An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming

Xiaofeng Yang, Jian Su, Jun Lang
Chew Lim Tan, Ting Liu and Sheng Li

Institute for Infocomm Research
Harbin Institute of Technology
National university of Singapore

June 18, 2008

Outline

Outline

What is Coreference Resolution

Coreference resolution is the process of linking multiple mentions that refer to the same entity.

What is Coreference Resolution

Coreference resolution is the process of linking multiple mentions that refer to the same entity.

Example

[Microsoft Corp.]$_1$ announced [[its]$_2$ new CEO]$_3$ [yesterday]$_4$. [The company]$_5$ said [he]$_6$ will . . .

Three entities:

e1 : *Microsoft Corp. - its - The company*
e2 : *its new CEO - he*
e3 : *yesterday*

Outline

- Introduction
  - Mention-Pair Model
  - Entity-Mention Model
- Entity-mention Model with ILP
  - Motivation
  - Applying ILP to coreference resolution
- Experiments and Results
  - Experimental Setup
  - Results and Discussions
- Conclusions

Mention-Pair Model

The model

- Train a classifier to determine the likelihood two mentions are coreferent or not (Soon et al., 2001; Ng and Cardie, 2002).
- $p(L|m_k, m_j)$

Instance

- $(m\_k, m\_j) : k < j$
- positive if *m_k* and *m_j* are coreferential
- negative if otherwise

Mention-Pair Model

The model

- Train a classifier to determine the likelihood two mentions are coreferent or not (Soon et al., 2001; Ng and Cardie, 2002).
- $p(L|m_k, m_j)$

Instance

- $(m\_k, m\_j) : k < j$
- positive if m_k and m_j are coreferential
- negative if otherwise

Resolution

- Testing instances: For each m_j, $<m_1, m_j>$, ...,$<m_{j-1}, m_j>$
- m_j links with the mention with the highest classification confidence (>0.5)

Feature set for Mention-Pair Model

Features describing an active mention, m_j	
defNP_mj	1 if m_j is a definite description; else 0
indefNP_mj	1 if m_j is an indefinite NP; else 0
nameNP_mj	1 if m_j is a named-entity; else 0
pron_mj	1 if m_j is a pronoun; else 0
bareNP_mj	1 if m_j is a bare NP (i.e., NP without determiners) ; else 0
Features describing a previous mention, m_k	
defNP_mk	1 if m_k is a definite description; else 0
indefNP_mk	1 if m_k is an indefinite NP; else 0
nameNP_mk	1 if m_k is a named-entity; else 0
pron_mk	1 if m_k is a pronoun; else 0
bareNP_mk	1 if m_k is a bare NP; else 0
subject_mk	1 if m_k is an NP in a subject position; else 0
Features describing the relationships between m_k and m_j	
sentDist	sentence distance between two mentions
numAgree	1 if two mentions match in the number agreement; else 0
genderAgree	1 if two mentions match in the gender agreement; else 0
parallelStruct	1 if two mentions have an identical collocation pattern; else 0
semAgree	1 if two mentions have the same semantic category; else 0
nameAlias	1 if two mentions are an alias of the other; else 0
apposition	1 if two mentions are in an appositive structure; else 0
predicative	1 if two mentions are in a predicative structure; else 0
strMatch_Head	1 if two mentions have the same head string; else 0
strMatch_Full	1 if two mentions contain the same strings, excluding the determiners; else 0
strMatch_Contain	1 if the string of m_j is fully contained in that of m_k; else 0

Shortcomings of Mention-pair Model

Could not capture enough information

- Example: "*Mr. Powell*", "*he*", and "*Powell*"
- ("*he*", "*Powell*"): "*he*" discloses nothing but gender.
- Consider whole entity instead of one mention: "*he*" refers to a male person named "*Powell*"

