IJCAI'22

ACL-IJCAI-SIGIR
顶级会议论文报告会
(AIS 2022)

Efficient Document-level Event Extraction via Pseudo-Trigger-aware Pruned Complete Graph

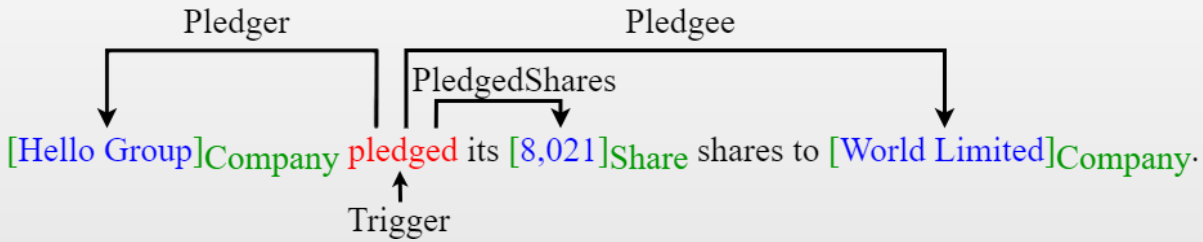
Tong Zhu[1], Xiaoye Qu[2], Wenliang Chen[1], Zhefeng Wang[2],
Baoxing Huai[2], Nicholas Yuan[2], Min Zhang[1]

