

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

Xiaofeng Yang¹ Jian Su¹ Jun Lang²
Chew Lim Tan³ Ting Liu² Sheng Li²

¹Institute for Infocomm Research
{xiaofengy,sujian}@i2r.a-star.edu.sg

²Harbin Institute of Technology
{bill_lang,tliu}@ir.hit.edu.cn
lisheng@hit.edu.cn

³National University of Singapore,
tancl@comp.nus.edu.sg

Abstract

The traditional mention-pair model for coreference resolution cannot capture information beyond mention pairs for both learning and testing. To deal with this problem, we present an expressive entity-mention model that performs coreference resolution at an entity level. The model adopts the Inductive Logic Programming (ILP) algorithm, which provides a relational way to organize different knowledge of entities and mentions. The solution can explicitly express relations between an entity and the contained mentions, and automatically learn first-order rules important for coreference decision. The evaluation on the ACE data set shows that the ILP based entity-mention model is effective for the coreference resolution task.

1 Introduction

Coreference resolution is the process of linking multiple mentions that refer to the same entity. Most of previous work adopts the mention-pair model, which recasts coreference resolution to a binary classification problem of determining whether or not two mentions in a document are co-referring (e.g. Aone and Bennett (1995); McCarthy and Lehnert (1995); Soon et al. (2001); Ng and Cardie (2002)). Although having achieved reasonable success, the mention-pair model has a limitation that information beyond mention pairs is ignored for training and testing. As an individual mention usually lacks adequate descriptive information of the referred entity, it is often difficult to judge whether or not two men-

tions are talking about the same entity simply from the pair alone.

An alternative learning model that can overcome this problem performs coreference resolution based on entity-mention pairs (Luo et al., 2004; Yang et al., 2004b). Compared with the traditional mention-pair counterpart, the entity-mention model aims to make coreference decision at an entity level. Classification is done to determine whether a mention is a referent of a partially found entity. A mention to be resolved (called *active mention* henceforth) is linked to an appropriate entity chain (if any), based on classification results.

One problem that arises with the entity-mention model is how to represent the knowledge related to an entity. In a document, an entity may have more than one mention. It is impractical to enumerate all the mentions in an entity and record their information in a single feature vector, as it would make the feature space too large. Even worse, the number of mentions in an entity is not fixed, which would result in variant-length feature vectors and make trouble for normal machine learning algorithms. A solution seen in previous work (Luo et al., 2004; Culotta et al., 2007) is to design a set of first-order features summarizing the information of the mentions in an entity, for example, “whether the entity has any mention that is a name alias of the active mention?” or “whether most of the mentions in the entity have the same head word as the active mention?” These features, nevertheless, are designed in an ad-hoc manner and lack the capability of describing each individual mention in an entity.

In this paper, we present a more expressive entity-

mention model for coreference resolution. The model employs Inductive Logic Programming (ILP) to represent the relational knowledge of an active mention, an entity, and the mentions in the entity. On top of this, a set of first-order rules is automatically learned, which can capture the information of each individual mention in an entity, as well as the global information of the entity, to make coreference decision. Hence, our model has a more powerful representation capability than the traditional mention-pair or entity-mention model. And our experimental results on the ACE data set shows the model is effective for coreference resolution.

2 Related Work

There are plenty of learning-based coreference resolution systems that employ the mention-pair model. A typical one of them is presented by Soon et al. (2001). In the system, a training or testing instance is formed for two mentions in question, with a feature vector describing their properties and relationships. At a testing time, an active mention is checked against all its preceding mentions, and is linked with the closest one that is classified as positive. The work is further enhanced by Ng and Cardie (2002) by expanding the feature set and adopting a “best-first” linking strategy.

Recent years have seen some work on the entity-mention model. Luo et al. (2004) propose a system that performs coreference resolution by doing search in a large space of entities. They train a classifier that can determine the likelihood that an active mention should belong to an entity. The entity-level features are calculated with an “Any-X” strategy: an entity-mention pair would be assigned a feature X, if any mention in the entity has the feature X with the active mention.

