Human Language Technology Research Institute

SinoCoreferencer: An End-to-End Chinese Event Coreference Resolver

Chen Chen and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas

Event Coreference Resolution

- Determine which event mentions in a text refer to the same real-world event

Event Coreference Resolution

- Determine which event mentions in a text refer to the same real-world event

Since there is little work on event coreference, our understanding of this task is fairly limited

Goal

- Understand how a state-of-the-art end-to-end event coreference resolver can be improved

Goal

- Understand how a state-of-the-art end-to-end event coreference resolver can be improved
  - An event coreference system lies towards the end of the standard information extraction pipeline

Goal

- Understand how a state-of-the-art end-to-end event coreference resolver can be improved
  - An event coreference system lies towards the end of the standard information extraction pipeline
  - To what extent is the noisy output of each of its upstream components limiting the performance?

Goal

- Understand how a state-of-the-art end-to-end event coreference resolver can be improved
  - An event coreference system lies towards the end of the standard information extraction pipeline
  - To what extent is the noisy output of each of its upstream components limiting the performance?

Focus on Chinese event coreference resolution

Why Chinese Event Coreference?

- Provide publicly available results
  - related work mostly on English event coreference
    - Humphreys at el.(1997), Chen et al.(2009), Bejan and Harabagiu(2010), Chen et al.(2011), Lee et al.(2012), …

Why Chinese Event Coreference?

- Provide publicly available results
  - related work mostly on English event coreference
    - Humphreys at el.(1997), Chen et al.(2009), Bejan and Harabagiu(2010), Chen et al.(2011), Lee et al.(2012), …
- Provide a publicly available implementation
  - SinoCoreferencer contains the major components of a typical information extraction pipeline
    - Each component can be run in a standalone manner
    - Complements the Stanford Chinese NLP tools

Plan for the Talk

- ACE Event Coreference
- System Architecture
- Evaluation

Plan for the Talk

- ACE Event Coreference
- System Architecture
- Evaluation

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]
  - [stabbed] and [criminal] are coreferent because they refer to the same real-world event

ACE Event Coreference: Example

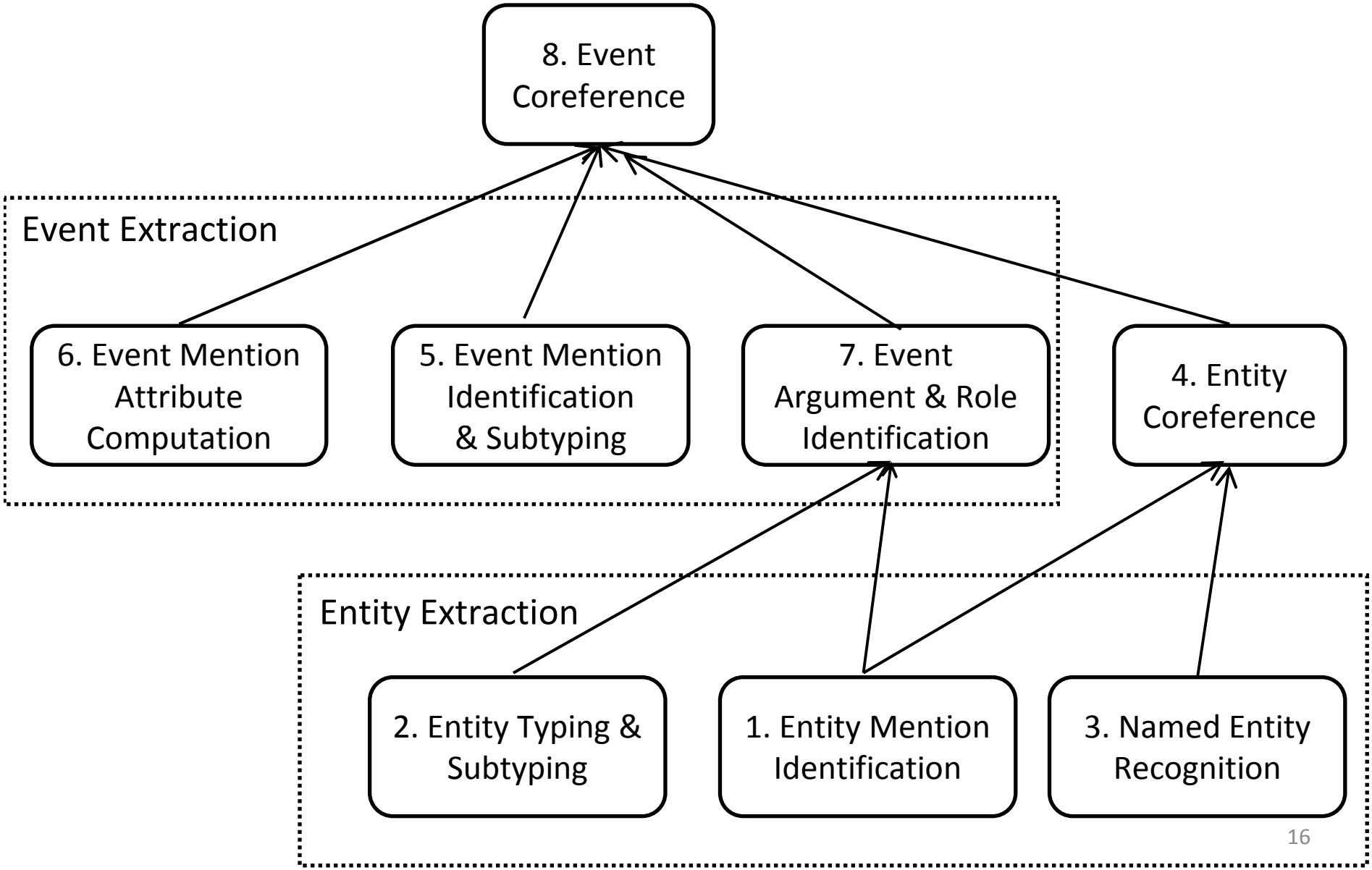
(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]
  - [stabbed] and [criminal] are coreferent because they refer to the same real-world event
- **Restricted** event coreference
  - Considers only event mentions belonging to certain types
    - ACE 2005: 7 event types and 33 event subtypes