Shortcomings of Mention-pair Model

Could not capture enough information

- Example: "*Mr. Powell*", "*he*", and "*Powell*"
- ("*he*", "*Powell*"): "*he*" discloses nothing but gender.
- Consider whole entity instead of one mention: "*he*" refers to a male person named "*Powell*"

Possibly leading to contradiction while testing

- Example: "*Mr. Powell*", "*Powell*, and "*she*"
- ("*Powell*", "*she*"): by proximity
- ("*Mr. Powell*", "*Powell*"): by same head

Outline

Entity-Mention Model

The model

- Train a classifier to determine the likelihood that a mention is referent of a partially found entity. (Luo et al., 2004; Yang et al., 2004)
- $p(L|e\_i, m_j)$

Instance

- (e_{i_j}, m_j)
  - m_j: active mention
  - $e_{i\_j} = \{m_k | m_k \in e_i, k < j\}$: partial entity found before m_j

Entity-Mention Model

The model

- Train a classifier to determine the likelihood that a mention is referent of a partially found entity. (Luo et al., 2004; Yang et al., 2004)
- $p(L|e\_i, m_j)$

Instance

- (e_{i_j}, m_j)
  - m_j: active mention
  - $e_{i\_j} = \{m_k | m_k \in e_i, k < j\}$: partial entity found before m_j

Resolution

- Testing instances: For each m_j, $<e_{1\_j}, m_j>$, ...,$<e_{k\_j}, m_j>$
- incrementally link m_j with the entity with the highest classification confidence (>0.5)

Problem: How to represent the knowledge of an entity

Describing mentions in an entity

- Number is not fixed.
- Cannot be represented in a single attribute-value vector

Problem: How to represent the knowledge of an entity

Describing mentions in an entity

- Number is not fixed.
- Cannot be represented in a single attribute-value vector

Solution of first-order features(Luo et al., 2004; Culotta et al., 2007))

- Similar to mention-pair model's, but calculate at an entity level
  - Lexical and grammatical: Any mention
    - e.g.: *namealias*=1 if at least one mention is *namealias* with m_j
  - Distance: minimum distance
- Ad-hoc way, can not capture the detail information of each mention.

Outline

Outline

Knowledge Representing for Entity-mention Model

Knowledge Representing

- Relational knowledge for active mention(m_j), an entity(e_i), and mentions in the entity($\{m_k \in e_i\}$)

Knowledge Representing for Entity-mention Model

Knowledge Representing

- Relational knowledge for active mention(m_j), an entity(e_i), and mentions in the entity($\{m_k \in e_i\}$)

Why inductive logic programming?

- Normal machine learning algorithms work on attribute-value vectors.
- To learn from relational knowledge, we need an algorithm that can express first-order logic.
- Inductive Logic Programming (ILP) is a learning algorithm capable of inferring logic programs.

Inductive Logic Programming

What is ILP?

- ILP uses logic programming as a uniform representation for examples, background knowledge and hypotheses.
- Given $E = E^{+} \cup E^{-}$ and K, induce h such that $K \wedge h \models E^{+}$ and $K \wedge h \not\models E^{-}$.

Inductive Logic Programming

What is ILP?

- ILP uses logic programming as a uniform representation for examples, background knowledge and hypotheses.
- Given $E = E^{+} \cup E^{-}$ and K, induce h such that $K \wedge h \models E^{+}$ and $K \wedge h \not\models E^{-}$.

ILP for NLP

- Parsing(Mooney, 1997), POS Tagging(Cussens, 1996)
- Lexicon Construction(Claveau, 2003), Word Sense Disambiguation(Specia, 2007)

Inductive Logic Programming

What is ILP?

- ILP uses logic programming as a uniform representation for examples, background knowledge and hypotheses.
- Given $E = E^{+} \cup E^{-}$ and K, induce h such that $K \wedge h \models E^{+}$ and $K \wedge h \not\models E^{-}$.

