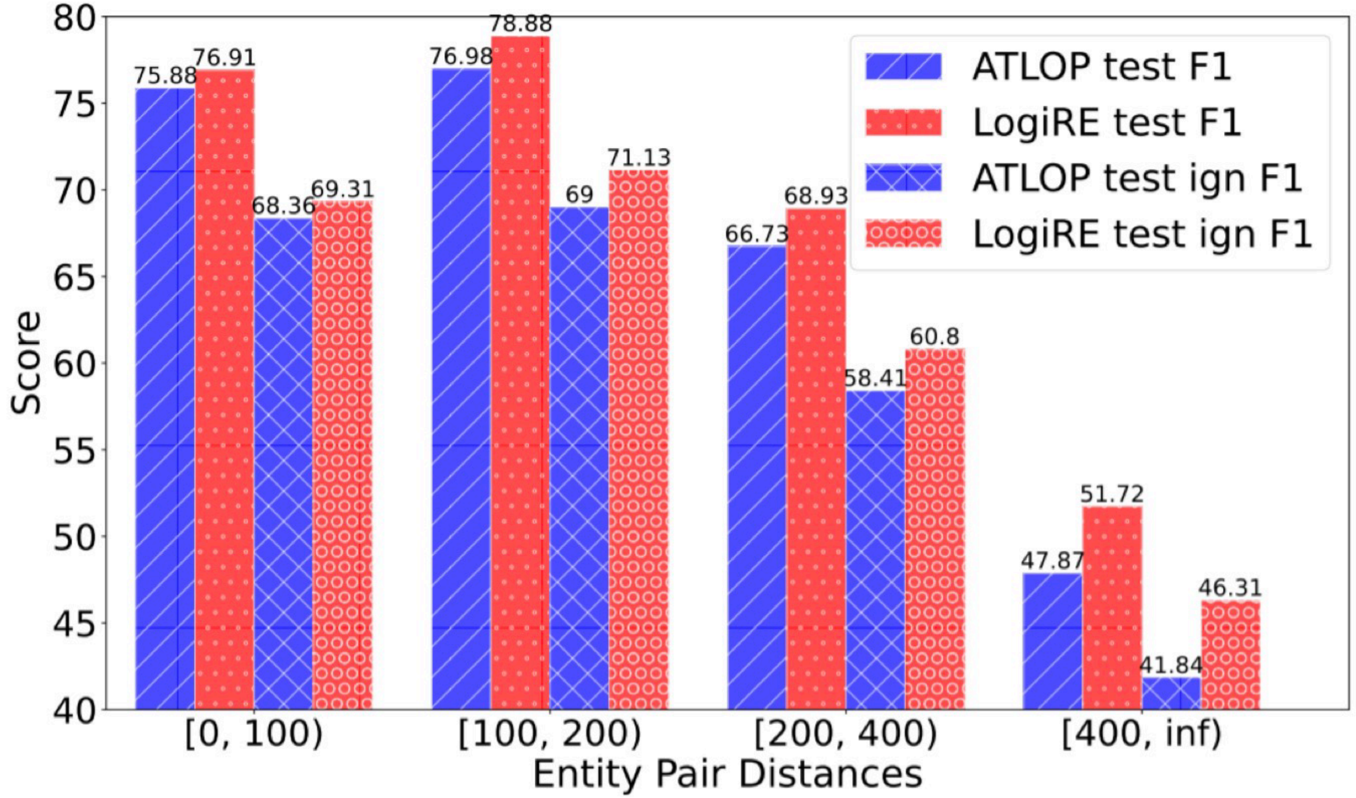
equivalence ensured by the conjugate property

$$\mathcal{L}_{\text{G}} = \mathbb{E}_{\hat{p}(rule|q)}[\text{AutoReg}_\theta(rule|q)]$$