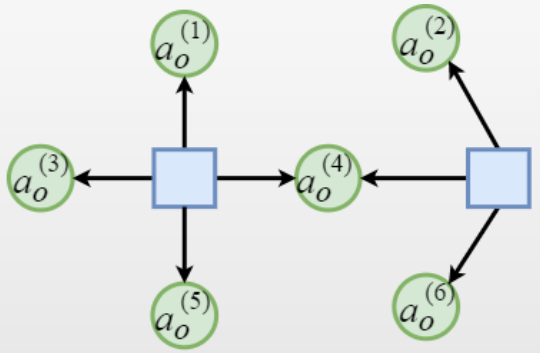
https://github.com/Spico197/DocEE

[1]Soochow University

[2]Huawei Cloud

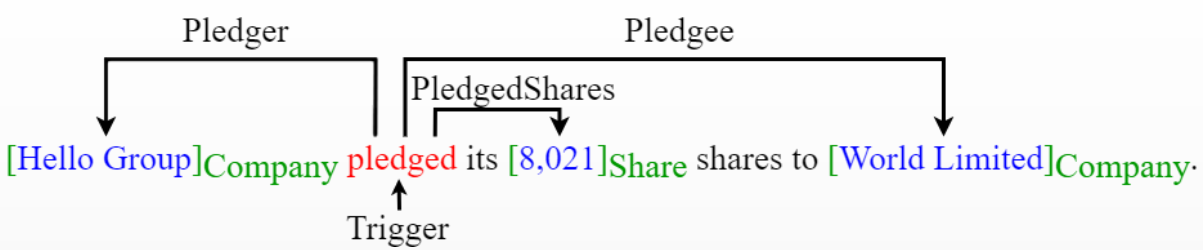
引言：句级事件抽取

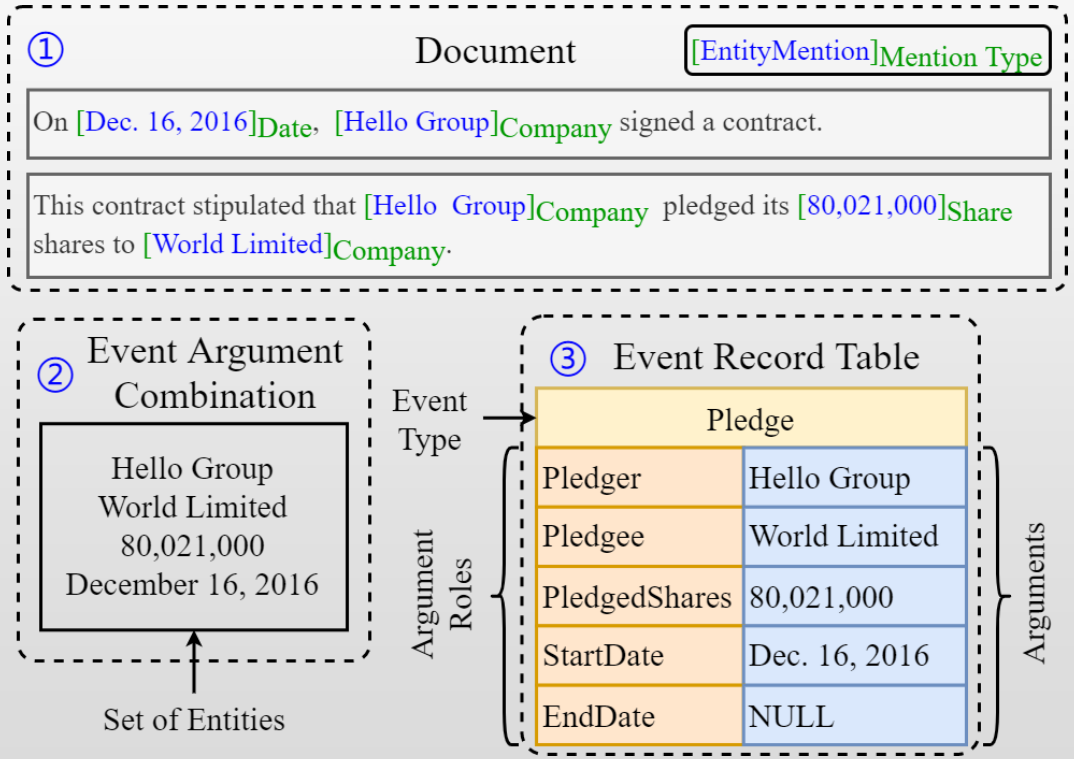


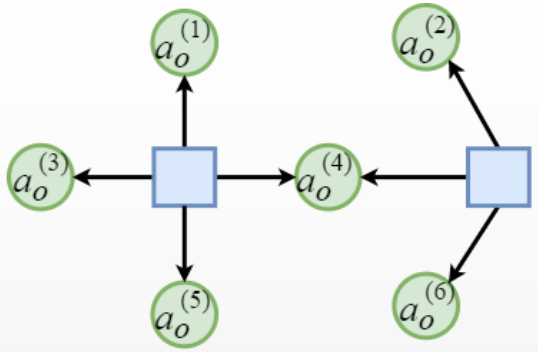


Annotated-Trigger-centered Trees

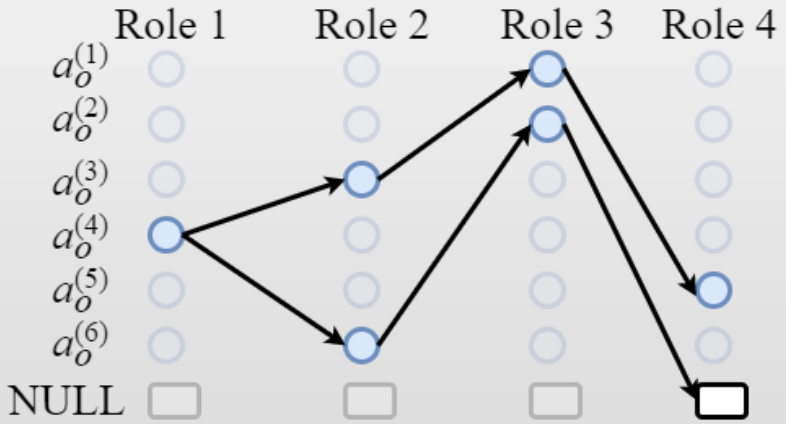
引言：篇章事件抽取





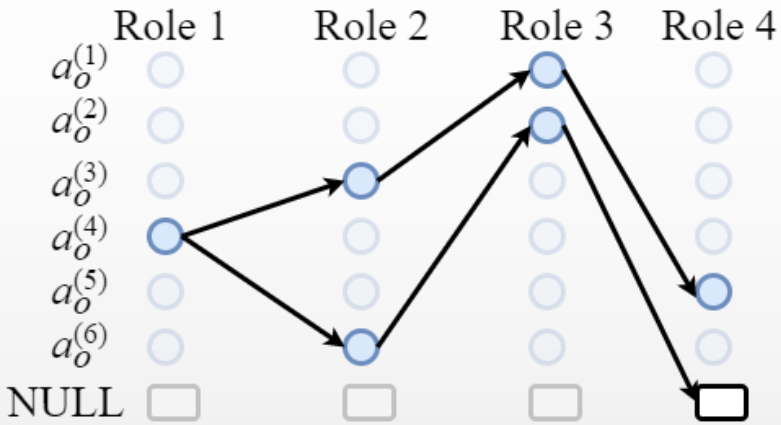