ILP for NLP

- Parsing(Mooney, 1997), POS Tagging(Cussens, 1996)
- Lexicon Construction(Claveau, 2003), Word Sense Disambiguation(Specia, 2007)

ALEPH

- ILP implementation by Srinivasan (2000)

Outline

ILP Modelling for Entity-Mention Model

Constants

- $m_j(1 \leqslant j \leqslant n)$: the jth mention in a document
- e_{i_j}: the partial entity before m_j

ILP Modelling for Entity-Mention Model

Constants

- $m_j(1 \leqslant j \leqslant n)$: the jth mention in a document
- e_{i_j}: the partial entity before m_j

Predicates for instances

- $link(e_{i\_j}, m_j)$, put in E^+ if m_j belongs to e_{i_j}, or E^- if otherwise

ILP Modelling for Entity-Mention Model

Constants

- m_j(1⩽j⩽n): the jth mention in a document
- e_{i_j}: the partial entity before m_j

Predicates for instances

- $link(e_{i\_j}, m_j)$, put in E^+ if m_j belongs to e_{i_j}, or E^- if otherwise

Predicates for background knowledge

- The information related to e_{i_j} and m_j
- The belonging relations between e_{i_j} and its mentions
- The information related to m_j and each mention m_k in e_{i_j}

Information related to e_{i_j} and m_j

Microsoft Corp.$^1_1$ *announced* *its*$^1_2$ *new CEO yesterday.* *The company*$^1_5$ *said* *he*$^2_6$ *will . . .*

Properties of m_j

- Form: $f(m, v)$
- Example:
  - $defNP(m_6, 0)$: m_6 is not a definite np
  - $pron(m_6, 1)$: m_6 is a pronoun

Information related to e_{i_j} and m_j

Microsoft Corp. $^1_1$ *announced its* $^1_2$ *new CEO yesterday. The company* $^1_5$ *said he* $^2_6$ *will ...*

Properties of m_j

- Form: $f(m, v)$
- Example:
  - $defNP(m_6, 0)$: m_6 is not a definite np
  - $pron(m_6, 1)$: m_6 is a pronoun

Relationships between e_{i_j} and m_j

- Form: $f(e, m, v)$
- Example:
  - $entNumAgree(e_{1\_6}, m_6, 1)$: all mentions in e_{1_6} agree in number with m_6
  - $entGenderAgree(e_{1\_6}, m_6, 0)$: not all mentions in e_{1_6} agree in gender with m_6

The belonging relations between e_{i_j} and its mentions

Microsoft Corp.$^1_1$ *announced* *its*$^1_2$ *new CEO yesterday.* *The company*$^1_5$ *said* *he*$^2_6$ *will . . .*

Form: *has_mention*(e, m)
Example:

- $has\_mention(e_{1\_6}, m_1)$
- $has\_mention(e_{1\_6}, m_2)$
- $has\_mention(e_{1\_6}, m_5)$

The information related to m_j and each mention m_k in e_{i_j}

Microsoft Corp.$^1_1$ *announced* *its*$^1_2$ *new CEO yesterday.* *The company*$^1_5$ *said* *he*$^2_6$ *will . . .*

Form: $f(m_k, m_j, v)$

Example

- $nameAlias(m_1, m_6, 0)$
- $sentDist(m_1, m_6, 1)$

Running ILP with Predicates

Key part of three kinds of predicates

- entity-mention ← *has_mention* → mention-pair
- Global entity information and Local individual mention information

Running ILP with Predicates

Key part of three kinds of predicates

- entity-mention ← *has_mention* → mention-pair
- Global entity information and Local individual mention information

Rules learned with ALEPH

link(A,B) :-
has_mention(A,C), numAgree(B,C,1),
strMatch_Head(B,C,1), bareNP(C,1).