Culotta et al. (2007) present a system which uses an online learning approach to train a classifier to judge whether two entities are coreferential or not. The features describing the relationships between two entities are obtained based on the information of every possible pair of mentions from the two entities. Different from (Luo et al., 2004), the entity-level features are computed using a “Most-X” strategy, that is, two given entities would have a feature X, if most of the mention pairs from the two entities

have the feature X.

Yang et al. (2004b) suggest an entity-based coreference resolution system. The model adopted in the system is similar to the mention-pair model, except that the entity information (e.g., the global number/gender agreement) is considered as additional features of a mention in the entity.

McCallum and Wellner (2003) propose several graphical models for coreference analysis. These models aim to overcome the limitation that pairwise coreference decisions are made independently of each other. The simplest model conditions coreference on mention pairs, but enforces dependency by calculating the distance of a node to a partition (i.e., the probability that an active mention belongs to an entity) based on the sum of its distances to all the nodes in the partition (i.e., the sum of the probability of the active mention co-referring with the mentions in the entity).

Inductive Logic Programming (ILP) has been applied to some natural language processing tasks, including parsing (Mooney, 1997), POS disambiguation (Cussens, 1996), lexicon construction (Claveau et al., 2003), WSD (Specia et al., 2007), and so on. However, to our knowledge, our work is the first effort to adopt this technique for the coreference resolution task.

3 Modelling Coreference Resolution

Suppose we have a document containing n mentions $\{m_j : 1 < j < n\}$, in which m_j is the j th mention occurring in the document. Let e_i be the i th entity in the document. We define

$$P(L|e_i, m_j), \quad (1)$$

the probability that a mention belongs to an entity. Here the random variable L takes a binary value and is 1 if m_j is a mention of e_i .

By assuming that mentions occurring after m_j have no influence on the decision of linking m_j to an entity, we can approximate (1) as:

$$\begin{aligned} & P(L|e_i, m_j) \\ \propto & P(L|\{m_k \in e_i, 1 \leq k \leq j-1\}, m_j) \quad (2) \\ \propto & \max_{m_k \in e_i, 1 \leq k \leq j-1} P(L|m_k, m_j) \quad (3) \end{aligned}$$

(3) further assumes that an entity-mention score can be computed by using the maximum mention-

[*Microsoft Corp.*]₁¹ announced [[*its*]₂¹ new CEO]₃²
[*yesterday*]₄³. [*The company*]₅¹ said [*he*]₆² will ...

Table 1: A sample text

pair score. Both (2) and (1) can be approximated with a machine learning method, leading to the traditional mention-pair model and the entity-mention model for coreference resolution, respectively.

The two models will be described in the next subsections, with the sample text in Table 1 used for demonstration. In the table, a mention m is highlighted as [m] _{mid} ^{eid} , where mid and eid are the IDs for the mention and the entity to which it belongs, respectively. Three entity chains can be found in the text, that is,

e1 : *Microsoft Corp.* - *its* - *The company*

e2 : *its new CEO* - *he*

e3 : *yesterday*

3.1 Mention-Pair Model

As a baseline, we first describe a learning framework with the mention-pair model as adopted in the work by Soon et al. (2001) and Ng and Cardie (2002).

In the learning framework, a training or testing instance has the form of $i\{m_k, m_j\}$, in which m_j is an active mention and m_k is a preceding mention. An instance is associated with a vector of features, which is used to describe the properties of the two mentions as well as their relationships. Table 2 summarizes the features used in our study.

For training, given each encountered anaphoric mention m_j in a document, one single positive training instance is created for m_j and its closest antecedent. And a group of negative training instances is created for every intervening mentions between m_j and the antecedent. Consider the example text in Table 1, for the pronoun “*he*”, three instances are generated: $i(\text{“The company”}, \text{“he”})$, $i(\text{“yesterday”}, \text{“he”})$, and $i(\text{“its new CEO”}, \text{“he”})$. Among them, the first two are labelled as negative while the last one is labelled as positive.

