Solving Hard Coreference Problems

Haoruo Peng, Daniel Khashabi and Dan Roth

One fundamental difficulty of **coreference resolution** is to resolve instances that require background **knowledge**. In this work, we propose an algorithmic solution that involves a new representation for the knowledge required to address **hard coreference problems**, along with a constrained optimization framework that uses this knowledge in coreference decision making.

Problem Description

◆ Problems with Existing Coref Systems

- Rely heavily on gender / plurality information

  e.g. *[John] adores [Mary] because [she] is pretty.*
- Hard Coreference Instances

  e.g. [A bird] perched on the [limb] and [it] bent.

  [A bird] perched on the [limb] and [it] sang.

◆ Goal

A better overall coreference system

- Improve on solving hard coreference problems
- Maintain the state-of-art performance on standard coreference problems

Predicate Schemas

◆ Characteristic of Required Knowledge

- Associated with "predicate"

◆ Type 1 Predicate Schema

$pred_m(m,a)$ *Predicate_of_m*(sub, obj)

[The bee] landed on [the flower] because [it] had pollen.

S(*have*(m=*[the flower]*, a=*[pollen]*)) >

S(*have*(m=*[the bee]*, a=*[pollen]*))

◆ Type 2 Schema

$pred_m(m,a)|\hat{pred}_m(m,\hat{a}), cn$

[Jim] was afraid of [Robert] because [he] gets scared around new people.

S(*be afraid of* (m=*, a=*) | *get scared around* (m=*, a=*), *because*) >

S(*be afraid of* (a=*, m=*) | *get scared around* (m=*, a=*), *because*)

◆ Schema Variations

Type 1	$\mathcal{S}(pred_m(m,a))$ $\mathcal{S}(pred_m(a,m))$ $\mathcal{S}(pred_m(m,*))$ $\mathcal{S}(pred_m(*,m))$
Type 2	$\mathcal{S}\left(pred_m(m,a)\,\vert\,\widehat{pred}_m(m,\hat{a}), cn\right)$ $\mathcal{S}\left(pred_m(a,m)\,\vert\,\widehat{pred}_m(m,\hat{a}), cn\right)$ $\mathcal{S}\left(pred_m(m,a)\,\vert\,\widehat{pred}_m(\hat{a},m), cn\right)$ $\mathcal{S}\left(pred_m(a,m)\,\vert\,\widehat{pred}_m(\hat{a},m), cn\right)$ $\mathcal{S}\left(pred_m(m,*)\,\vert\,\widehat{pred}_m(m,*), cn\right)$ $\vdots$

◆ Example Beyond Above Schemas

[Lakshman] asked [Vivan] to get him some ice cream because [he] was hot.