Annotated-Trigger-centered Trees



Directed Acyclic Graph
w/o Annotated Triggers

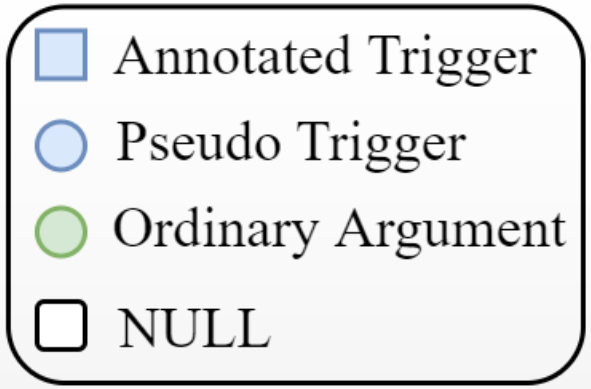
引言：任务难点和痛点

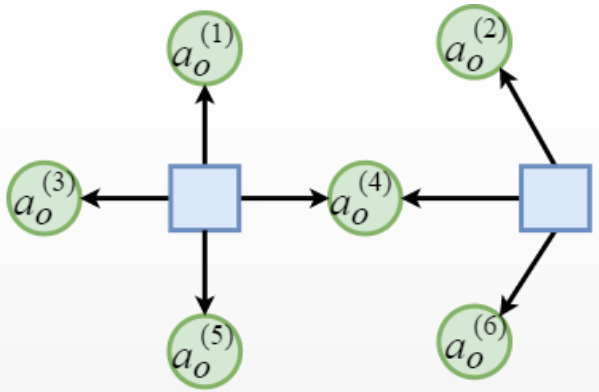


Directed Acyclic Graph w/o Annotated Triggers

- 任务本身的难点：
  - 无触发词
  - 长文本处理
- 当前方法的痛点：
  - 训练慢：4-8卡跑将近一周
  - 推理时消耗资源多

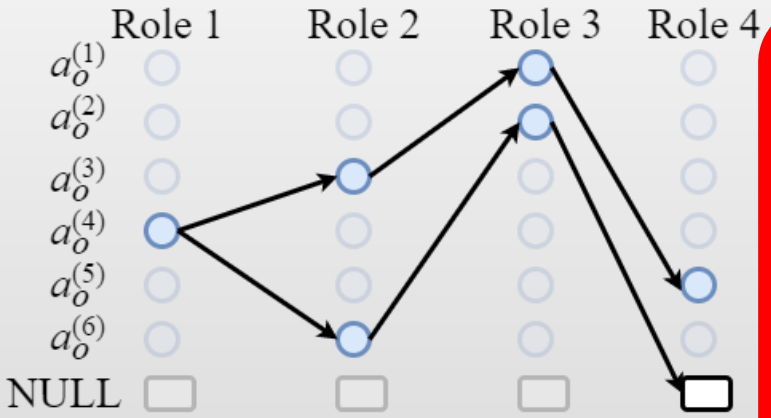
解决思路：完全图+剪枝