Based on the training instances, a binary classifier can be generated using any discriminative learning algorithm. During resolution, an input document is processed from the first mention to the last. For each

encountered mention m_j , a test instance is formed for each preceding mention, m_k . This instance is presented to the classifier to determine the coreference relationship. m_j is linked with the mention that is classified as positive (if any) with the highest confidence value.

3.2 Entity-Mention Model

The mention-based solution has a limitation that information beyond a mention pair cannot be captured. As an individual mention usually lacks complete description about the referred entity, the coreference relationship between two mentions may be not clear, which would affect classifier learning. Consider a document with three coreferential mentions “*Mr. Powell*”, “*he*”, and “*Powell*”, appearing in that order. The positive training instance $i(\text{“he”}, \text{“Powell”})$ is not informative, as the pronoun “*he*” itself discloses nothing but the gender. However, if the whole entity is considered instead of only one mention, we can know that “*he*” refers to a male person named “*Powell*”. And consequently, the coreference relationships between the mentions would become more obvious.

The mention-pair model would also cause errors at a testing time. Suppose we have three mentions “*Mr. Powell*”, “*Powell*”, and “*she*” in a document. The model tends to link “*she*” with “*Powell*” because of their proximity. This error can be avoided, if we know “*Powell*” belongs to the entity starting with “*Mr. Powell*”, and therefore refers to a male person and cannot co-refer with “*she*”.

The entity-mention model based on Eq. (2) performs coreference resolution at an entity-level. For simplicity, the framework considered for the entity-mention model adopts similar training and testing procedures as for the mention-pair model. Specifically, a training or testing instance has the form of $i\{e_i, m_j\}$, in which m_j is an active mention and e_i is a partial entity found before m_j . During training, given each anaphoric mention m_j , one single positive training instance is created for the entity to which m_j belongs. And a group of negative training instances is created for every partial entity whose last mention occurs between m_j and the closest antecedent of m_j .

See the sample in Table 1 again. For the pronoun “*he*”, the following three instances are generated for

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

Table 2: Feature set for coreference resolution

entity e_1 , e_3 and e_2 :

$i(\{\text{"Microsoft Corp."}, \text{"its"}, \text{"The company"}\}, \text{"he"}),$

$i(\{\text{"yesterday"}\}, \text{"he"}),$

$i(\{\text{"its new CEO"}\}, \text{"he"}).$

Among them, the first two are labelled as negative, while the last one is positive.

The resolution is done using a greedy clustering strategy. Given a test document, the mentions are processed one by one. For each encountered mention m_j , a test instance is formed for each partial entity found so far, e_i . This instance is presented to the classifier. m_j is appended to the entity that is classified as positive (if any) with the highest confidence value. If no positive entity exists, the active mention is deemed as non-anaphoric and forms a new entity. The process continues until the last mention of the document is reached.

One potential problem with the entity-mention model is how to represent the entity-level knowledge. As an entity may contain more than one candidate and the number is not fixed, it is impractical to enumerate all the mentions in an entity and put their properties into a single feature vector. As a baseline, we follow the solution proposed in (Luo et al., 2004) to design a set of first-order features. The features are similar to those for the mention-pair model as shown in Table 2, but their values are calculated at an entity level. Specifically, the lexical and grammatical features are computed by testing any mention¹ in the entity against the active mention, for ex-

ample, the feature *nameAlias* is assigned value 1 if at least one mention in the entity is a name alias of the active mention. The distance feature (i.e., *sentDist*) is the minimum distance between the mentions in the entity and the active mention.

The above entity-level features are designed in an ad-hoc way. They cannot capture the detailed information of each individual mention in an entity. In the next section, we will present a more expressive entity-mention model by using ILP.

4 Entity-mention Model with ILP

4.1 Motivation

The entity-mention model based on Eq. (2) requires relational knowledge that involves information of an active mention (m_j), an entity (e_i), and the mentions in the entity ($\{m_k \in e_i\}$). However, normal machine learning algorithms work on attribute-value vectors, which only allows the representation of atomic proposition. To learn from relational knowledge, we need an algorithm that can express first-order logic. This requirement motivates our use of Inductive Logic Programming (ILP), a learning algorithm capable of inferring logic programs. The relational nature of ILP makes it possible to explicitly represent relations between an entity and its mentions, and thus provides a powerful expressiveness for the coreference resolution task.

erence relationship with antecedents in a local discourse. Hence, if an active mention is a pronoun, we only consider the mentions in its previous two sentences for feature computation.