Utilizing Knowledge

◆ Knowledge as Features

$$f_{u,v} = \mathbf{w}^T\phi(u,v) + \tilde{\mathbf{w}}^T\mathbf{s}(u,v)$$

Pairwise Mention Scoring Function

Scoring Function for Predicate Schemas

- Noise in Knowledge
- Implicit Textual Inference

◆ Knowledge as Constraints

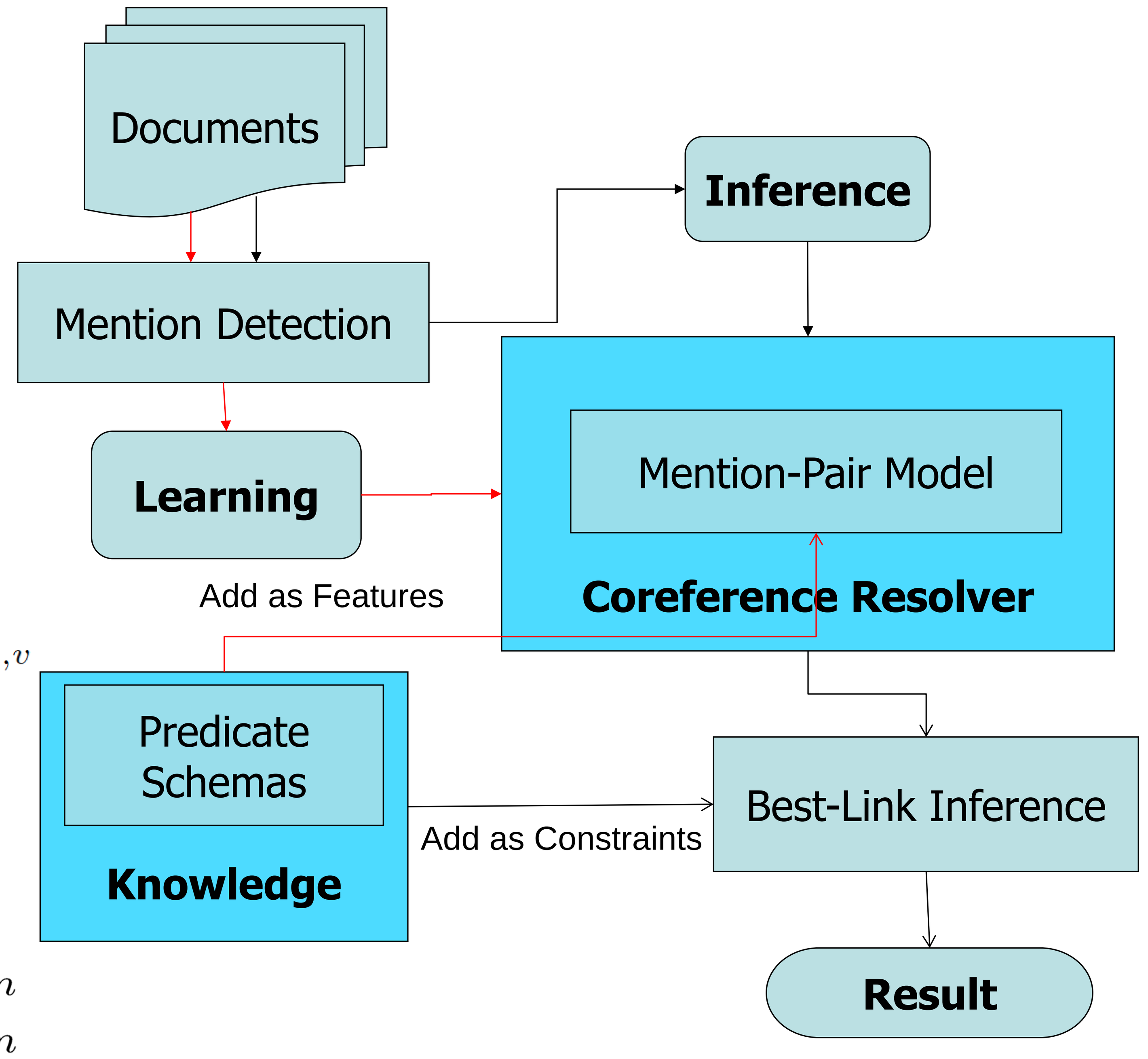
- Generating Constraints

$$\begin{cases} \text{if } s_i(u,v) \geq \alpha_i s_i(w,v) \Rightarrow y_{u,v} \geq y_{w,v}, \\ \text{if } s_i(u,v) \geq s_i(w,v) + \beta_i \Rightarrow y_{u,v} \geq y_{w,v} \end{cases}$$

- ILP Inference (Best-Link)

$$\arg\max_y \sum_{u,v} w_{uv} y_{uv}$$
$$\text{s.t. } \sum_{u<v} y_{uv} \leq 1, \forall v$$
$$y_{uv} \in \{0,1\}, \forall u,v$$
$$\text{if } s_{uv} \geq t + s_{um} \text{ then } y_{uv} \geq y_{um}$$
$$\text{if } s_{uv} \geq t' \cdot s_{um} \text{ then } y_{uv} \geq y_{um}$$