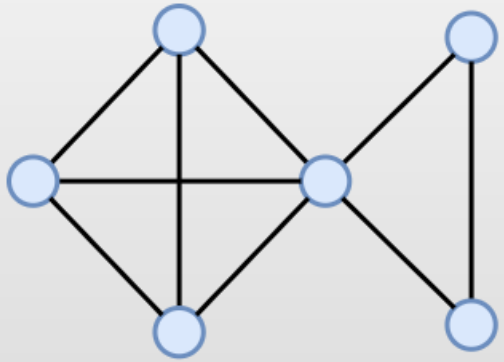


Annotated-Trigger-centered Trees

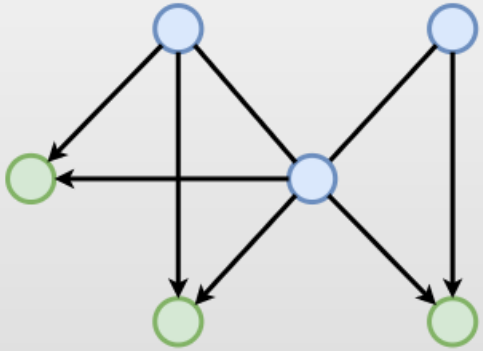
如果每个事件实例只选择一个核心论元，则退化为句级事件抽取中的常用组合方法



Directed Acyclic Graph
w/o Annotated Triggers



Complete Graph
w/o Annotated Triggers



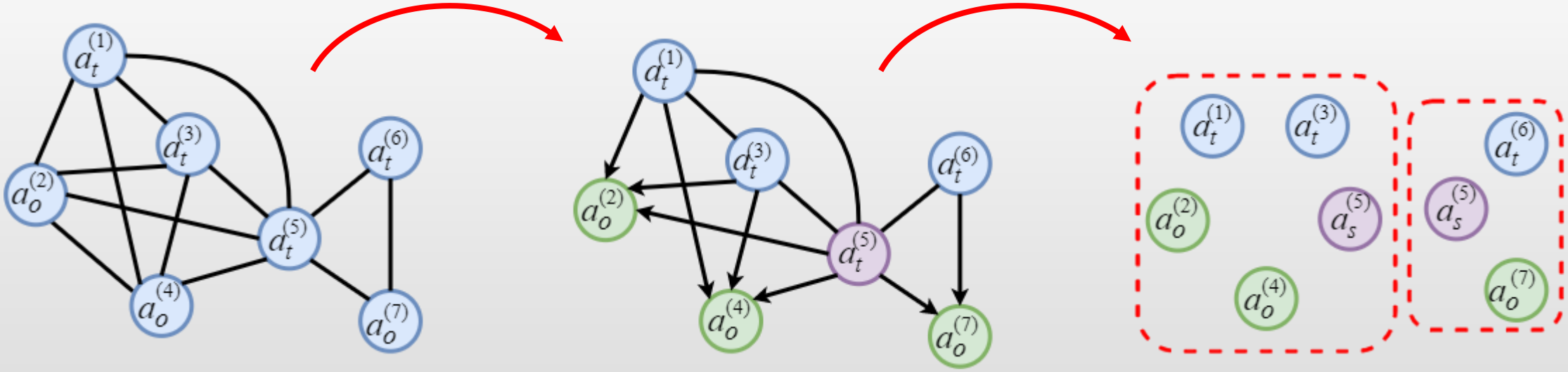
Pruned Complete Graph
w/o Annotated Triggers

通过选择重要的论元，可以进行剪枝操作

当前需要解决的问题

如何选择最“重要”的论元，
从而构建为剪枝完全图?