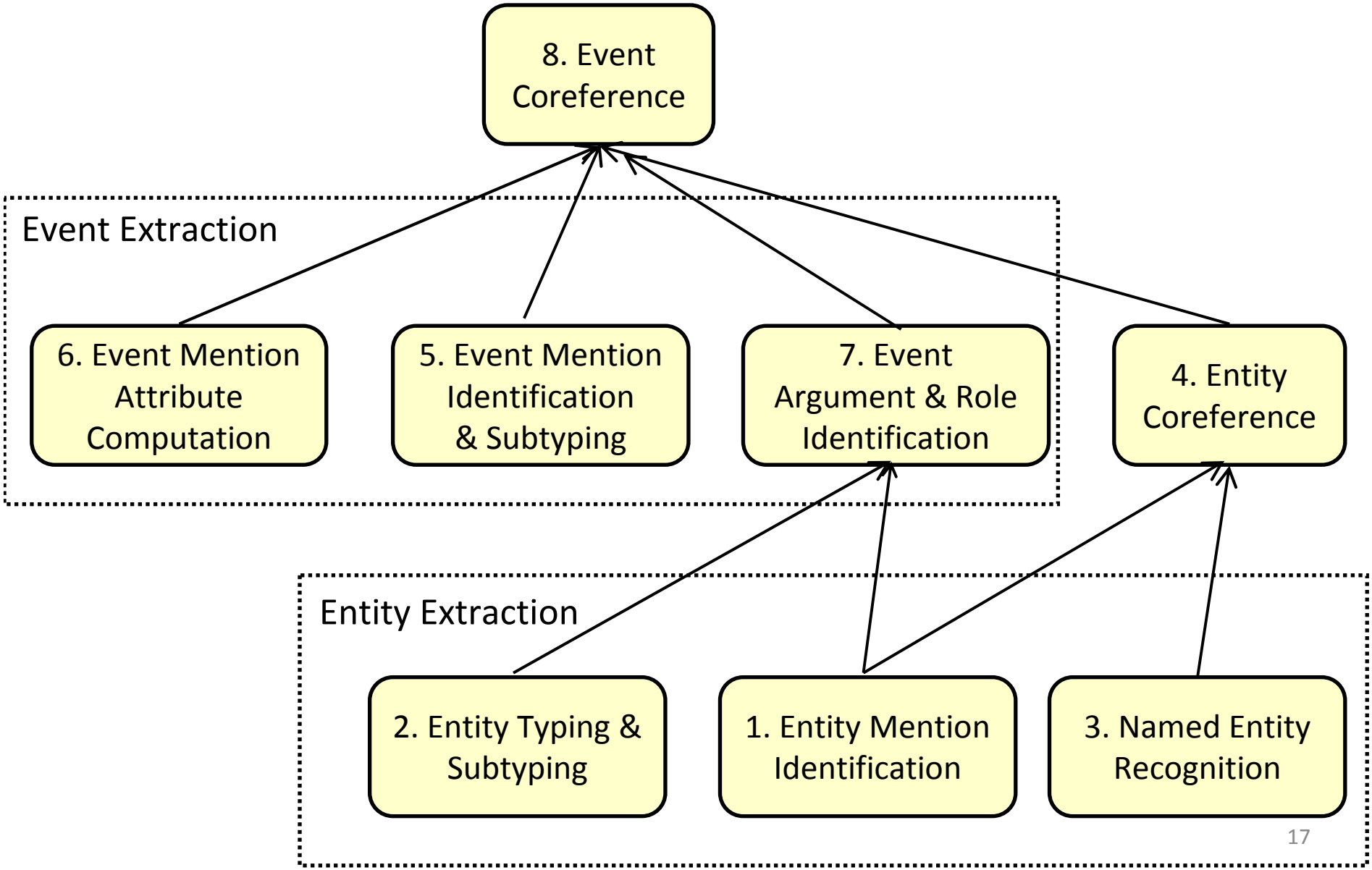
Plan for the Talk

- ACE Event Coreference
- System Architecture
- Evaluation

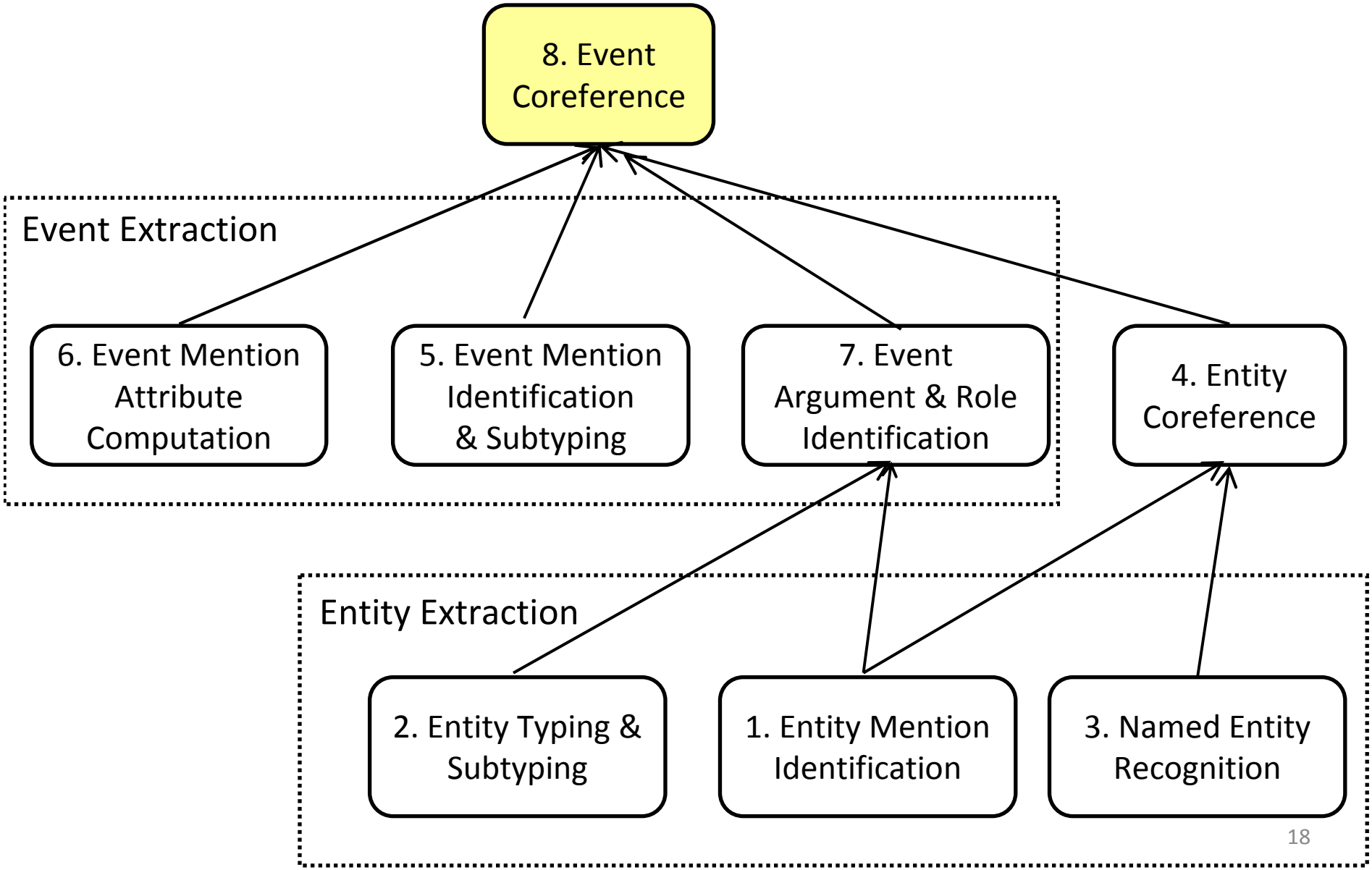
SinoCoreferencer: System Architecture



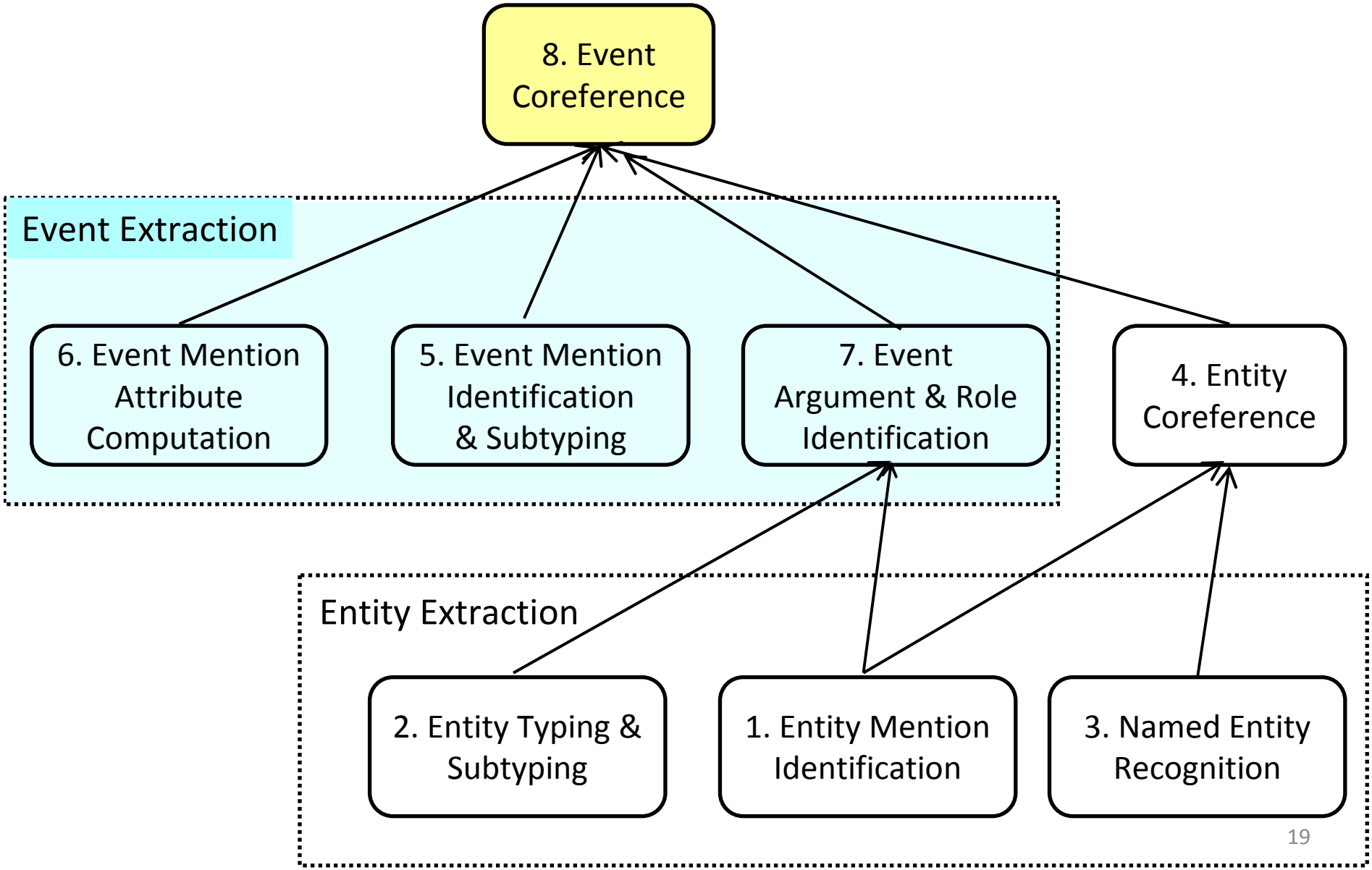
SinoCoreferencer: System Architecture



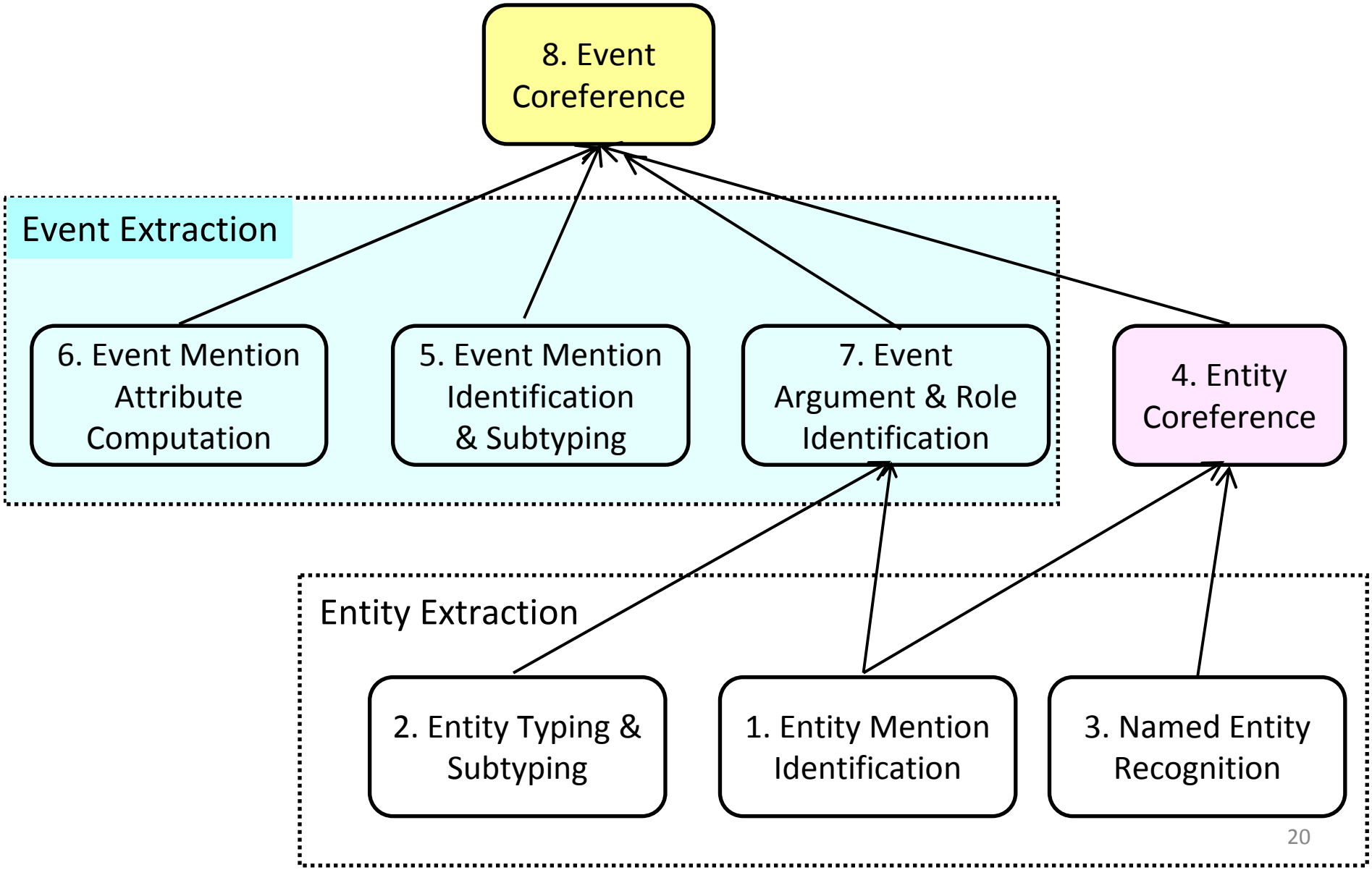
SinoCoreferencer: System Architecture



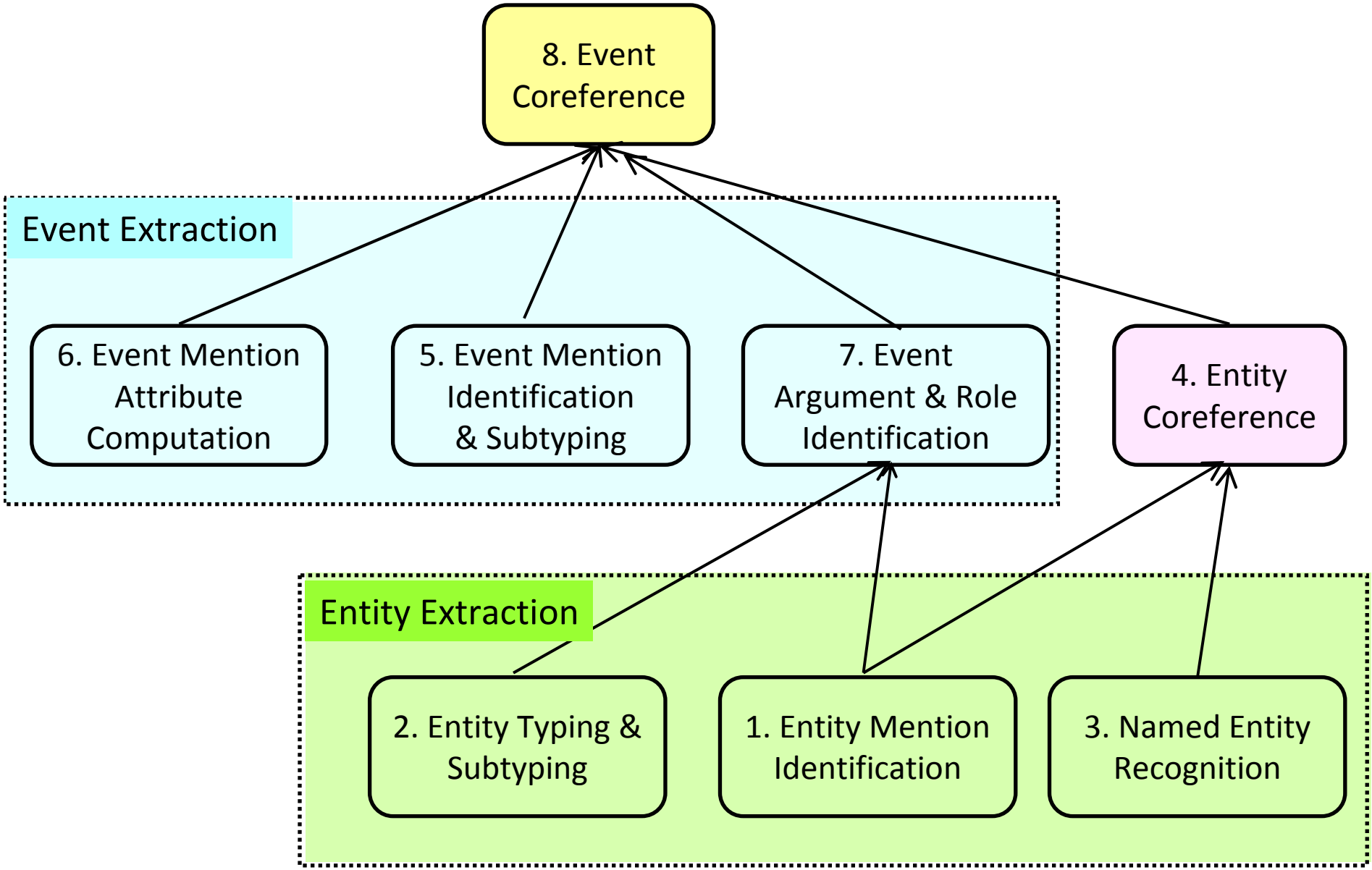
SinoCoreferencer: System Architecture



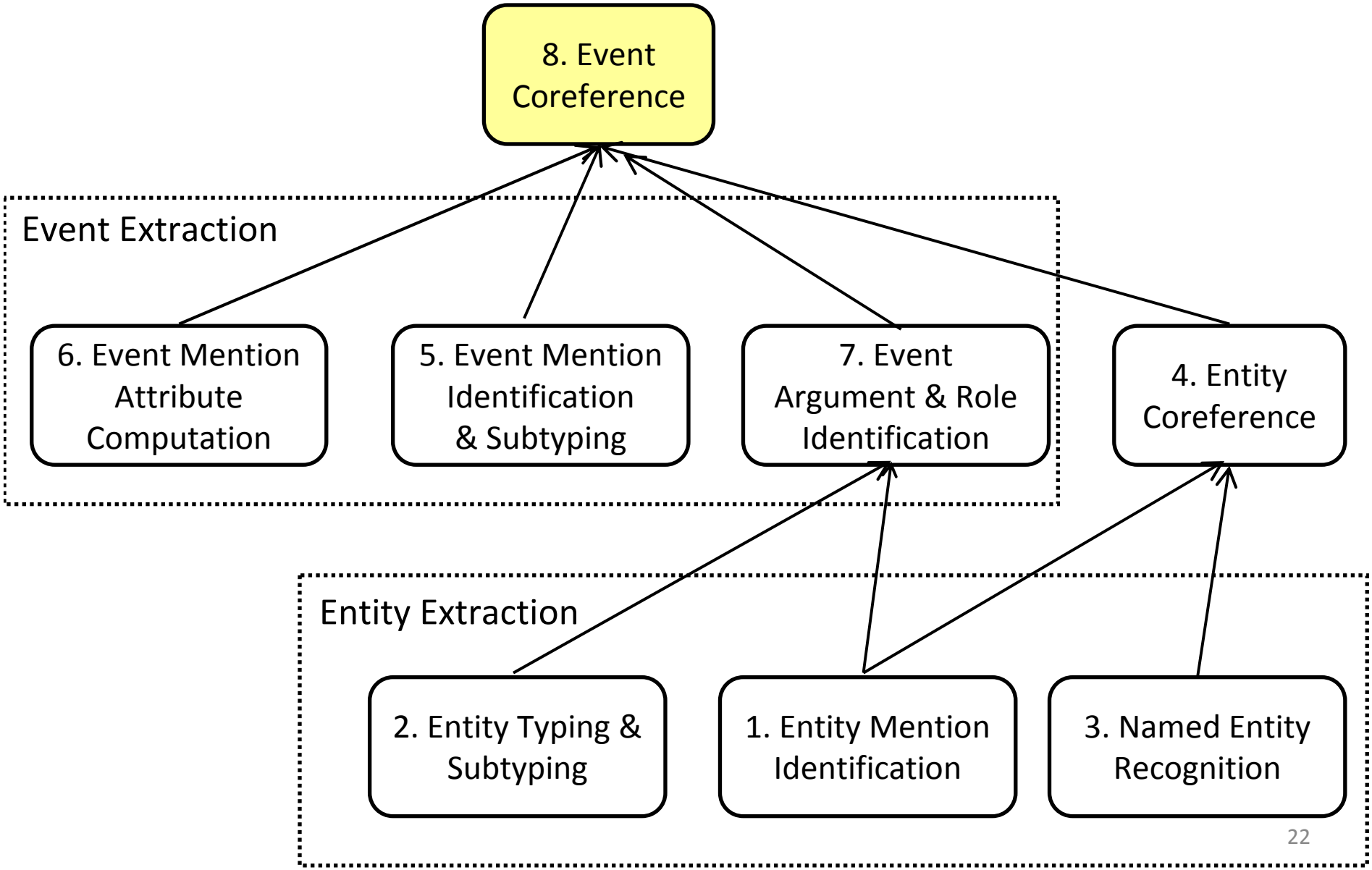
SinoCoreferencer: System Architecture