¹Linguistically, pronouns usually have the most direct coref-

ILP uses logic programming as a uniform representation for examples, background knowledge and hypotheses. Given a set of positive and negative example $E = E^+ \cup E^-$, and a set of background knowledge K of the domain, ILP tries to induce a set of hypotheses h that covers most of E^+ with no E^- , i.e., $K \wedge h \models E^+$ and $K \wedge h \not\models E^-$.

In our study, we choose ALEPH², an ILP implementation by Srinivasan (2000) that has been proven well suited to deal with a large amount of data in multiple domains. For its routine use, ALEPH follows a simple procedure to induce rules. It first selects an example and builds the most specific clause that entertains the example. Next, it tries to search for a clause more general than the bottom one. The best clause is added to the current theory and all the examples made redundant are removed. The procedure repeats until all examples are processed.

4.2 Apply ILP to coreference resolution

Given a document, we encode a mention or a partial entity with a unique constant. Specifically, m_j represents the j th mention (e.g., m_6 for the pronoun “he”). e_{i-j} represents the partial entity i before the j th mention. For example, e_{1-6} denotes the part of e_1 before m_6 , i.e., {“Microsoft Corp.”, “its”, “the company”}, while e_{1-5} denotes the part of e_1 before m_5 (“The company”), i.e., {“Microsoft Corp.”, “its”}.

Training instances are created as described in Section 3.2 for the entity-mention model. Each instance is recorded with a predicate $link(e_{i-j}, m_j)$, where m_j is an active mention and e_{i-j} is a partial entity. For example, the three training instances formed by the pronoun “he” are represented as follows:

$link(e_{1-6}, m_6)$.

$link(e_{3-6}, m_6)$.

$link(e_{2-6}, m_6)$.

The first two predicates are put into E^- , while the last one is put to E^+ .

The background knowledge for an instance $link(e_{i-j}, m_j)$ is also represented with predicates, which are divided into the following types:

1. Predicates describing the information related to e_{i-j} and m_j . The properties of m_j are pre-

²http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph_toc.html

sented with predicates like $f(m, v)$, where f corresponds to a feature in the first part of Table 2 (removing the suffix $_mj$), and v is its value. For example, the pronoun “he” can be described by the following predicates:

$defNP(m_6, 0)$. $indefNP(m_6, 0)$.

$nameNP(m_6, 0)$. $pron(m_6, 1)$.

$bareNP(m_6, 0)$.

The predicates for the relationships between e_{i-j} and m_j take a form of $f(e, m, v)$. In our study, we consider the number agreement (*entNumAgree*) and the gender agreement (*entGenderAgree*) between e_{i-j} and m_j . v is 1 if all of the mentions in e_{i-j} have consistent number/gender agreement with m_j , e.g., $entNumAgree(e_{1-6}, m_6, 1)$.

2. Predicates describing the belonging relations between e_{i-j} and its mentions. A predicate $has_mention(e, m)$ is used for each mention in e ³. For example, the partial entity e_{1-6} has three mentions, m_1 , m_2 and m_5 , which can be described as follows:

$has_mention(e_{1-6}, m_1)$.

$has_mention(e_{1-6}, m_2)$.

$has_mention(e_{1-6}, m_5)$.

3. Predicates describing the information related to m_j and each mention m_k in e_{i-j} . The predicates for the properties of m_k correspond to the features in the second part of Table 2 (removing the suffix $_mk$), while the predicates for the relationships between m_j and m_k correspond to the features in the third part of Table 2. For example, given the two mentions m_1 (“Microsoft Corp.”) and m_6 (“he”), the following predicates can be applied:

$nameNP(m_1, 1)$.

$pron(m_1, 0)$.

...

$nameAlias(m_1, m_6, 0)$.

$sentDist(m_1, m_6, 1)$.