如何根据构建的图解码出一
个事件组合?



伪触发词的选择

触发词在事件抽取中究竟承担什么角色？或者有什么特征？

- 存在性（Existence）：触发词在实例中必须存在，从而指示（identify）事件实例
- 区分性（Distinguishability）：触发词不被共享，可以区分不同的事件实例

可以使用 一组论元角色R 对应的 论元 作为 事件实例的 “伪触发词”

$$\text{一组论元角色}R\text{的存在性} = \frac{R\text{对应的论元中，至少有一个论元在实例中存在的数量}}{\text{整个事件实例的数量}}$$

$$\text{一组论元角色}R\text{的区分性} = \frac{R\text{对应的论元中，至少有一个论元在实例中存在的数量}}{\text{整个事件实例的数量}}$$

$$\text{重要性} = \text{存在性} \times \text{区分性}$$

伪触发词的选择

Document $\mathcal{D}_1$

	Plankton		Plankton		Squidward
	Krabs		Sandy		Pearl
	NULL		NULL		3,456,000
	Dec. 16, 2016		NULL		Nov. 16, 2016

Document $\mathcal{D}_2$

	Patrick		Gary		NULL
	NULL		Patrick		SpongeBob
	6,800		NULL		6,800
	Jan. 16, 1993		NULL		Dec. 16, 2016

伪触发词的选择

Document $\mathcal{D}_1$

	Plankton		Plankton		Squidward
	Krabs		Sandy		Pearl
	NULL		NULL		3,456,000
	Dec. 16, 2016		NULL		Nov. 16, 2016

Document $\mathcal{D}_2$

	Patrick		Gary		NULL
	NULL		Patrick		SpongeBob
	6,800		NULL		6,800
	Jan. 16, 1993		NULL		Dec. 16, 2016

对每一组可能的角色都进行计算

For $|\mathcal{R}| = 2$ and $|$ $|$ $= 6$ records

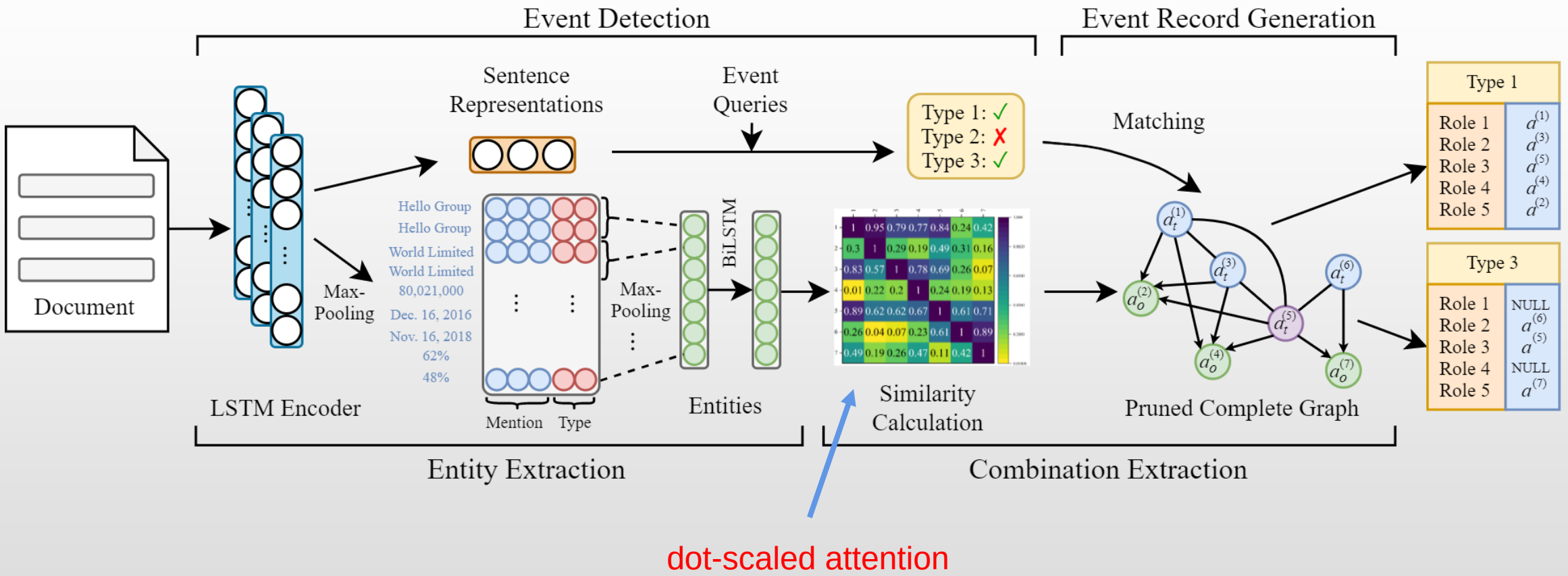
Roles	Existence	Distinguishability	Importance	Sort
{, }	$\frac{1+1+1+1+1+1}{6}$	$\frac{1+1+1+1+1+1}{6}$	$1 \times 1 = 1$	→ {, } ✓
{, }	$\frac{1+1+1+1+1+1}{6}$	$\frac{0+0+1+1+1+1}{6}$	$1 \times \frac{4}{6} = \frac{2}{3}$	→ {, }
{, }	$\frac{1+1+1+1+1+1}{6}$	$\frac{1+1+1+1+1+1}{6}$	$1 \times 1 = 1$	→ {, }
{, }	$\frac{1+1+1+1+1+1}{6}$	$\frac{1+1+1+1+1+1}{6}$	$1 \times 1 = 1$	→ {, }
{, }	$\frac{1+1+1+1+1+1}{6}$	$\frac{1+1+1+1+1+1}{6}$	$1 \times 1 = 1$	→ {, }
{, }	$\frac{1+0+1+1+0+1}{6}$	$\frac{1+1+1+1+1+1}{6}$	$\frac{4}{6} \times 1 = \frac{2}{3}$	→ {, }