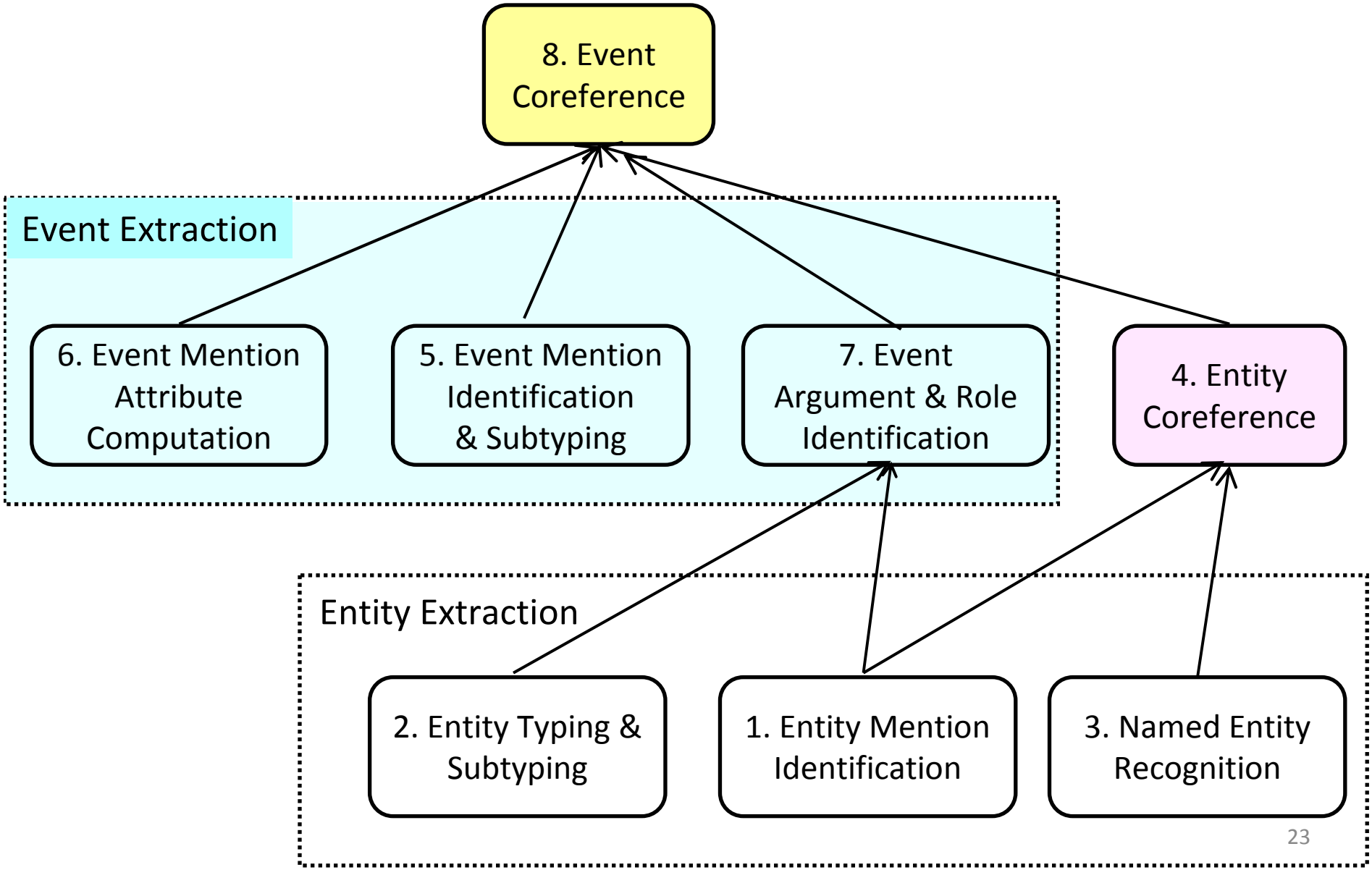
SinoCoreferencer: System Architecture



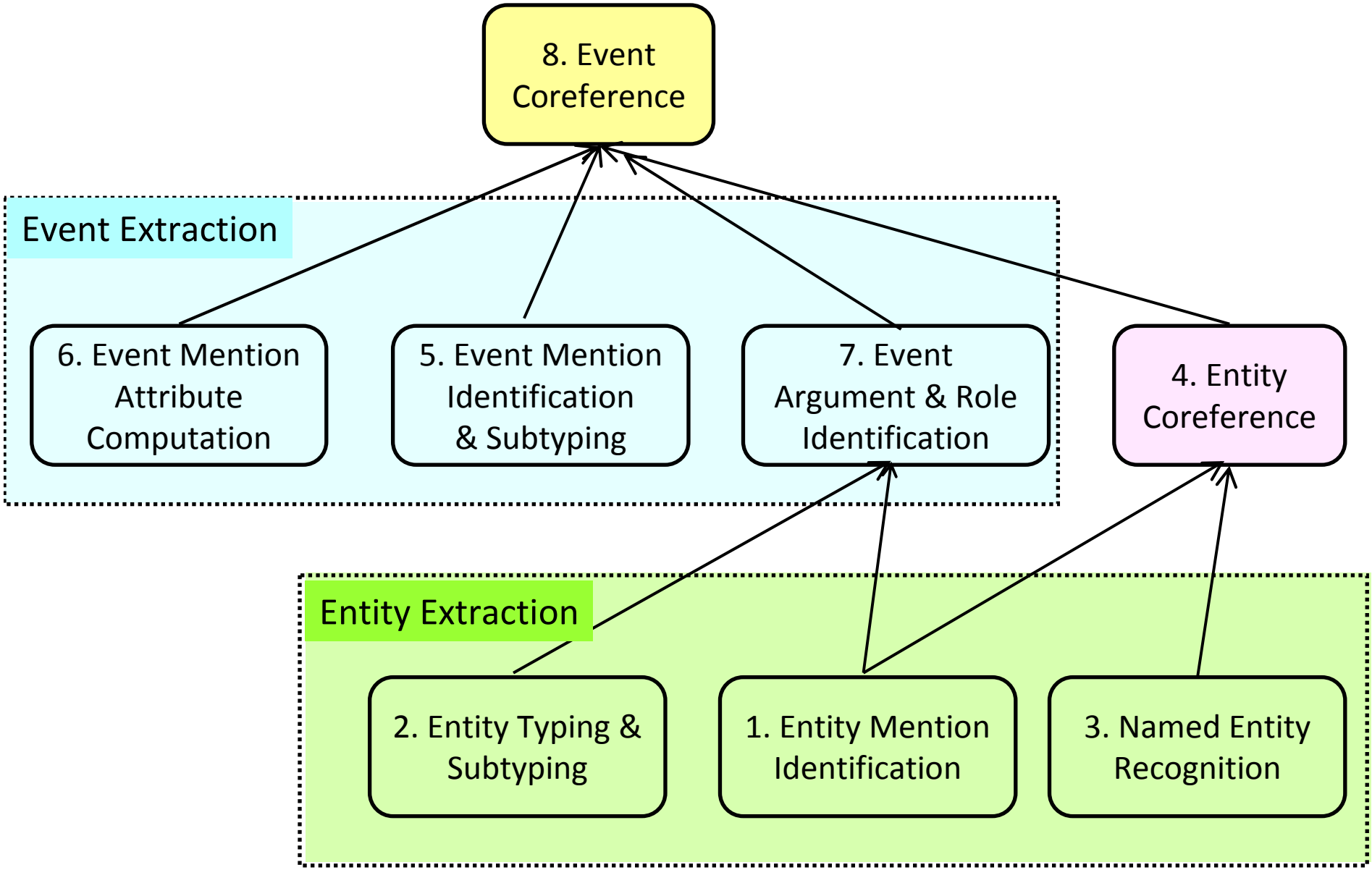
SinoCoreferencer: System Architecture



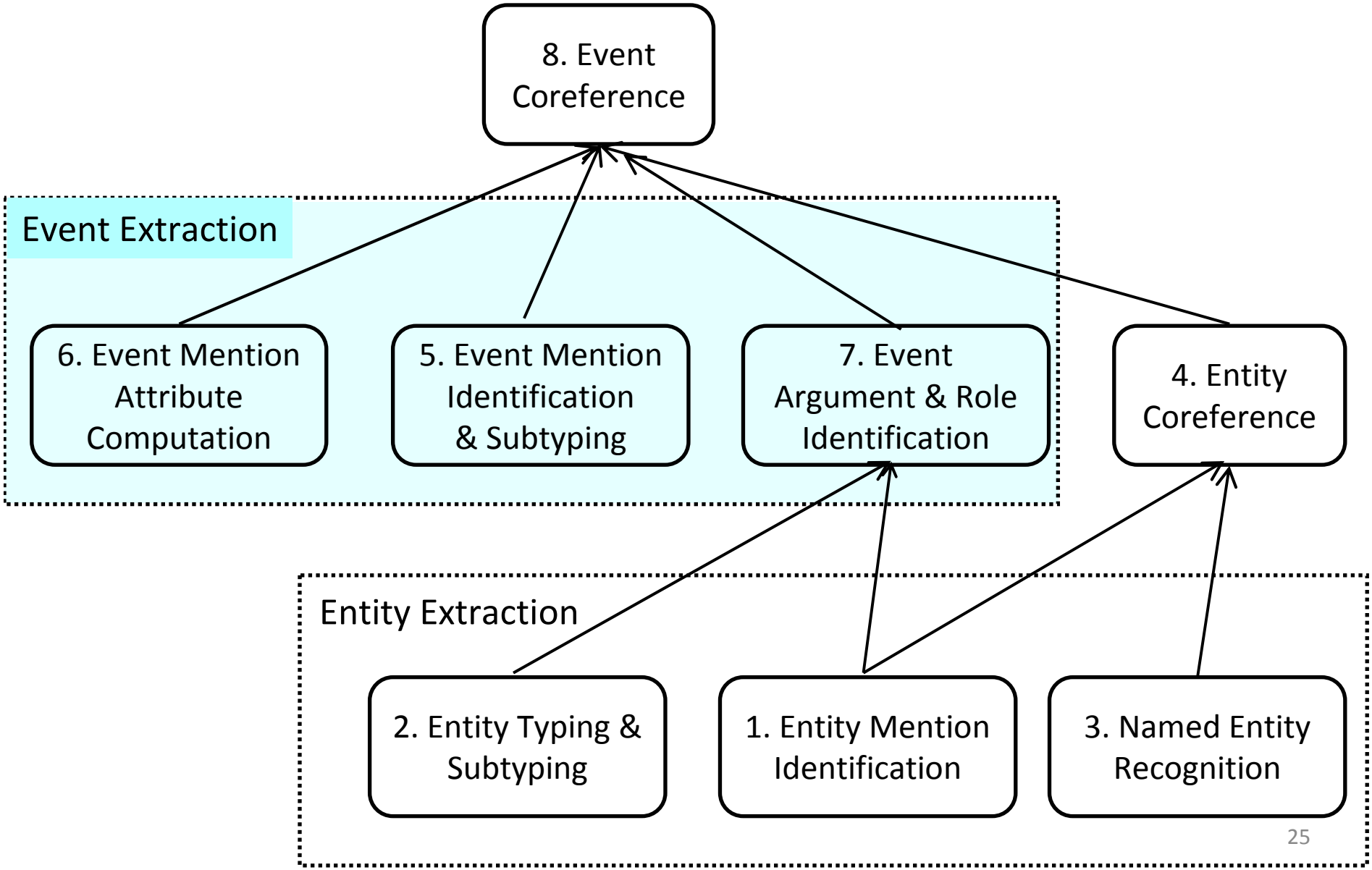
SinoCoreferencer: System Architecture



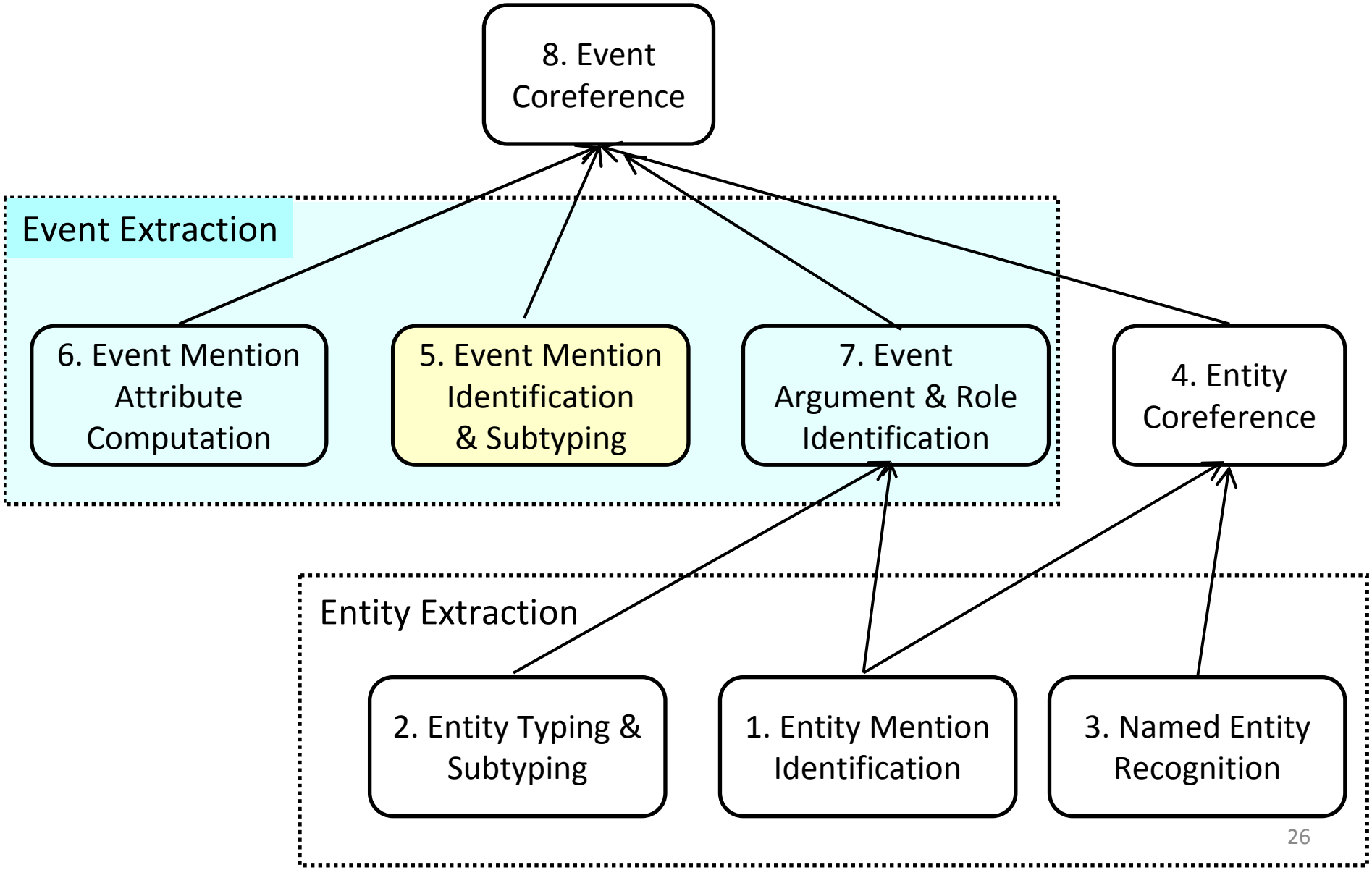
SinoCoreferencer: System Architecture



SinoCoreferencer: System Architecture



SinoCoreferencer: System Architecture



Event Mention Identification & SubTyping

- Goals
  - Identify the event mentions
  - Label each event mention with its event subtype

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]

Event mentions	Subtype
injured	**INJURE**
stabbed	**ATTACK**
criminal	**ATTACK**

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]

Event mentions	Subtype
injured	**INJURE**
stabbed	**ATTACK**
criminal	**ATTACK**

33 event subtypes defined in ACE 2005

Event Mention Identification & SubTyping

- Goals
  - Identify the event mentions
  - Label each event mention with its event subtype

- Why is this component useful for event coreference?