Running ILP with Predicates

Key part of three kinds of predicates

- entity-mention $\leftarrow$ *has_mention* $\rightarrow$ mention-pair
- Global entity information and Local individual mention information

Rules learned with ALEPH

link(A,B) :-
has_mention(A,C), numAgree(B,C,1),
strMatch_Head(B,C,1), bareNP(C,1).

read as

$$\forall A, B(\exists C, has\_mention(A, C) \wedge numAgree(B, C, 1) \wedge strMatch\_Head(B, C, 1) \wedge bareNP(C, 1) \rightarrow link(A, B))$$

Testing with ILP Rules

Link or not from m_j to e_i

- Using ILP rules for $link(e_i,m_j)$
- Each rule has an accuracy for producing the training instances.
- Confidence is the maximum score of applicable rules
- Link(e_i,m_j) holds if confidence is above 0.5

Outline

Outline

Experimental Setup

Corpus: ACE-2 V1.0, (NIST, 2003)

	Train		Test	
	#entity	#mention	#entity	#mention
NWire	1678	9861	411	2304
NPaper	1528	10277	365	2290
BNews	1695	8986	468	2493

Table: statistics of entities (length $>$ 1) and contained mentions

Experimental Setup

Corpus: ACE-2 V1.0, (NIST, 2003)

	Train		Test	
	#entity	#mention	#entity	#mention
NWire	1678	9861	411	2304
NPaper	1528	10277	365	2290
BNews	1695	8986	468	2493

Table: statistics of entities (length $>$ 1) and contained mentions

Evaluation

Vilain, 1995

Experimental Setup

Corpus: ACE-2 V1.0, (NIST, 2003)

	Train		Test	
	#entity	#mention	#entity	#mention
NWire	1678	9861	411	2304
NPaper	1528	10277	365	2290
BNews	1695	8986	468	2493

Table: statistics of entities (length $>$ 1) and contained mentions

Evaluation

Vilain, 1995

ALEPH setting

Minimum accuracy of generated rules: 0.5
Maximum number of predicates in a rule: 10

Outline

Experimental results

	NWire			NPaper			BNews		
	R	P	F	R	P	F	R	P	F
C4.5									
- Mention-Pair	68.2	54.3	60.4	67.3	50.8	57.9	66.5	59.5	62.9
- Entity-Mention (first-order features)	66.8	55.0	60.3	64.2	53.4	58.3	64.6	60.6	62.5
ILP									
- Mention-Pair	66.1	54.8	59.5	65.6	54.8	59.7	63.5	60.8	62.1
- Entity-Mention	65.0	58.9	**61.8**	63.4	57.1	**60.1**	61.7	65.4	**63.5**

Table: Results of different systems for coreference resolution

Generated rules by ILP of Entity-mention model

link(A,B) :-
bareNP(B,0), has_mention(A,C), appositive(C,1).

link(A,B) :-
has_mention(A,C), numAgree(B,C,1), strMatch_Head(B,C,1), bareNP(C,1).

link(A,B) :-
nameNP(B,0), has_mention(A,C), predicative(C,1).

link(A,B) :-
has_mention(A,C), strMatch_Contain(B,C,1), strMatch_Head(B,C,1), bareNP(C,0).

link(A,B) :-
nameNP(B,0), has_mention(A,C), nameAlias(C,1), bareNP(C,0).

link(A,B) :-
pron(B,1), has_mention(A,C), nameNP(C,1), has_mention(A,D), indefNP(D,1), subject(D, 1).
...

Figure: Examples of rules produced by ILP (entity-mention model)

Outline

Conclusions

Contributions

- An expressive entity-mention model for coreference resolution by using ILP.
- Perform better than mention-pair model
- Also better than entity-mention model based on heuristical first-order features.

Conclusions

Contributions

- An expressive entity-mention model for coreference resolution by using ILP.
- Perform better than mention-pair model
- Also better than entity-mention model based on heuristical first-order features.

Future work

- Currently, incremental and greedy link
- Global optimization, e.g., (Denis and Baldridge, 2007)
- More complicated clustering e,g, (Luo et al., 2004)

Thanks!