$\mathcal{D}_1 =$ Krabs, Plankton, Dec. 16, 2016, Sandy, Squidward, Nov. 16, 2016, Pearl, 3,456,000

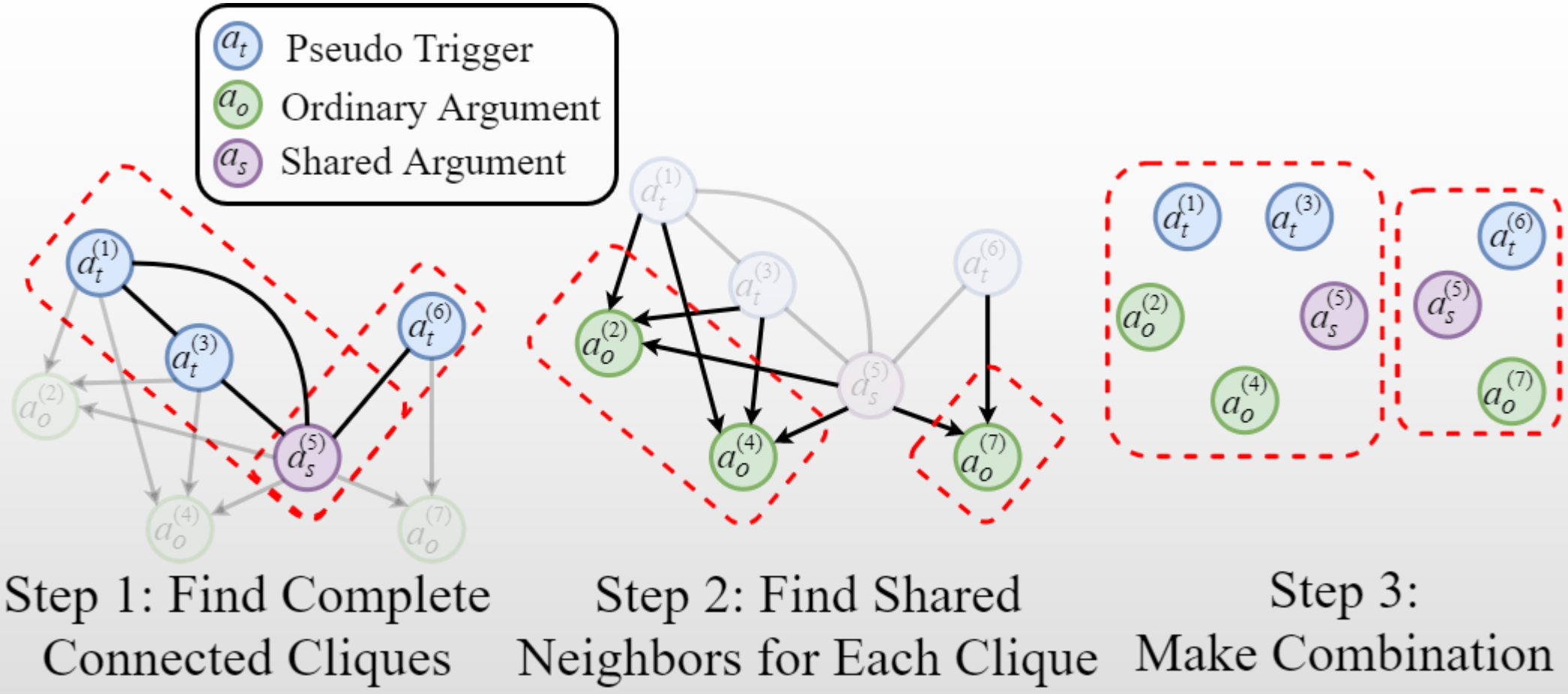
$\mathcal{D}_2 =$ Jan. 16, 1993, 6,800, Patrick, SpongeBob, Gary, Dec. 16, 2016

重要性指标最大的一组触发词论元角色对应的论元就是实例中的伪触发词

如何让模型预测出“剪枝完全图”