...

the last two predicates represent that m_1 and

³If an active mention m_j is a pronoun, only the previous mentions in two sentences apart are recorded by *has_mention*, while the farther ones are ignored as they have less impact on the resolution of the pronoun.

m_6 are not name alias, and are one sentence apart.

By using the three types of predicates, the different knowledge related to entities and mentions are integrated. The predicate *has_mention* acts as a bridge connecting the entity-mention knowledge and the mention-pair knowledge. As a result, when evaluating the coreference relationship between an active mention and an entity, we can make use of the “global” information about the entity, as well as the “local” information of each individual mention in the entity.

From the training instances and the associated background knowledge, a set of hypotheses can be automatically learned by ILP. Each hypothesis is output as a rule that may look like:

$link(A,B):-$
 $predi1, predi2, \dots, has_mention(A,C), \dots, prediN.$
 which corresponds to first-order logic
 $\forall A, B(predi1 \wedge predi2 \wedge \dots \wedge$
 $\exists C(has_mention(A, C) \wedge \dots \wedge prediN)$
 $\rightarrow link(A, B))$

Consider an example rule generated from our system:

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

Here, variables A and B stand for an entity and an active mention in question. The first-order logic is implemented by using non-instantiated arguments C in the predicate *has_mention*. This rule states that a mention B should belong to an entity A , if there exists a mention C in A such that C is a bare noun phrase with the same head string as B , and matches in number with B . In this way, the detailed information of each individual mention in an entity can be captured for resolution.

A rule is applicable to an instance $link(e, m)$, if the background knowledge for the instance can be described by the predicates in the body of the rule. Each rule is associated with a score, which is the accuracy that the rule can produce for the training instances.

The learned rules are applied to resolution in a similar way as described in Section 3.2. Given an active mention m and a partial entity e , a test instance $link(e, m)$ is formed and tested against every

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

Table 3: statistics of entities (length > 1) and contained mentions

rule in the rule set. The confidence that m should belong to e is the maximal score of the applicable rules. An active mention is linked to the partial entity with the highest confidence value (above 0.5), if any.

5 Experiments and Results

5.1 Experimental Setup

In our study, we did evaluation on the ACE-2 V1.0 corpus (NIST, 2003), which contains two data sets, training and devtest, used for training and testing respectively. Each of these sets is further divided into three domains: newswire (NWire), newspaper (NPaper), and broadcast news (BNews). The number of entities with more than one mention, as well as the number of the contained mentions, is summarized in Table 3.

For both training and resolution, an input raw document was processed by a pipeline of NLP modules including Tokenizer, Part-of-Speech tagger, NP Chunker and Named-Entity (NE) Recognizer. Trained and tested on Penn WSJ TreeBank, the POS tagger could obtain an accuracy of 97% and the NP chunker could produce an F-measure above 94% (Zhou and Su, 2000). Evaluated for the MUC-6 and MUC-7 Named-Entity task, the NER module (Zhou and Su, 2002) could provide an F-measure of 96.6% (MUC-6) and 94.1% (MUC-7). For evaluation, Vilain et al. (1995)’s scoring algorithm was adopted to compute the recall and precision of resolution.

By default, the ALEPH algorithm only generates rules that have 100% accuracy for the training data. And each rule contains at most three predicates. To accommodate for coreference resolution, we loosened the restrictions to allow rules that have above 50% accuracy and contain up to ten predicates. Default parameters were applied for all the other settings in ALEPH as well as other learning algorithms

	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	66.8	55.0	60.3	64.2	53.4	58.3	64.6	60.6	62.5
- Mention-Pair (all mentions in entity)	66.7	49.3	56.7	65.8	48.9	56.1	66.5	47.6	55.4
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 4: Results of different systems for coreference resolution

used in the experiments.