Event Mention Identification & SubTyping

- Goals
  - Identify the event mentions
  - Label each event mention with its event subtype

- Why is this component useful for event coreference?
  - Provide the event mentions for event coreference
  - Subtyping information is useful for determining whether two event mentions are coreferent
    - » Two event mentions with different event subtypes cannot be coreferent

Event Mention Identification & SubTyping

- How to implement this component?
  - train a CRF (using CRF++) to jointly perform event mention identification and subtypng

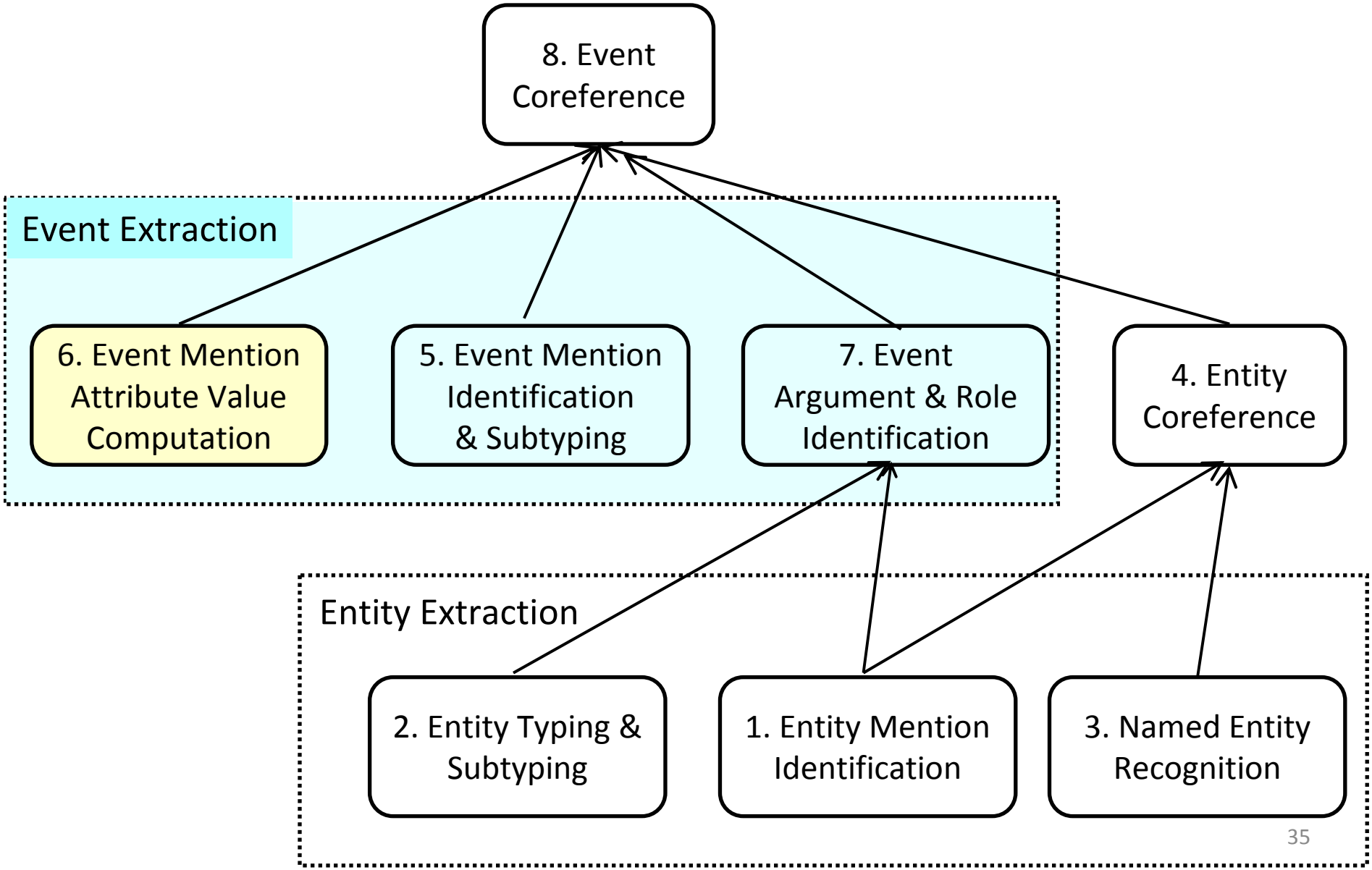
Event Mention Identification & SubTyping

- How to implement this component?
  - train a CRF (using CRF++) to jointly perform event mention identification and subtypng

- **Results** (on ACE 2005)

Identification			SubTyping		
R	P	F	R	P	F
60.0	71.3	65.1	56.4	67.1	61.3

SinoCoreferencer: System Architecture



Event Mention Attribute Value Computation

- Goal
  - Compute 4 attributes for each event mention: POLARITY, MODALITY, GENERICITY and TENSE

Event Mention Attribute Value Computation

- Goal
  - Compute 4 attributes for each event mention: POLARITY, MODALITY, GENERICITY and TENSE

Indicates whether the event happened or not

Verb tense (if the mention is a verb)

Event Mention Attribute Value Computation

- Goal
  - Compute 4 attributes for each event mention: POLARITY, MODALITY, GENERICITY and TENSE

Indicates whether the event happened or not

Verb tense (if the mention is a verb)

- Why is this component useful for event coreference?

Event Mention Attribute Value Computation

- Goal
  - Compute 4 attributes for each event mention: POLARITY, MODALITY, GENERICITY and TENSE

Indicates whether the event happened or not

Verb tense (if the mention is a verb)

- Why is this component useful for event coreference?
  - Two event mentions that differ in any of the four attributes cannot be coreferent

Event Mention Attribute Value Computation

- How to implement this component?
  - train 4 classifiers to predict these 4 event mention attributes, with one classifier per attribute

Event Mention Attribute Value Computation

- How to implement this component?
  - train 4 classifiers to predict these 4 event mention attributes, with one classifier per attribute

- **Results** (in terms of F-score)
  - Perfect vs. predicted event mentions

	POLARITY	MODALITY	GENERICITY	TENSE
Input	F	F	F	F
Perfect	96.5	86.9	91.2	67.1
Predicted	62.9	56.2	59.4	36.7

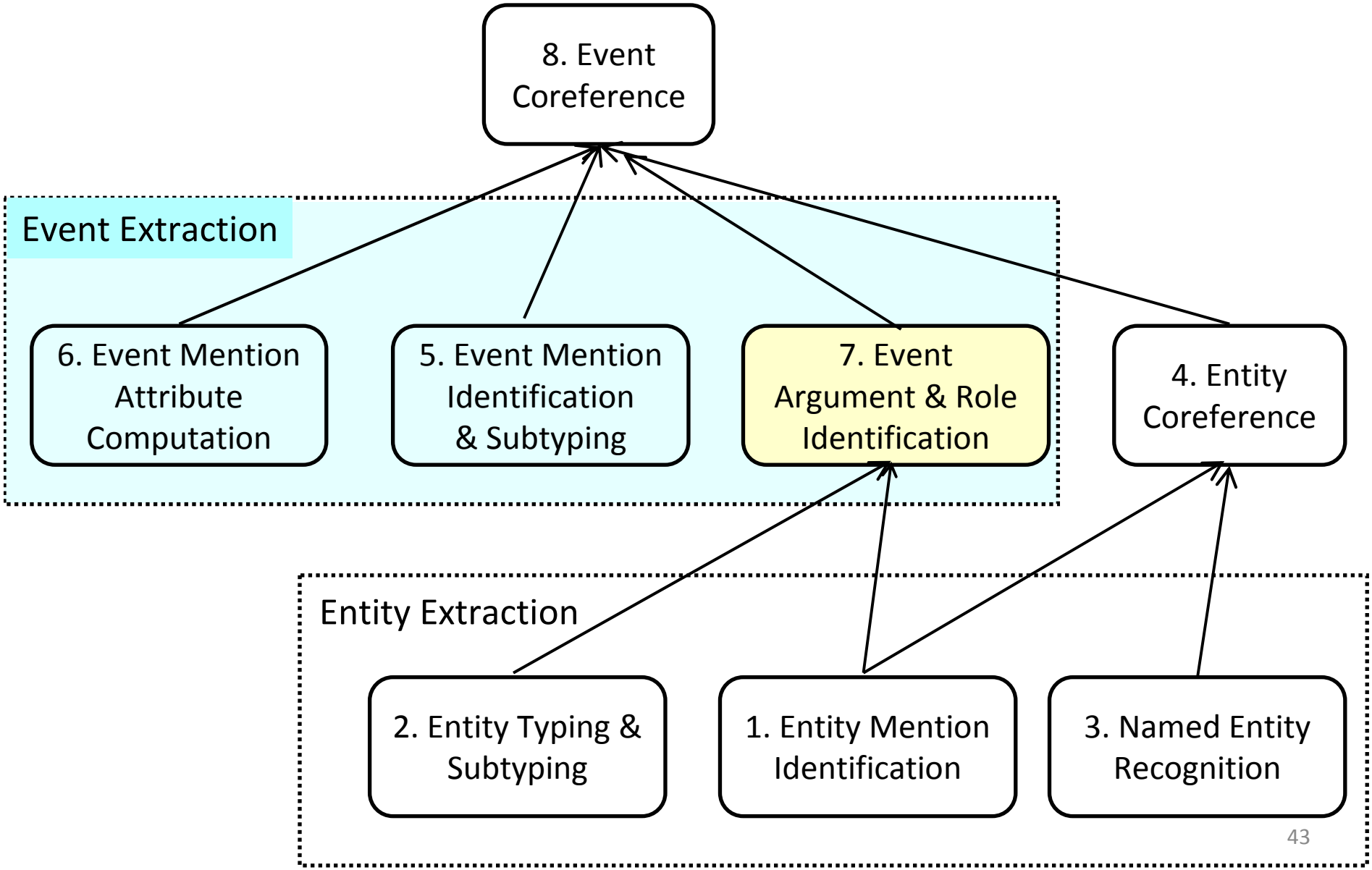
Event Mention Attribute Value Computation

- How to implement this component?
  - train 4 classifiers to predict these 4 event mention attributes, with one classifier per attribute

- **Results** (in terms of F-score)
  - Perfect vs. predicted event mentions

	POLARITY	MODALITY	GENERICITY	TENSE
Input	F	F	F	F
Perfect	96.5	86.9	91.2	67.1
Predicted	62.9	56.2	59.4	36.7

SinoCoreferencer: System Architecture



Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)

Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)
  - Assign a role (e.g. VICTIM, PLACE, TIME-WITHIN) to each argument

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]

Event mentions	Arguments and their ROLEs		
Criminal	The men **[ATTACKER]**	John Cole **[TARGET]**	

ACE Event Coreference: Example

(John Cole) was cycling on (the road) (yesterday) and was [injured] when (two men) [stabbed] (him) with (a knife). (The men)'s [criminal] motivation may have something to do with (John Cole)'s testimony in a criminal case.

- Three event mentions: [injured], [stabbed], [criminal]

Event mentions	Arguments and their ROLEs		
Criminal	The mens **[ATTACKER]**	John Cole **[TARGET]**	
stabbed	two men **[ATTACKER]**	him **[TARGET]**	a knife **[INSTRUMENT]**

Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)
  - Assign a role (e.g. VICTIM, PLACE, TIME-WITHIN) to each argument
- Why is this component useful for event coreference?

Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)
  - Assign a role (e.g. VICTIM, PLACE, TIME-WITHIN) to each argument
- Why is this component useful for event coreference?
  - Two event mentions can be coreferent only if the arguments in the corresponding roles are coreferent

Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)
  - Assign a role (e.g. VICTIM, PLACE, TIME-WITHIN) to each argument
- Why is this component useful for event coreference?

Event mentions	Arguments and their ROLEs		
Criminal	The men **[ATTACKER]**	John Cole **[TARGET]**	
stabbed	two men **[ATTACKER]**	him **[TARGET]**	a knife **[INSTRUMENT]**

Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)
  - Assign a role (e.g. VICTIM, PLACE, TIME-WITHIN) to each argument
- Why is this component useful for event coreference?

Event mentions	Arguments and their ROLEs		
Criminal	The men **[ATTACKER]**	John Cole **[TARGET]**	
stabbed	two men **[ATTACKER]**	him **[TARGET]**	a knife **[INSTRUMENT]**

Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)
  - Assign a role (e.g. VICTIM, PLACE, TIME-WITHIN) to each argument
- Why is this component useful for event coreference?