Learning and Inference Framework



Knowledge Acquisition

◆ Gigaword co-occurrences

- Extract instances of Type 1 and 2 Predicate Schemas from Gigaword
- Use counts as scores

e.g. *The bee landed on the flower because the bee wanted pollen.*

S_{giga}(*land*(m=*[the bee]*, a=*[flower]*)) **+1** S_{giga}(*want*(m=*[the bee]*, a=*[pollen]*)) **+1**

e.g. *The police arrested [John] because [John] robbed Jack.*

S_{giga}(*arrest*(a=*[the police]*, m=*[John]*) | *rob*(m=*[John]*, a=*[Jack]*), because) **+1**

Use chunker + dependency parser to extract (sub, predicate, obj) triples)

Use heuristics to determine coreferent mentions in triples

◆ Wikipedia Disambiguated Co-occurrences

- Extract **disambiguated** noun, verbs and entities, etc. in Wikipedia (Illinois-Wikifier)
- Use statistics: (immediately) after / (immediately) before to approximate Type 1 Predicate Shema

e.g. S_{wiki}(*land*(m=*[the bee]*, a=*)) **+1** S_{wiki}(*want*(m=*[the bee]*, a=*)) **+1**

e.g. S_{wiki}(*land*(a=*, m=*[flower]*)) **+1** S_{wiki}(*want*(a=*, m=*[pollen]*)) **+1**

◆ Web Search Statistics

- Generate web queries to approximate Type 1 Predicate Schema
- Use returned Google counts as scores

e.g. S_{web}(*want*(m=*[the bee]*, a=*[pollen]*)): "the bee want", "the bee want pollen", "the bee pollen"

e.g. S_{web}(*want*(m=*[flower]*, a=*[pollen]*)): "flower want", "flower want pollen", "flower pollen"

◆ Polarity Information

- Initialize polarities of the mention according to its predicate (Wilson et al, 2005)
- Negate the polarity when the mention role is *object*
- Reverse polarity when there is a polarity reversing connective (such as "but")

e.g. Steve annoyed Carl, but he liked to hang out with Chris.

S_{pol}(*annoy*(m=*[Steve]*, a=*[Carl]*) | *like*(m=*[he]*, a=*[Chris]*), but) = **NOT (-1 AND +1) = +1**

Results

◆ Datasets

- Winograd (Rahman&Ng, 2012)
- Winocoref: Winograd plus more mentions

e.g. *[Stephen] threw the bags of [Johnny] into the water since [he] mistakenly asked [him] to carry [his] bags.*

- Standard Coref: ACE, Ontonotes

◆ Metrics

- Precision (Winograd, binary classification)
- AntePre
  - k pronouns in sentence
  - Each pronoun has n_i antecedents
  - Altogether m correct binary decisions

$$\text{AntePre} = \frac{m}{\sum_{i=1}^{k} n_i}$$

- MUC, BCUB, CEAF (standard coref)

◆ Evaluations:

- Hard Coreference Problems

Datasets	Winograd	Winocoref
Metrics	Precision	AntePre
State of the Art Coref (ILL)	51.48	68.37
Rahman and Ng (2012)	73.05	------
KnowFeat (Our paper)	71.81	88.48
KnowCons (Our paper)	74.93	88.95
KnowComb (Our paper)	**76.41**	**89.32**

- Standard Coreference Problems

System	MUC	BCUB	CEAFe	AVG
ACE				
IlliCons	**78.17**	81.64	**78.45**	**79.42**
KnowComb	77.51	**81.97**	77.44	78.97
OntoNotes				
IlliCons	84.10	**78.30**	**68.74**	**77.05**
KnowComb	**84.33**	78.02	67.95	76.76

- Ablation Study
  - We categorized instances in Winocoref data
    - Cat1 / Cat2: Instances that can be solved using Type 1 / Type 2 Predicate Schemas
    - Cat 3: All remaining instances
    - Cat1 – 16.8% Cat2 - 56.2% Cat3 - 27.0%
  - Schemas evaluated on each category

Schema	AntePre(Test)	AntePre(Train)
Type 1	76.67	86.79
Type 2	79.55	88.86
Type 1 (Cat1)	90.26	93.64
Type 2 (Cat2)	83.38	92.49

This work is supported by NSF and DARPA