如何将剪枝完全图还原成事件实例



Bron-Kerbosch 算法可解

效果怎么样

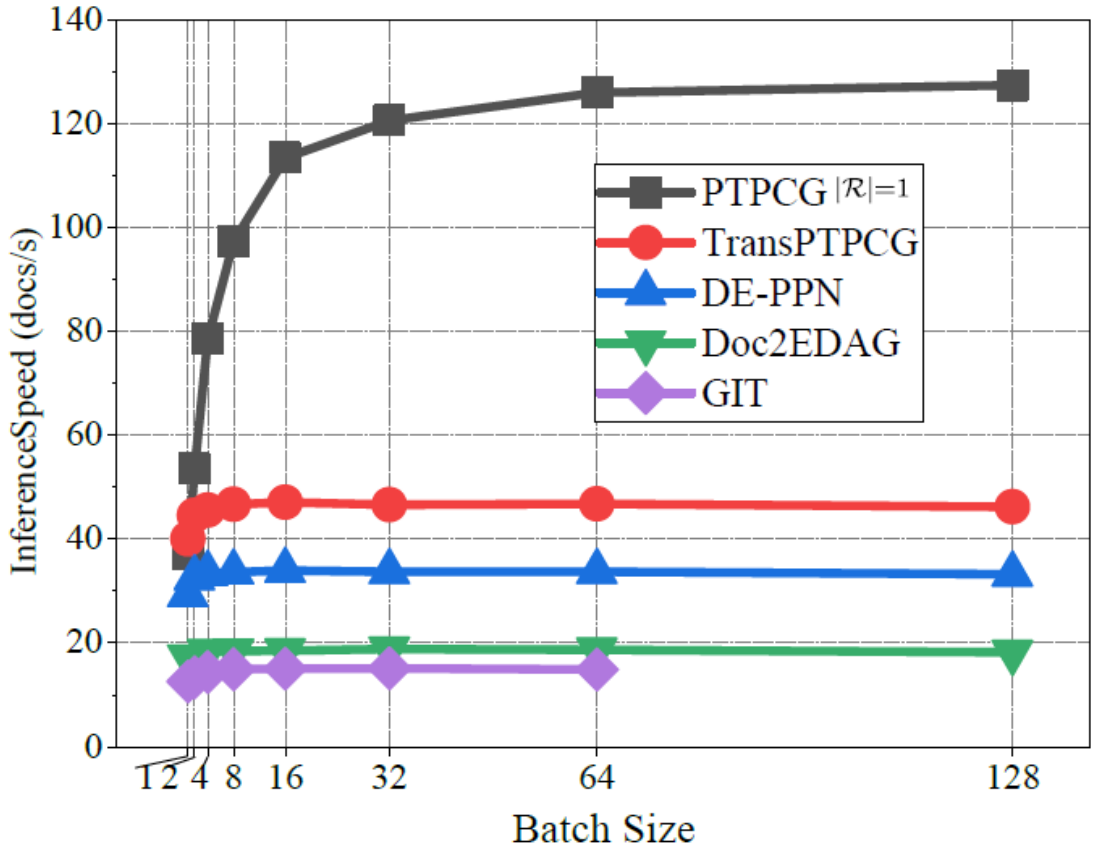
有向无环图方法

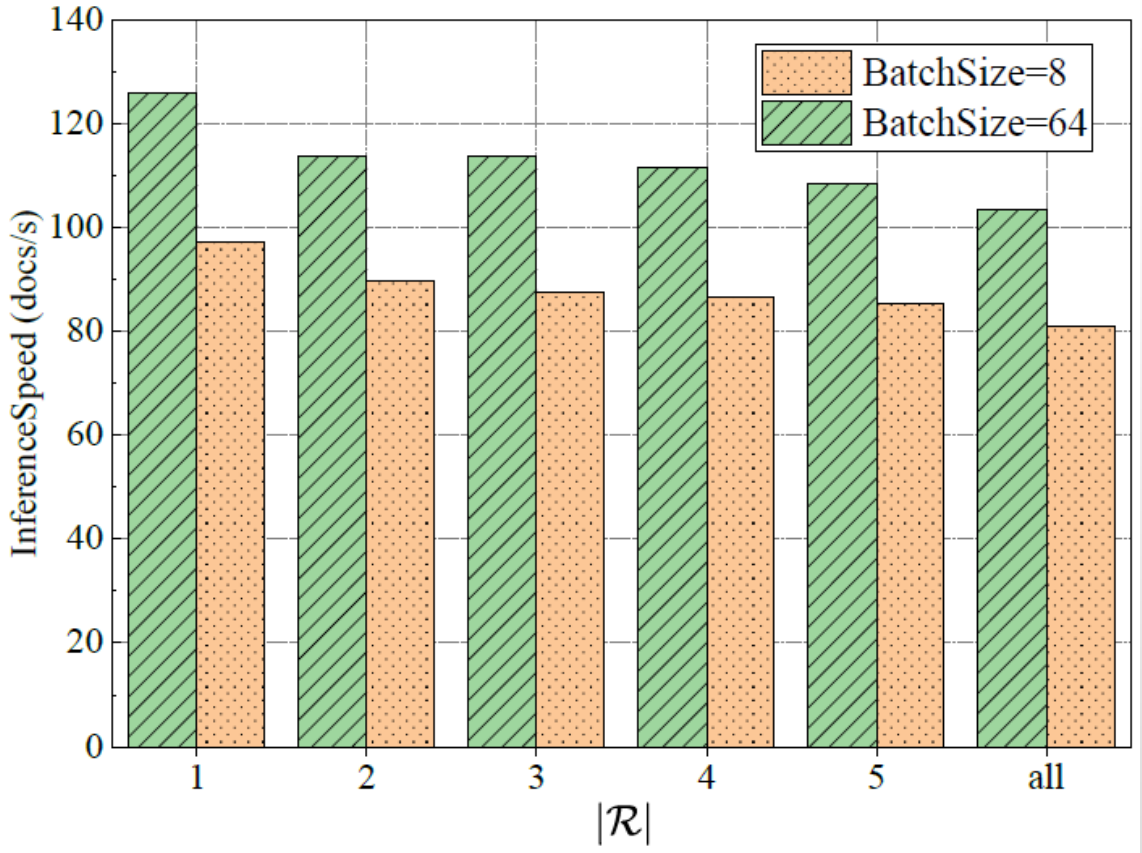
Model	#Params (w/o Emb)	GPU Hours	ChFinAnn-Single			ChFinAnn-All			DuEE-fin w/o Tgg			DuEE-fin w/ Tgg		
			P	R	F1	P	R	F1	P	R	F1	P	R	F1
DCFEE-O*	32M (16M)	192.0	73.2	71.6	72.4	69.7	57.8	63.2	56.2	48.2	51.9	51.9	49.6	50.7
DCFEE-M*	32M (16M)	192.0	64.9	71.7	68.1	60.1	61.3	60.7	38.7	52.3	44.5	37.3	48.6	42.2
GreedyDec*	64M (48M)	604.8	83.9	77.3	80.4	81.9	51.2	63.0	59.6	41.8	49.1	59.0	42.1	49.2
Doc2EDAG*	64M (48M)	604.8	83.2	89.3	86.2	81.1	77.0	79.0	66.7	50.0	57.2	67.1	51.3	**58.1**
GIT*	97M (81M)	633.6	85.0	88.7	86.8	82.4	**77.6**	**79.9**	**68.2**	43.4	53.1	**70.3**	46.0	55.6
DE-PPN*	119M(103M)	197.6	78.3	70.1	74.0	74.2	58.6	65.5	63.4	18.4	28.5	**70.3**	11.8	20.2
PTPCG$_{\|\mathcal{R}\|=1}$	32M (16M)	**24.0**	**86.3**	**90.1**	**88.2**	**83.7**	75.4	79.4	66.7	**54.6**	**60.0**	62.0	**54.8**	**58.1**

- 效果和DAG方法比差不多，甚至在单事件单实例上的效果比它们还好
- 模型参数非常少，只是GIT参数量的19.8%
- 训练速度非常快！单卡只要训练24小时，而GIT需要4卡训练将近1星期，GPU卡时是GIT的3.9%

折算为V100卡时，每个模型节省 ¥3658元（按1卡时¥6计算）

能不能部署，推理时间呢？


InferenceSpeed (docs/s)
Batch Size
PTPCG |R|=1
TransPTPCG
DE-PPN
Doc2EDAG
GIT
0
20
40
60
80
100
120
140
1
2
4
8
16
32
64
128


InferenceSpeed (docs/s)
|R|
BatchSize=8
BatchSize=64
0
20
40
60
80
100
120
140
1
2
3
4
5
all