Event mentions	**Arguments and their ROLEs**		
Criminal	The men **[ATTACKER]**	John Cole **[TARGET]**	
stabbed	two men **[ATTACKER]**	him **[TARGET]**	a knife **[INSTRUMENT]**

Event Argument & Role Classification

- Goals
  - Identify arguments for an event mention (e.g., the participants, time, place)
  - Assign a role (e.g. VICTIM, PLACE, TIME-WITHIN) to each argument
- Why is this component useful for event coreference?

Event mentions	Arguments and their ROLEs		
Criminal	The men **[ATTACKER]**	John Cole **[TARGET]**	
stabbed	two men **[ATTACKER]**	him **[TARGET]**	a knife **[INSTRUMENT]**

Event Argument & Role Classification

- How to implement this component?
  - Implemented as part of our Chinese event extraction system (Chen and Ng, 2012)
    - Jointly learn the event arguments and their roles

Event Argument & Role Classification

- How to implement this component?
  - Implemented as part of our Chinese event extraction system (Chen and Ng, 2012)
    - Jointly learn the event arguments and their roles
- **Results** (in terms of R, P, F)
  - Perfect vs. predicted event mention boundary & subtype

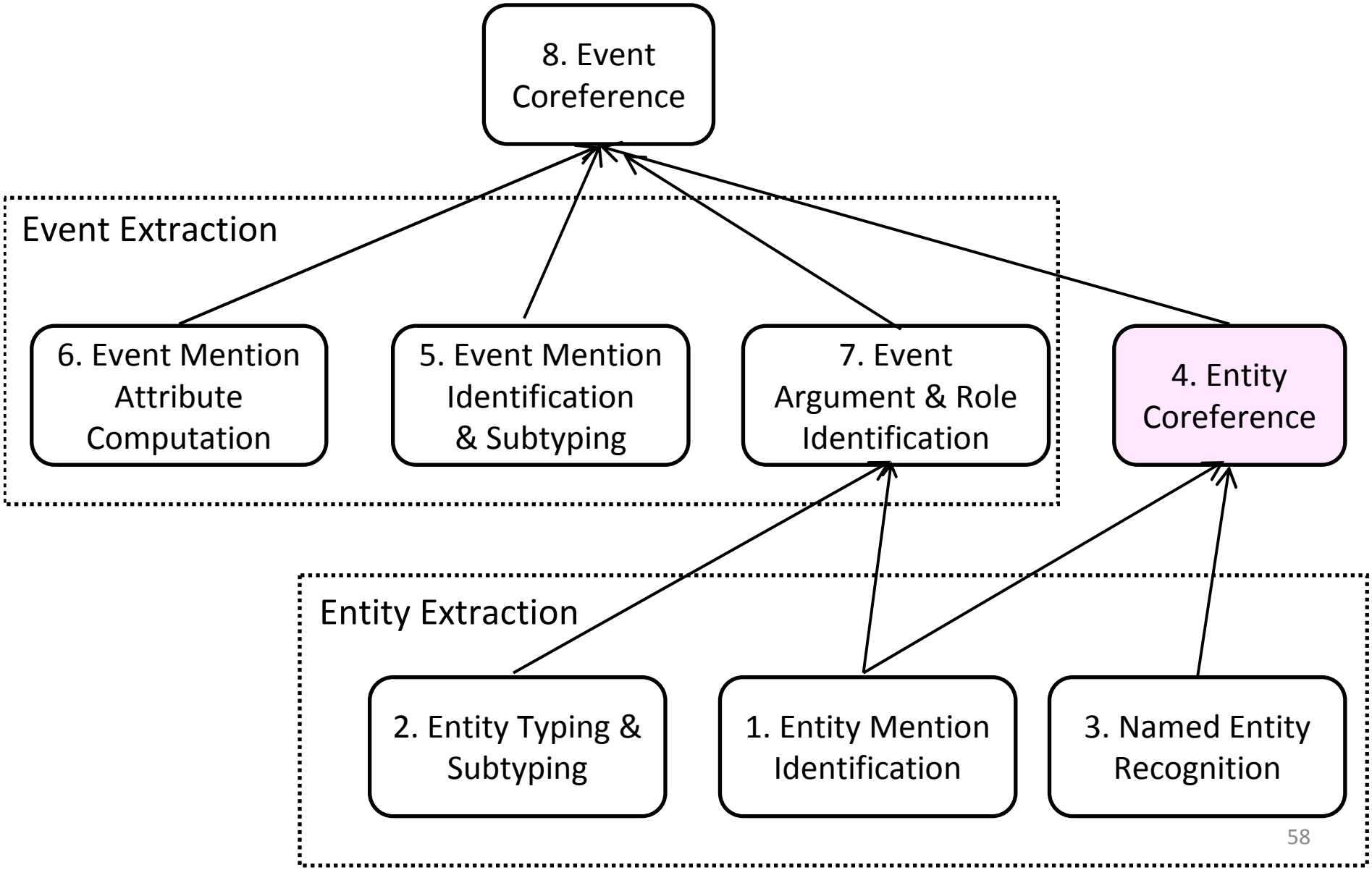
	Argument			Role		
Input	R	P	F	R	P	F
Perfect	68.9	87.1	76.9	61.1	77.2	68.2
Predicted	23.1	36.7	28.3	20.0	31.9	24.6

Event Argument & Role Classification

- How to implement this component?
  - Implemented as part of our Chinese event extraction system (Chen and Ng, 2012)
    - Jointly learn the event arguments and their roles
- **Results** (in terms of R, P, F)
  - Perfect vs. predicted event mention boundary & subtype

	Argument			Role		
Input	R	P	F	R	P	F
Perfect	68.9	87.1	76.9	61.1	77.2	68.2
Predicted	23.1	36.7	28.3	20.0	31.9	24.6

SinoCoreferencer: System Architecture



Entity Coreference Resolution

- Goal
  - Create entity coreference clusters

Entity Coreference Resolution

- Goal
  - Create entity coreference clusters

- Why is this component useful for event coreference?

Entity Coreference Resolution

- Goal
  - Create entity coreference clusters

- Why is this component useful for event coreference?
  - Two event mentions having coreferent arguments are likely to be coreferent

Entity Coreference Resolution

- How to implement this component?
  - Employs our Chinese entity coreference resolver (Chen and Ng, 2012)
    - Hybrid rule-based and learning-based approach
    - Ranked first in the CoNLL-2012 shared task

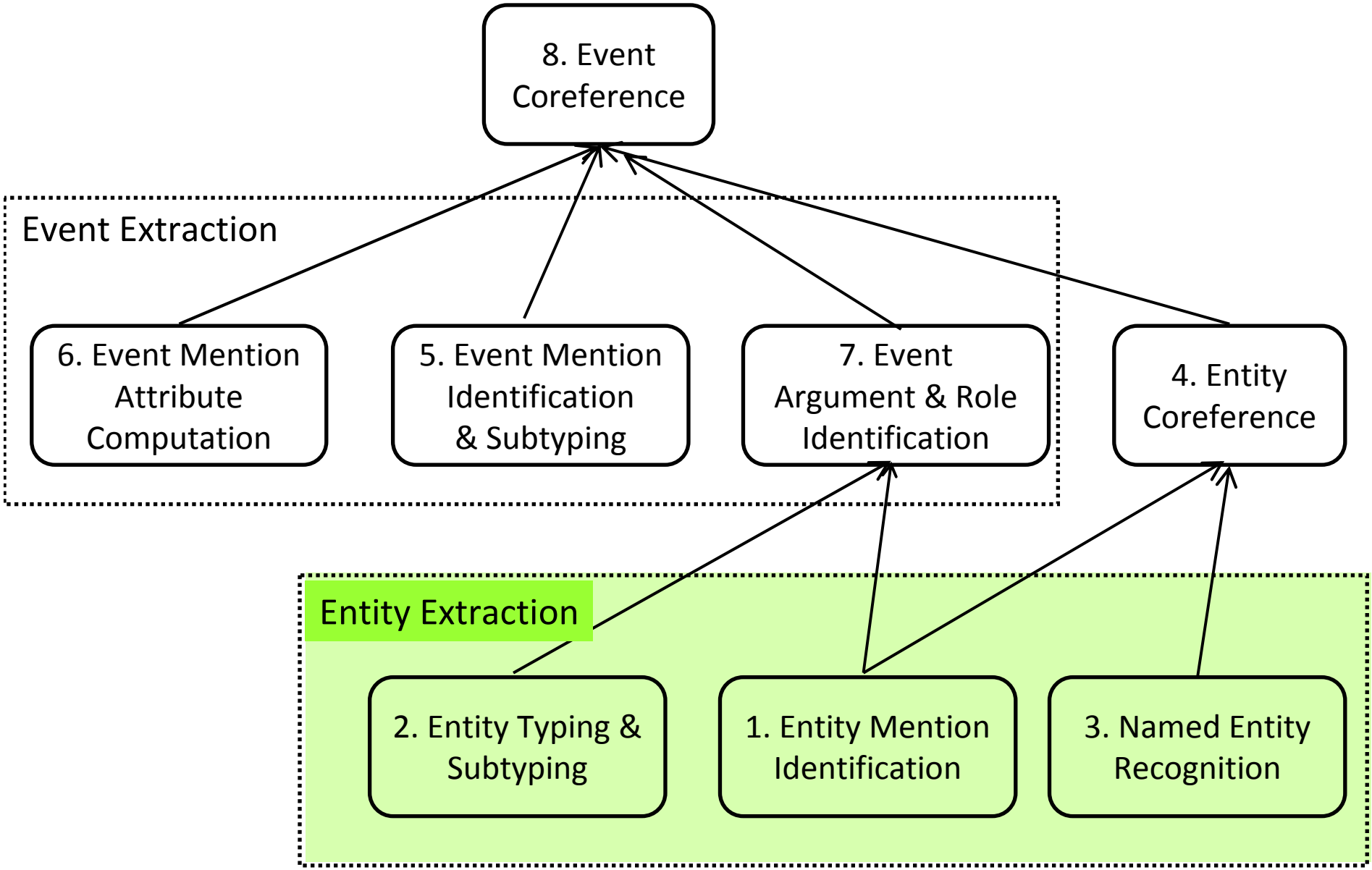
Entity Coreference Resolution

- How to implement this component?
  - Employs our Chinese entity coreference resolver (Chen and Ng, 2012)
    - Hybrid rule-based and learning-based approach
    - Ranked first in the CoNLL-2012 shared task