5.2 Results and Discussions

Table 4 lists the performance of different coreference resolution systems. For comparison, we first examined the C4.5 algorithm⁴ which is widely used for the coreference resolution task. The first line of the table shows the baseline system that employs the traditional mention-pair model (MP) as described in Section 3.1. From the table, our baseline system achieves a recall of around 66%-68% and a precision of around 50%-60%. The overall F-measure for NWire, NPaper and BNews is 60.4%, 57.9% and 62.9% respectively. The results are comparable to those reported in (Ng, 2005) which uses similar features and gets an F-measure ranging in 50-60% for the same data set. As our system relies only on simple and knowledge-poor features, the achieved F-measure is around 2-4% lower than the state-of-the-art systems do, like (Ng, 2007) and (Yang and Su, 2007) which utilized sophisticated semantic or real-world knowledge. Since ILP has a strong capability in knowledge management, our system could be further improved if such helpful knowledge is incorporated, which will be explored in our future work.

The second line of Table 4 is for the system that employs the entity-mention model (EM) with “Any-X” based entity features, as described in Section 3.2. We can find that the EM model does not show superiority over the baseline MP model. It achieves a higher precision (up to 2.6%), but a lower recall (2.9%), than MP. As a result, we only see $\pm 0.4\%$ difference between the F-measure. The results are consistent with the reports by Luo et al. (2004) that the entity-mention model with the “Any-

X” first-order features performs worse than the normal mention-pair model. In our study, we also tested the “Most-X” strategy for the first-order features as in (Culotta et al., 2007), but got similar results without much difference ($\pm 0.5\%$ F-measure) in performance. Besides, as with our entity-mention predicates described in Section 4.2, we also tried the “All-X” strategy for the entity-level agreement features, that is, whether all mentions in a partial entity agree in number and gender with an active mention. However, we found this bring no improvement against the “Any-X” strategy.

As described, given an active mention m_j , the MP model only considers the mentions between m_j and its closest antecedent. By contrast, the EM model considers not only these mentions, but also their antecedents in the same entity link. We were interested in examining what if the MP model utilizes all the mentions in an entity as the EM model does. As shown in the third line of Table 4, such a solution damages the performance; while the recall is at the same level, the precision drops significantly (up to 12%) and as a result, the F-measure is even lower than the original MP model. This should be because a mention does not necessarily have direct coreference relationships with all of its antecedents. As the MP model treats each mention-pair as an independent instance, including all the antecedents would produce many less-confident positive instances, and thus adversely affect training.

The second block of the table summarizes the performance of the systems with ILP. We were first concerned with how well ILP works for the mention-pair model, compared with the normally used algorithm C4.5. From the results shown in the fourth line of Table 4, ILP exhibits the same capability in