High Performance & Logical Consistency

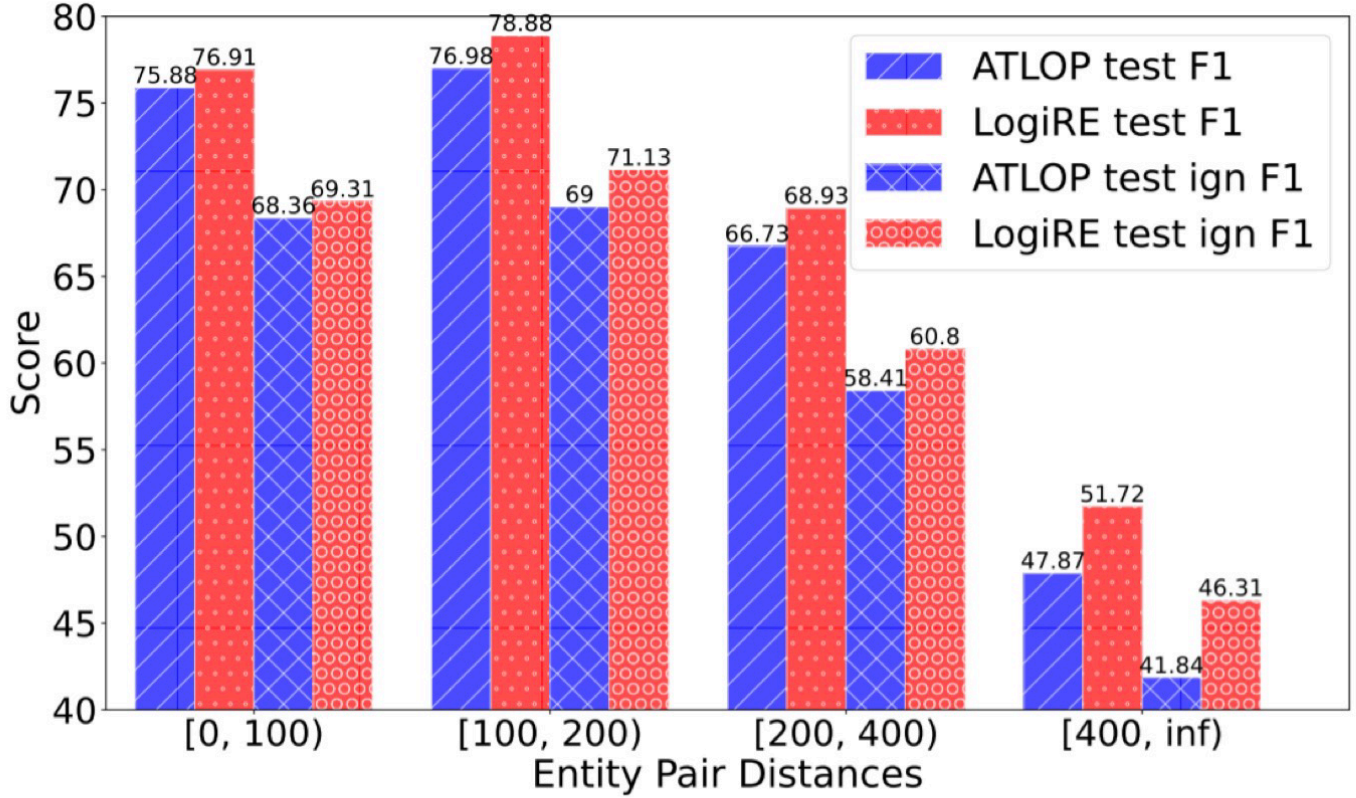
Model	Dev			Test		
	ign F1	F1	logic	ign F1	F1	logic
CNN	37.65	47.73	51.70	34.65	46.14	54.69
CNN + LogiRE	40.31(+2.65)	50.04(+2.71)	72.84(+21.14)	39.21(+4.65)	50.44(+4.30)	73.47(+18.78)
LSTM	40.86	51.77	65.64	40.81	52.60	61.64
LSTM + LogiRE	42.79(+1.93)	53.60(+1.83)	69.74(+4.10)	43.82(+3.01)	55.03(+2.43)	71.27(+9.63)
BiLSTM	40.46	51.92	64.87	42.03	54.47	64.41
BiLSTM + LogiRE	42.59(+2.13)	53.83(+1.91)	73.37(+8.50)	43.65(+1.62)	55.14(+0.67)	77.11(+12.70)
Context-Aware	42.06	53.05	69.27	45.37	56.58	70.01
Context-Aware + LogiRE	43.88(+1.82)	54.49(+1.44)	73.98(+4.71)	48.10(+2.73)	59.22(+2.64)	75.94(+5.93)
GAIN	58.63	62.55	78.30	62.37	67.57	86.19
GAIN + LogiRE	60.12 (+1.49)	63.91(+1.36)	**87.86** (+9.56)	**64.43** (+2.06)	69.40(+1.83)	**91.22**(+5.02)
ATLOP	59.03	64.82	81.98	62.09	69.94	82.76
ATLOP + LogiRE	**60.24**(+1.21)	**66.76**(+1.94)	86.98(+5.00)	64.11(+2.02)	**71.78**(+1.84)	86.07(+3.31)