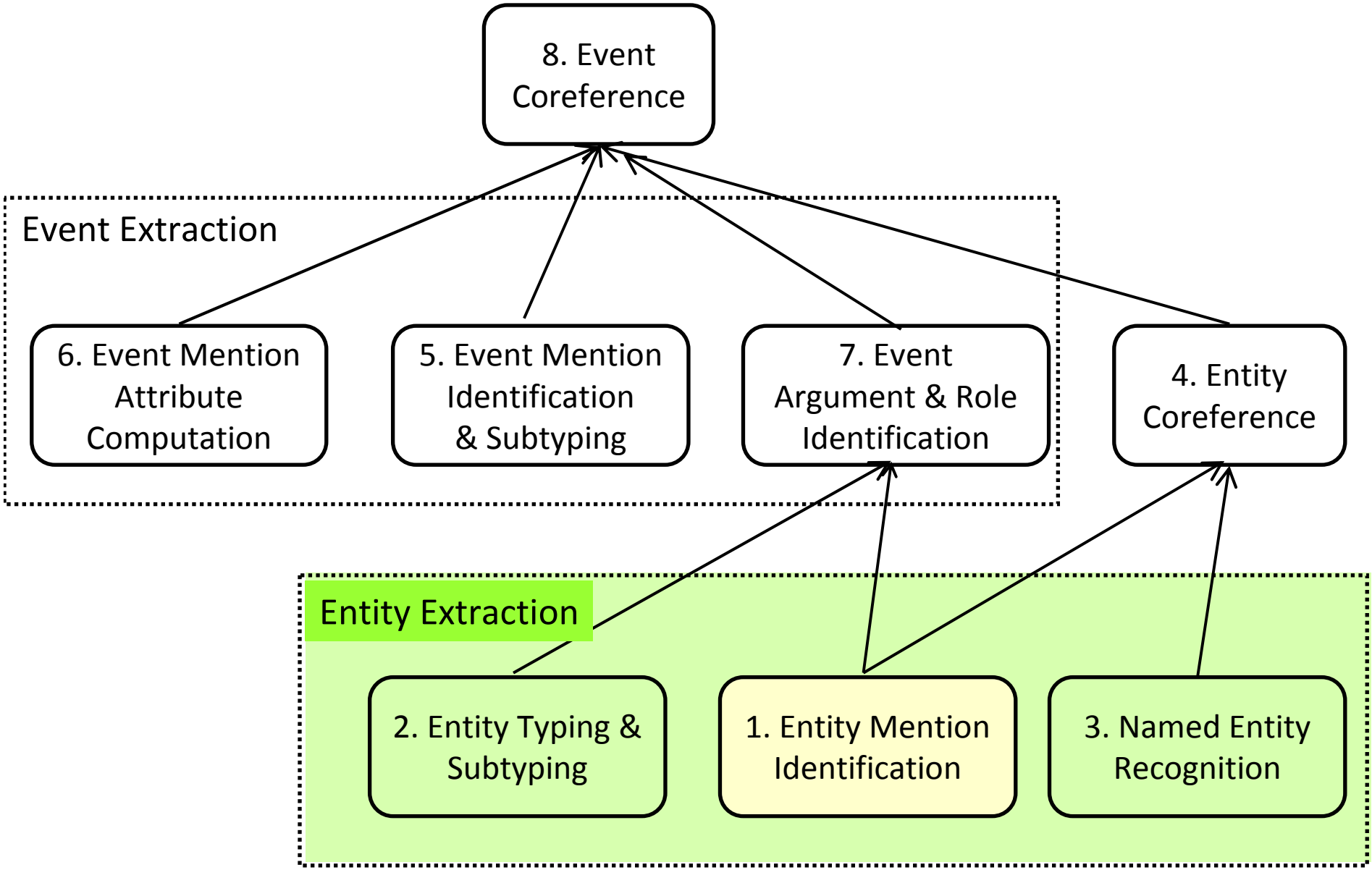
- **Results** (in terms of MUC, B^3, and $CEAF_e$)
  - Perfect vs. predicted entity mentions

		MUC			B^3			$CEAF_e$		AvgF
Input	R	P	F	R	P	F	R	P	F	F
Perfect	71.5	85.8	78.0	67.4	88.0	76.4	69.4	48.8	57.3	70.6
Predicted	61.7	78.0	68.9	63.6	84.6	72.6	57.9	40.3	47.6	63.0

SinoCoreferencer: System Architecture



SinoCoreferencer: System Architecture



Entity Mention Identification

- Goal
  - Provide the entity mentions needed by the downstream components
    - Candidate arguments of event mentions
    - Entity mentions needed for entity coreference

Entity Mention Identification

- Goal
  - Provide the entity mentions needed by the downstream components
    - Candidate arguments of event mentions
    - Entity mentions needed for entity coreference
    - Indirect influence on event coreference

Entity Mention Identification

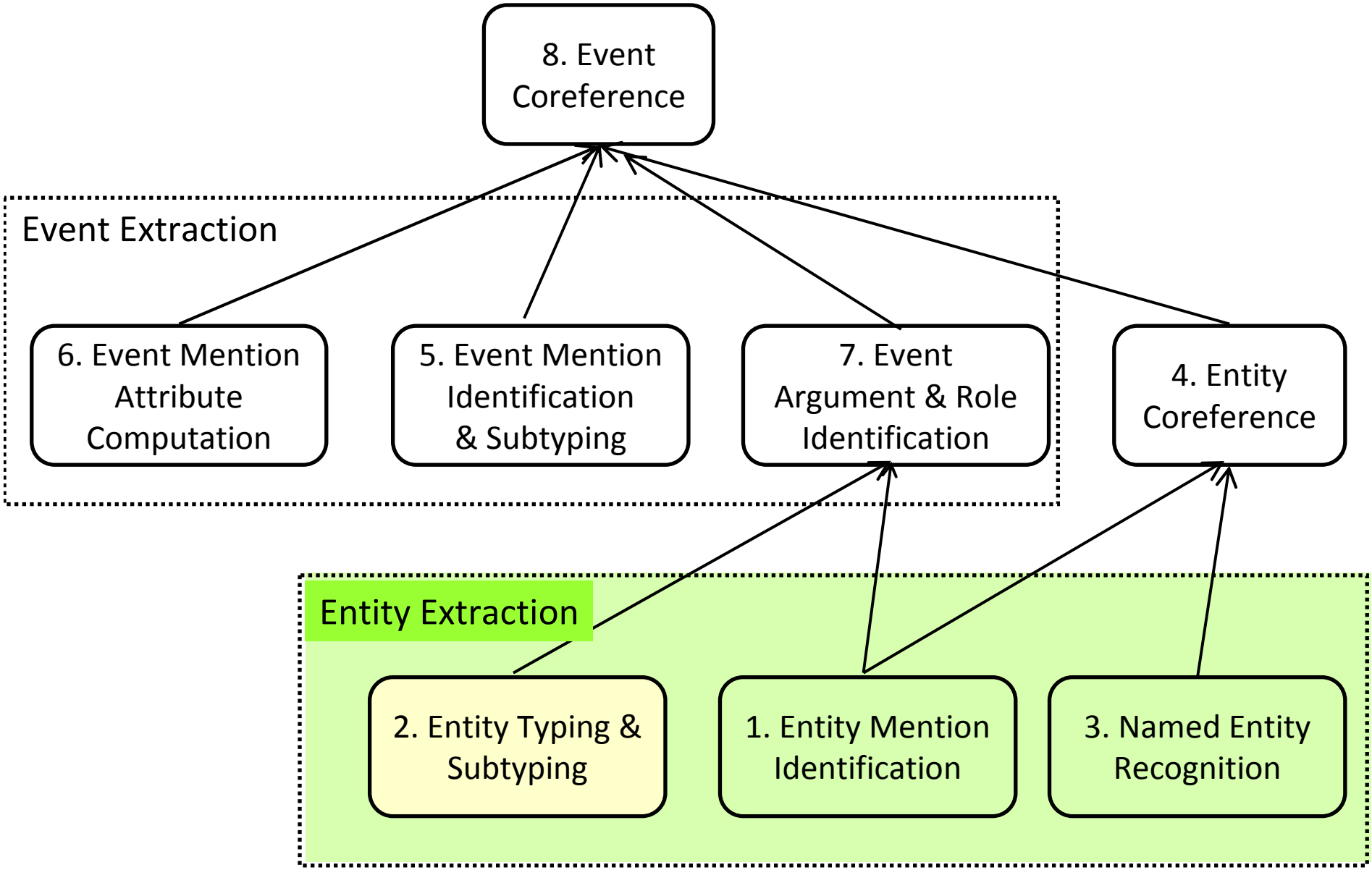
- How to implement this component?
  - We train CRF classifiers to extract entity mentions

Entity Mention Identification

- How to implement this component?
  - We train CRF classifiers to extract entity mentions

- achieves an overall F-score of 84.7% on ACE 2005

SinoCoreferencer: System Architecture



Entity Typing & SubTyping

- Goal
  - Determine the type and subtype of entity mention

Entity Typing & SubTyping

- Goal
  - Determine the type and subtype of entity mention

- Why is this component useful for event coreference?

Entity Typing & SubTyping

- Goal
  - Determine the type and subtype of entity mention

- Why is this component useful for event coreference?
  - Provide features needed to train a classifier for classifying the role of an event argument in event extraction
    - Indirect influence on event coreference

Entity Typing & SubTyping

- How to implement this component?
  - train two SVM classifiers
    - One for classifying types and the other subtypes

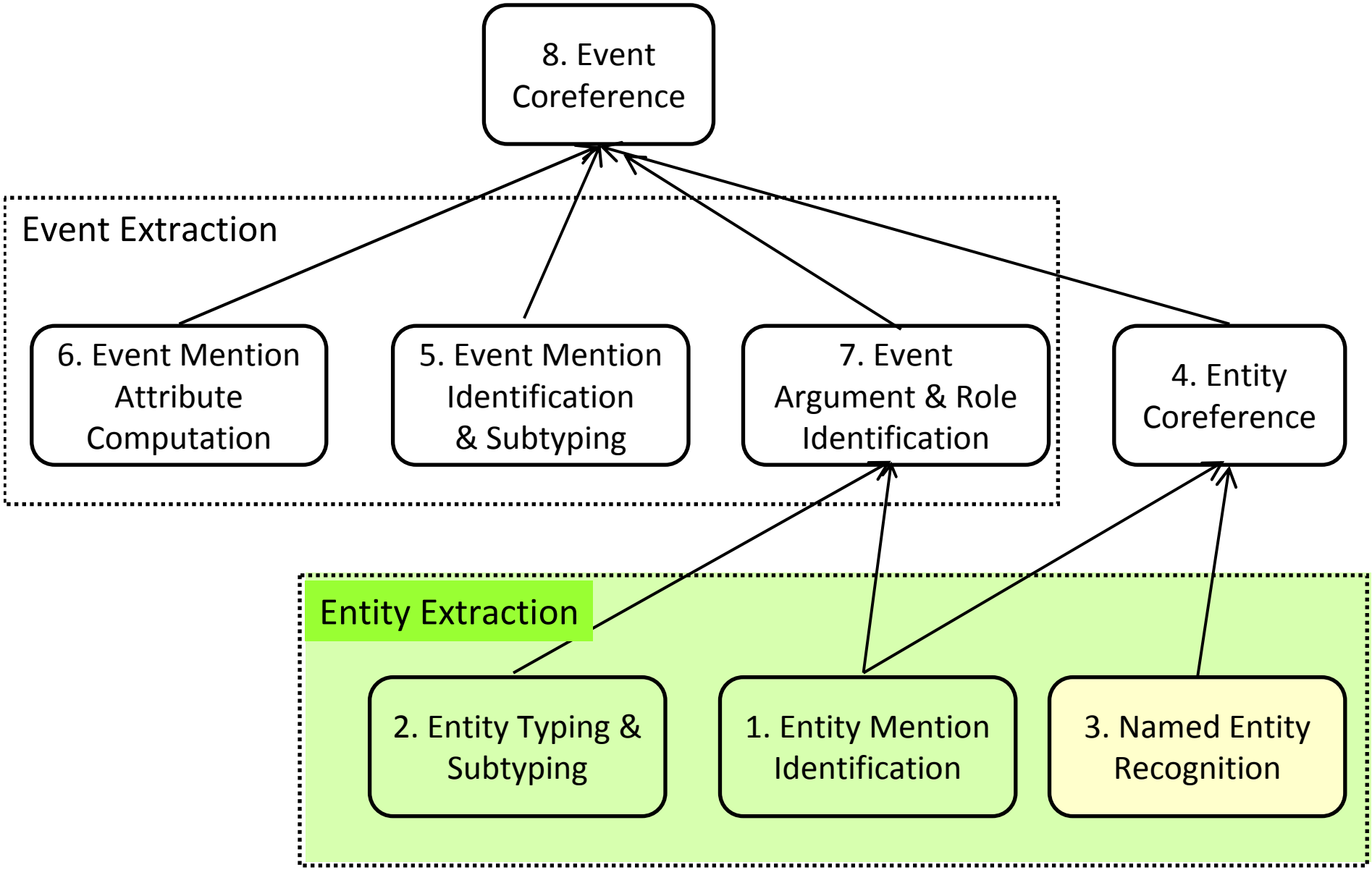
Entity Typing & SubTyping

- How to implement this component?
  - train two SVM classifiers
    - One for classifying types and the other subtypes

- **Results** (in terms of R, P, and F)
  - Perfect vs. predicted entity mention boundaries

	Entity Typing			Entity SubTyping		
Input	R	P	F	R	P	F
Perfect	90.1	90.1	90.1	81.6	81.6	81.6
Predicted	80.5	77.6	79.0	73.1	70.4	71.7

SinoCoreferencer: System Architecture



Named Entity Recognition

- Goal
  - recognize named entities

Named Entity Recognition

- Goal
  - recognize named entities

- Why is this component useful for event coreference?