我们有些好玩的发现

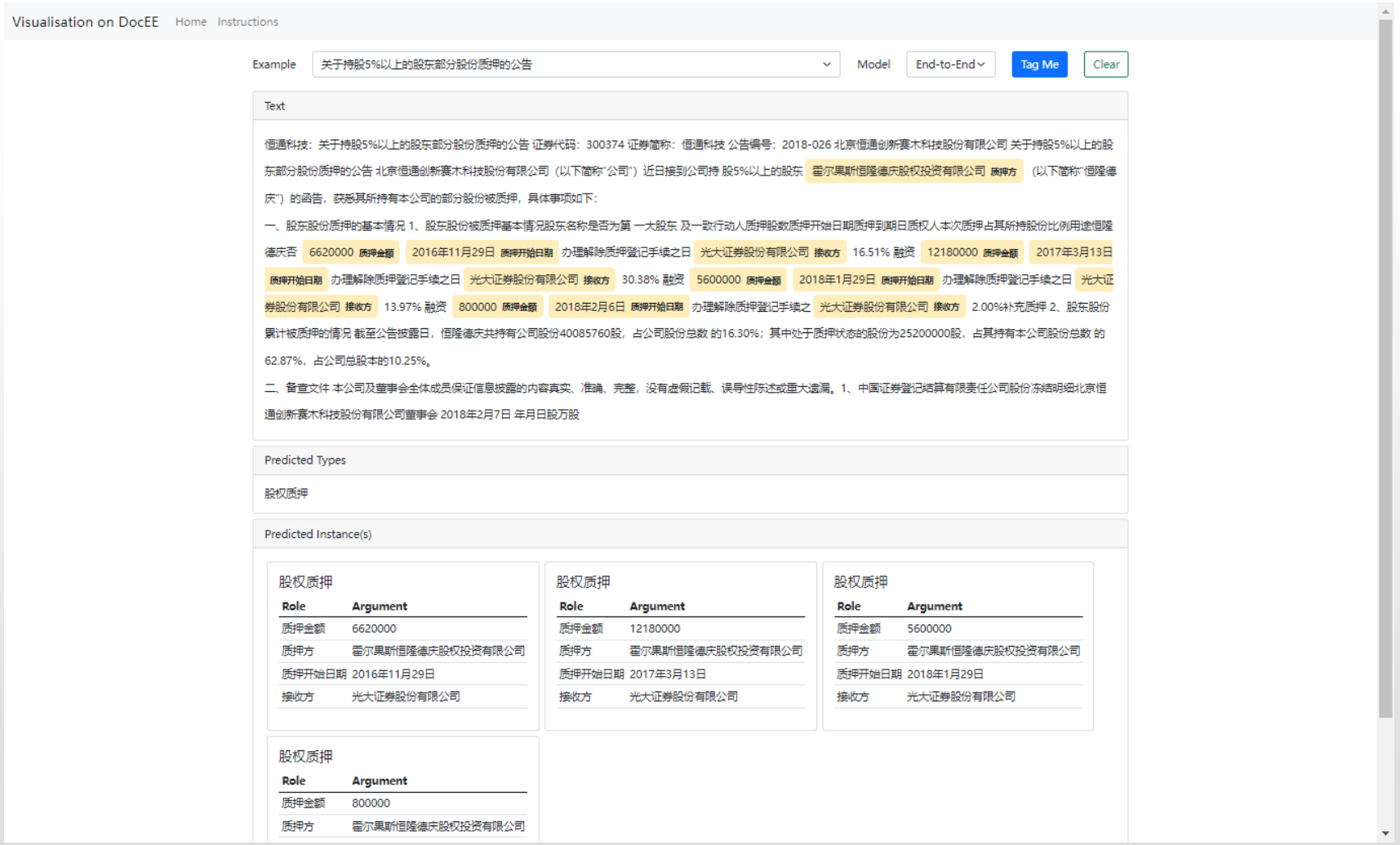
Model	Tgg	$\|\mathcal{R}\|$	Impt.	Dev			Online Test		
				P	R	F1	P	R	F1
Doc2EDAG	×	-	-	70.8	55.3	62.1	66.7	50.0	57.2
	✓	-	-	73.7	59.8	66.0	67.1	51.3	58.1
GIT	×	-	-	72.4	58.4	64.7	68.2	43.4	53.1
	✓	-	-	**75.4**	61.4	**67.7**	**70.3**	46.0	55.6
PTPCG	✓	0	62.9	73.5	59.4	65.7	67.0	50.1	57.3
	✓	1	93.7	68.8	64.2	66.4	62.0	54.8	58.1
	✓	2	97.1	64.7	**64.9**	64.8	59.1	56.5	57.8
	×	1	83.8	71.0	61.7	66.0	66.7	54.6	**60.0**
	×	2	94.3	63.8	64.8	64.3	60.2	58.4	59.3
	×	3	97.2	56.7	64.3	60.3	52.6	**58.9**	55.6

- DuEE-fin数据集包含了触发词标注，但存在触发词共享的情况（区分性不为100%）
- 使用伪触发词可以辅助提升结果
- 只使用伪触发词的效果比只使用金标触发词的结果还要好！

$\lvert\mathcal{R}\rvert$	Impt.	SE	ME	TotE	#links	Adj Acc.	F1
1	88.3	5.0	37.5	14.6	10,502	65.8	79.4
2	95.7	1.0	20.4	6.7	23,847	59.1	77.7
3	97.2	0.9	18.0	5.9	55,961	56.7	74.9
4	97.6	0.5	16.9	5.3	75,334	58.2	74.0
5	97.8	0.4	13.9	4.4	88,752	59.5	73.1
all	97.8	0.2	13.4	4.1	140,989	60.1	69.5

- 理论上限不为100%
  - 使用BK算法解码具有一定的理论误差，但当前模型的效果离理论误差的距离还很远
- 实体预测的结果对最终效果影响巨大
  - 当我们使用金标实体进行预测时，最终的整体F1值可以提升至少10%
- 随着伪触发词数量的增加，结果在不断下降
  - 主要原因：图中连接的数量随着伪触发词的增加而增加，预测难度也在不断加大
  - 我们需要对相似度计算和连接预测部分做进一步的优化

我们有个Demo！



https://github.com/Spico197/DocEE

http://hlt.suda.edu.cn/docee

Thanks

Q&A

Tong Zhu
tzhu7@stu.suda.edu.cn

https://github.com/Spico197/DocEE