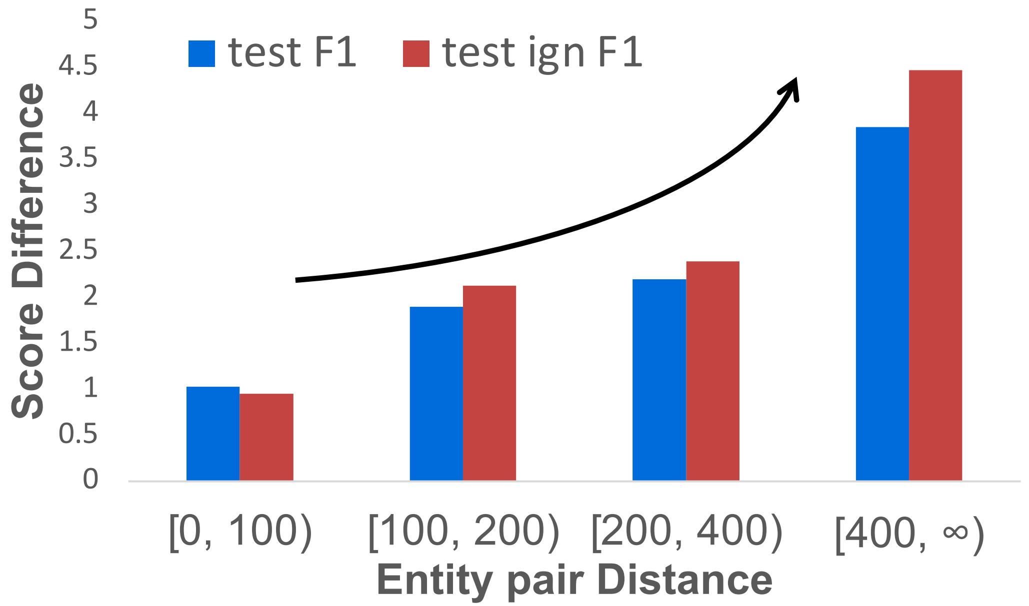
Good Long-range Dependencies Modeling



The absolute performance of both models decreases as the entity pair distances increase.