⁴<http://www.rulequest.com/see5-info.html>

```

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

the resolution; it tends to produce a slightly higher precision but a lower recall than C4.5 does. Overall, it performs better in F-measure (1.8%) for Npaper, while slightly worse (<1%) for Nwire and BNews. These results demonstrate that ILP could be used as a good classifier learner for the mention-pair model.

The fifth line of Table 4 is for the ILP based entity-mention model (described in Section 4.2). We can observe that the model leads to a better performance than all the other models. Compared with the system with the MP model (under ILP), the EM version is able to achieve a higher precision (up to 4.6% for BNews). Although the recall drops slightly (up to 1.8% for BNews), the gain in the precision could compensate it well; it beats the MP model in the overall F-measure for all three domains (2.3% for Nwire, 0.4% for Npaper, 1.4% for BNews). Especially, the improvement in NWire and BNews is statistically significant under a 2-tailed t test ($p < 0.05$). Compared with the EM model with the manually designed first-order feature (the second line), the ILP-based EM solution also yields better performance in precision (with a slightly lower recall) as well as the overall F-measure (1.0% - 1.8%).

The improvement in precision against the mention-pair model confirms that the global information beyond a single mention pair, when being considered for training, can make coreference relations clearer and help classifier learning. The better performance against the EM model with heuristically designed features also suggests that ILP is able to learn effective first-order rules for the coreference resolution task.

In Figure 1, we illustrate part of the rules produced by ILP for the entity-mention model (NWire domain), which shows how the relational knowledge of entities and mentions is represented for decision making. An interesting finding, as shown in the last rule of the table, is that multiple non-instantiated arguments (i.e. C and D) could possibly appear in the same rule. According to this rule, a pronominal mention should be linked with a partial entity which contains a named-entity and contains an indefinite NP in a subject position. This supports the claims in (Yang et al., 2004a) that coreferential information is an important factor to evaluate a candidate antecedent in pronoun resolution. Such complex logic makes it possible to capture information of multiple mentions in an entity at the same time, which is difficult to implemented in the mention-pair model and the ordinary entity-mention model with heuristic first-order features.

6 Conclusions

This paper presented an expressive entity-mention model for coreference resolution by using Inductive Logic Programming. In contrast to the traditional mention-pair model, our model can capture information beyond single mention pairs for both training and testing. The relational nature of ILP enables our model to explicitly express the relations between an entity and its mentions, and to automatically learn the first-order rules effective for the coreference resolution task. The evaluation on ACE data set shows that the ILP based entity-model performs better than the mention-pair model (with up to 2.3% increase in F-measure), and also beats the entity-mention model with heuristically designed first-order features.

Our current work focuses on the learning model that calculates the probability of a mention belonging to an entity. For simplicity, we just use a greedy clustering strategy for resolution, that is, a mention is linked to the current best partial entity. In our future work, we would like to investigate more sophisticated clustering methods that would lead to global optimization, e.g., by keeping a large search space (Luo et al., 2004) or using integer programming (Denis and Baldridge, 2007). We hope that such efforts could further improve the coreference resolution performance.

Acknowledgements This research is supported by a Specific Targeted Research Project (STREP) of the European Union's 6th Framework Programme within IST call 4, Bootstrapping Of Ontologies and Terminologies STRategic REsearch Project (BOOTStrep).

References

- C. Aone and S. W. Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 122–129.
- V. Claveau, P. Sebillot, C. Fabre, and P. Bouillon. 2003. Learning semantic lexicons from a part-of-speech and semantically tagged corpus using inductive logic programming. *Journal of Machine Learning Research*, 4:493–525.
- A. Culotta, M. Wick, and A. McCallum. 2007. First-order probabilistic models for coreference resolution. In *Proceedings of the Annual Meeting of the North America Chapter of the Association for Computational Linguistics (NAACL)*, pages 81–88.
- J. Cussens. 1996. Part-of-speech disambiguation using ilp. Technical report, Oxford University Computing Laboratory.
- P. Denis and J. Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of the Annual Meeting of the North America Chapter of the Association for Computational Linguistics (NAACL)*, pages 236–243.
- X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 135–142.
- A. McCallum and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of IJCAI-03 Workshop on Information Integration on the Web*, pages 79–86.
- J. McCarthy and W. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Conference on Artificial Intel-ligences (IJCAI)*, pages 1050–1055.
- R. Mooney. 1997. Inductive logic programming for natural language processing. In *Proceedings of the sixth International Inductive Logic Programming Workshop*, pages 3–24.
- V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 104–111, Philadelphia.
- V. Ng. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 157–164.
- V. Ng. 2007. Semantic class induction and coreference resolution. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 536–543.
- W. Soon, H. Ng, and D. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.
- L. Specia, M. Stevenson, and M. V. Nunes. 2007. Learning expressive models for words sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 41–48.
- A. Srinivasan. 2000. The aleph manual. Technical report, Oxford University Computing Laboratory.
- M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 45–52, San Francisco, CA. Morgan Kaufmann Publishers.
- X. Yang and J. Su. 2007. Coreference resolution using semantic relatedness information from automatically discovered patterns. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 528–535.
- X. Yang, J. Su, G. Zhou, and C. Tan. 2004a. Improving pronoun resolution by incorporating coreferential information of candidates. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 127–134, Barcelona.
- X. Yang, J. Su, G. Zhou, and C. Tan. 2004b. An NP-cluster approach to coreference resolution. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 219–225, Geneva.
- G. Zhou and J. Su. 2000. Error-driven HMM-based chunk tagger with context-dependent lexicon. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 71–79, Hong Kong.
- G. Zhou and J. Su. 2002. Named Entity recognition using a HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 473–480, Philadelphia.