Named Entity Recognition

- Goal
  - recognize named entities

- Why is this component useful for event coreference?
  - Provide features for entity coreference resolution
    - Indirect influence on event coreference

Named Entity Recognition

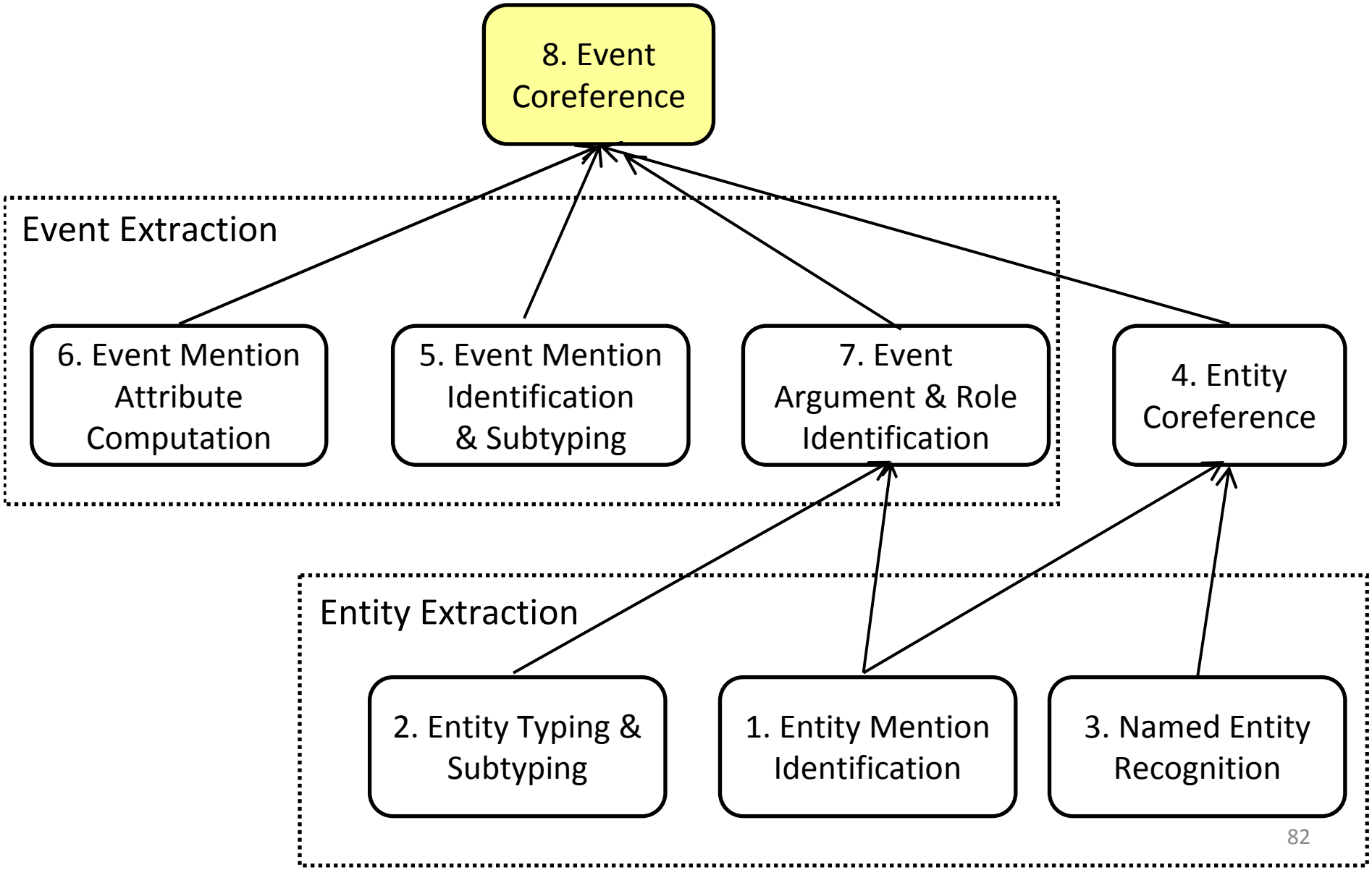
- How to implement this component?
  - recast NER as a sequence labeling task
    - train an NE tagger on 37 NE classes defined in OntoNotes 5.0 with 18 features

Named Entity Recognition

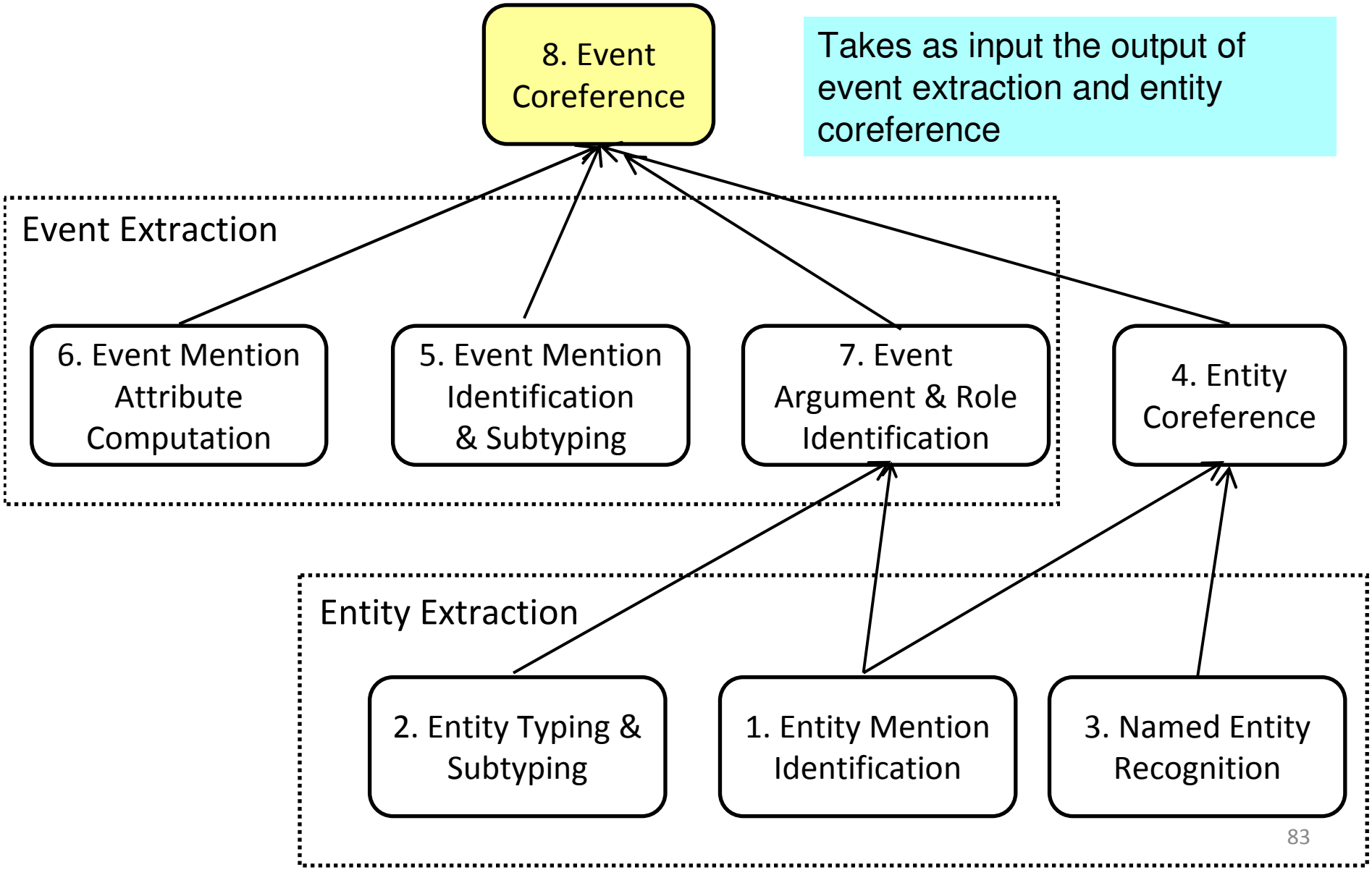
- How to implement this component?
  - recast NER as a sequence labeling task
    - train an NE tagger on 37 NE classes defined in OntoNotes 5.0 with 18 features

- Overall F-score is 66.4%

SinoCoreferencer: System Architecture



SinoCoreferencer: System Architecture



Event Coreference

- Encode the output of the entity coreference subsystem and the event extraction subsystem as features for training an event coreference classifier
  - Pairwise classifier that determines whether two event mentions are coreferent

Event Coreference

- Encode the output of the entity coreference subsystem and the event extraction subsystem as features for training an event coreference classifier
  - Pairwise classifier that determines whether two event mentions are coreferent

- During testing, it selects as the antecedent of each event mention the closest preceding event mention that is classified as coreferent with it

Results of Event Coreference

		MUC			B^3			$CEAF_e$		AvgF
Input	R	P	F	R	P	F	R	P	F	F
Perfect	80.4	70.0	74.8	88.4	79.7	83.8	57.3	66.8	61.7	73.4
Predicted	37.4	36.7	37.1	72.8	71.1	71.9	40.6	41.1	40.8	49.9

Evaluation

- To what extent is the noisy output of each of its upstream components limiting the performance?

Ablation Experiments

- Start with an event coreference resolver that assumes all upstream components are error free
- Replace each oracle component with its predicted (i.e., automatically computed) counterpart one by one

Summary of Ablation Results

- Components whose noise has big impact on event coreference performance:
  - those in event extraction and entity coreference
- Components whose noise has little impact:
  - those in entity extraction

Same as English Event Coreference?

Same as English Event Coreference?

- Chinese event coreference is more challenging in part because the upstream components may have lower accuracies
  - Chinese has **no morphology**, so in Chinese it is hard to correctly classify verb tenses
  - Chinese requires **segmentation**. Segmentation errors affect event mention detection
  - Chinese has **zero pronouns**, which make it harder to find the arguments of events

Summary

- Analyzed an ACE-style Chinese event coreference resolver
- Made our implementation publicly available
  - Facilitate the development of Chinese NLP applications

Future Work

- Examine partial coreference relations (Hovy et al., 2013)
  - **subevent**
    - Subevent relations form a sterotypical sequence of events
      - e.g., bombing → destroyed → wounding
  - **membership**
    - multiple instances of the same event
      - e.g., injury events corresponding to different people

NAACL HLT 2013 and ACL 2014 workshops