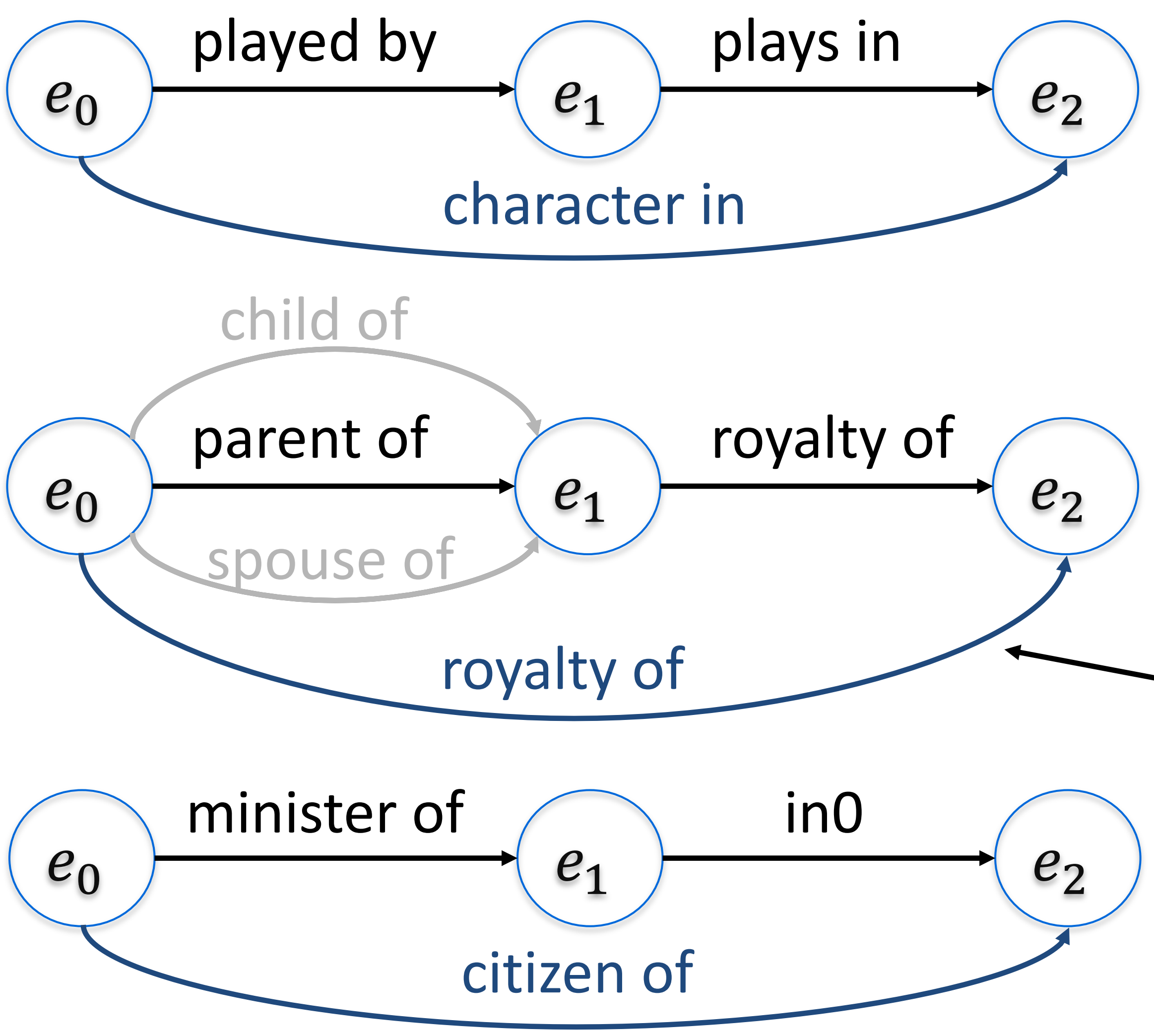
Good Long-range Dependencies Modeling





The absolute performance of both models decreases as the entity pair distances increase. The gap between them, however, increases.

More Interpretability & Transparency



Rule patterns extracted from the well trained rule generator.

Blue arcs are deducted from the intermediate entities and relations.

Gain Analysis

Model	Test	
	ign F1	**F1**
GAIN	57.93	60.07
GAIN + LogiRE	58.62(+0.69)	60.61(+0.54)
ATLOP	59.14	61.13
ATLOP + LogiRE	59.48(+0.34)	61.45(+0.32)

The improvements on DocRED are less significant than those on DWIE.

Gain Analysis

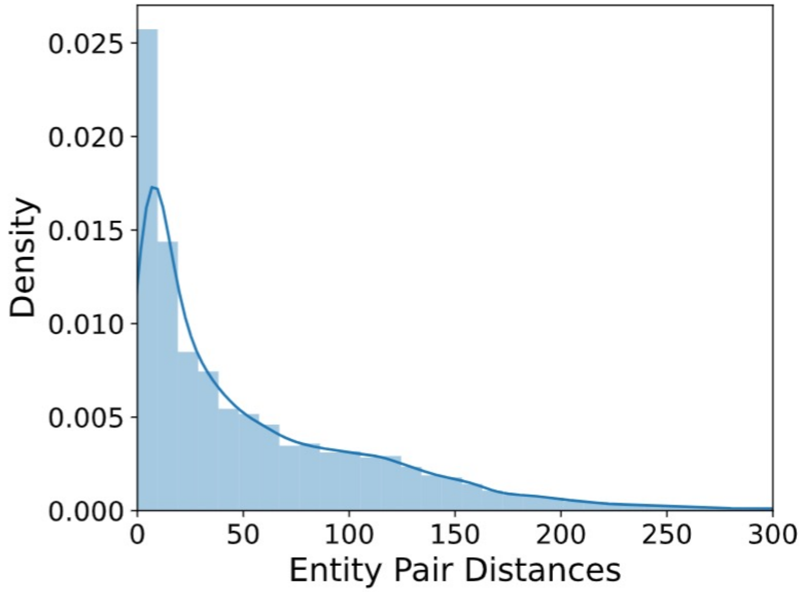
Model	Test	
	ign F1	**F1**
GAIN	57.93	60.07
GAIN + LogiRE	58.62(+0.69)	60.61(+0.54)
ATLOP	59.14	61.13
ATLOP + LogiRE	59.48(+0.34)	61.45(+0.32)

The improvements on DocRED are less significant than those on DWIE.

Logical inconsistency in annotations

implication rule	error rate
$\text{father}(h,z) \wedge \text{spouse}(z,t) \rightarrow \text{mother}(h,t)$	24.07%
$\text{replaces}^{-1}(h,t) \rightarrow \text{replaced\_by}(h,t)$	22.22%
$\text{capital}^{-1}(h,t) \rightarrow \text{capital\_of}(h,t)$	28.24%
$\text{father}^{-1}(h,t) \rightarrow \text{child}(h,t)$	10.26%
$\text{followed}^{-1}(h,t) \rightarrow \text{follows}(h,t)$	22.40%
$\text{capital}^{-1}(h,t) \rightarrow \text{capital\_of}(h,t)$	28.24%
$\text{P150}^{-1}(h,t) \rightarrow \text{P131}(h,t)$	19.71%

shorter dependencies



Summary

- Conclusion
  - Logic rules + NN → doc-level RE
  - Iterative EM optimization for latent rules
  - Better long-range dependencies modeling and logical consistency

Summary

- Conclusion
  - Logic rules + NN → doc-level RE
  - Iterative EM optimization for latent rules
  - Better long-range dependencies modeling and logical consistency

- Future directions for doc-level RE
  - Learning with noisy data
  - Global decoding for doc-level RE
  - Zero-shot / few-shot meta relation induction and learning

Thanks!

Repo Link